



Advancements in Optimization: Critical Analysis of Evolutionary, Swarm, and Behavior-Based Algorithms

Noor A. Rashed ^{1,*} , Yossra H. Ali ¹ and Tarik A. Rashid ^{2,*} 

¹ Computer Sciences Department, University of Technology, Baghdad 10081, Iraq; yossra.h.ali@uotechnology.edu.iq

² Computer Sciences & Engineering Department, University of Kurdistan Hewler, Erbil 44001, Iraq

* Correspondence: cs.20.62@grad.uotechnology.edu.iq (N.A.R.); tarik.ahmed@ukh.edu.krd (T.A.R.)

Abstract: The research work on optimization has witnessed significant growth in the past few years, particularly within multi- and single-objective optimization algorithm areas. This study provides a comprehensive overview and critical evaluation of a wide range of optimization algorithms from conventional methods to innovative metaheuristic techniques. The methods used for analysis include bibliometric analysis, keyword analysis, and content analysis, focusing on studies from the period 2000–2023. Databases such as IEEE Xplore, SpringerLink, and ScienceDirect were extensively utilized. Our analysis reveals that while traditional algorithms like evolutionary optimization (EO) and particle swarm optimization (PSO) remain popular, newer methods like the fitness-dependent optimizer (FDO) and learner performance-based behavior (LPBB) are gaining attraction due to their adaptability and efficiency. The main conclusion emphasizes the importance of algorithmic diversity, benchmarking standards, and performance evaluation metrics, highlighting future research paths including the exploration of hybrid algorithms, use of domain-specific knowledge, and addressing scalability issues in multi-objective optimization.

Keywords: single-objective optimization; metaheuristic algorithms; multi-objective optimization; optimization algorithms; evolutionary algorithms; scheduling; algorithmic advancements



Citation: Rashed, N.A.; Ali, Y.H.; Rashid, T.A. Advancements in Optimization: Critical Analysis of Evolutionary, Swarm, and Behavior-Based Algorithms. *Algorithms* **2024**, *17*, 416. <https://doi.org/10.3390/a17090416>

Academic Editors: Mohammad Rohaninejad and Frank Werner

Received: 3 August 2024

Revised: 25 August 2024

Accepted: 2 September 2024

Published: 19 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multi-objective optimization (MOO) has gained attention and become a topic of application and study in recent times [1]. Those who practice and research have attempted to tackle numerous kinds of optimization and search problems faced in practical situations. They did this by presenting computationally efficient as well as novel optimization algorithms that were aimed at addressing these issues. Real-world problems of optimization are not simple to solve because they can be found in many types or forms. There might be just one goal to be achieved in certain optimization problems, several competing objectives in others, severe constraints in some, and multiple optimal solutions in others. To solve these problems, a user first evaluates the underlying problem and selects an appropriate algorithm [2].

This is because a single-objective optimization problem's lone optimum cannot be effectively found by an algorithm that is designed to identify several optimal solutions to another optimization problem. To address a variety of issues, a user must be knowledgeable about a variety of algorithms, each of which is focused on resolving a specific class of optimization problems [3], whether they include maximizing or minimizing values, or employing a single objective or several objectives [4]. The optimization problems typically include multiple objectives that conflict with one another, making a single solution unfeasible. Rather, our goal is to identify optimal "trade-off" solutions that optimally balance the various objectives [5]. MOO refers to problems with multiple objectives. Various fields, including engineering, mathematics, economics, social studies, aviation, agriculture, and

automobiles, encounter this kind of issue daily [6,7]. EO algorithms use a population of solutions approach, where various solutions are involved in each iteration and aid in the creation of a new population of solutions [8,9]. Their broad appeal is a result of several reasons, including (i) no derivative information being required for executive orders (EOs); (ii) EOs are quite simple to execute; and (iii) EOs have a wide range of applications and are adaptable. While it might not seem necessary to use a population of solutions to find the single best solution for single-objective optimization problems, doing so is a great way to solve MOO problems. The inventors of PSO classify their heuristic search method as an evolutionary algorithm [6]. Scholars are now investigating the use of the PSO method in different disciplines due to its success as a single-objective optimizer, especially in continuous search spaces. The multi-objective particle swarm optimizer (MOPSO) was first proposed [10,11], and the first PSO technique modification for solving multi-objective problems was presented in a 1999 paper [12].

The field of optimization algorithms has seen significant advancements, with evolutionary, swarm, and behavior-based methods playing a pivotal role in solving complex computational problems. Despite extensive research, the integration of multi-objective optimization into behavior-based algorithms remains underexplored. This paper addresses this gap by introducing a novel framework that enhances computational efficiency and solution quality. Our comprehensive review of the literature from 2000 to 2023, utilizing sources like IEEE Xplore, ScienceDirect, and Web of Science, uncovers emerging trends and identifies key contributors shaping the future of optimization algorithms.

After this first attempt, researchers were highly interested in expanding PSO; nevertheless, strangely, the next suggestion was not published until 2002. Nonetheless, around twenty-five different MOPSO concepts are already documented in the specialized literature. The authors offer a comprehensive analysis of the many MOPSO algorithms that have been published in academic journals [11,13,14].

Differential evolution (DE) is a stochastic optimization approach that makes use of the direct search evolution technique. It is renowned for being reasonably sturdy and quick. DE was developed in the 1990s and was used in practice to address problems with scientific optimization [15]. Its application was extended to various problem areas, such as MOO, due to its established achievements [16]. DE was applied to address problems in multi-objective domains in several recent studies. The literature has described this research [16–19].

Abdullah and Ahmed (2019) state that the fitness dependent optimizer (FDO) is a computational program that takes its cues from bee swarm reproduction. This algorithm simulates a swarm of bees going through their reproductive behavior. The core idea of the algorithm comes from the way scout bees search among several possible hives for a new, ideal colony. This program views each one of the scout bees that visit new hives as a possible solution. Furthermore, selecting the best hive from a collection of quality hives is thought to be convergent toward optimality [20].

In the year (2021), Rahman and Rashid introduced the idea of learner performance-based behavior (LPB). The learner performance behavior-based algorithm, a new optimization technique, was proposed. The LPB technique mimics the procedures used by different institutions to admit high school graduates and the actions taken by these students that affect their academic performance in college. It also looks at the elements that could help students change their high school study routines that do not work for college-level coursework. To accomplish this, multiple populations may be used to present students with different GPAs. This ultimately results in a beneficial balance between exploitation and exploration [21].

The need for efficient and effective optimization approaches that can handle multi-objective decision-making problems in the actual world is what motivated this study. Researchers want to take advantage of current algorithmic advances while addressing the inherent difficulties in MOO tasks by extending single-objective algorithms to multi-objective frameworks. By expanding upon proven single-objective optimization methods,

this approach not only increases the adaptability of optimization algorithms but simplifies the development process. This study provides a thorough overview of single and MOO algorithms, including DE, EO, FDO, PSO, and LPB, as well as a concise description of each algorithm. These algorithms can be extended to solve multi-objective world problems. This study's primary contribution is a review of multi-and single-objective algorithms, which aims to provide academics studying optimization algorithms with up-to-date, comprehensive knowledge. The main contributions are as described below:

1. The presented work consolidates the current state-of-the-art approaches in multi- and single-objective optimization algorithms, offering a holistic understanding of both classical methodologies and modern metaheuristic approaches.
2. By conducting a detailed examination of various algorithms, and their conceptual underpinnings, strengths, and limitations, this paper provides valuable insights into their applicability across various problem domains.
3. The review identifies recent advancements, comparative studies, and emerging trends in optimization research, shedding light on novel techniques and methodologies that are shaping the field's future trajectory.
4. By emphasizing the importance of collaboration between academia and industry, as well as the need for innovative approaches to address real-world challenges, the paper aims to stimulate further research and development in optimization.
5. The study offers a roadmap for future research endeavors by identifying important research gaps and highlighting prospects. It encourages the investigation of hybridization techniques, domain-specific optimizations, and scalability advances in MOO.

The process of developing popular intelligent optimization algorithms through strong publication journals is depicted in Figure 1.

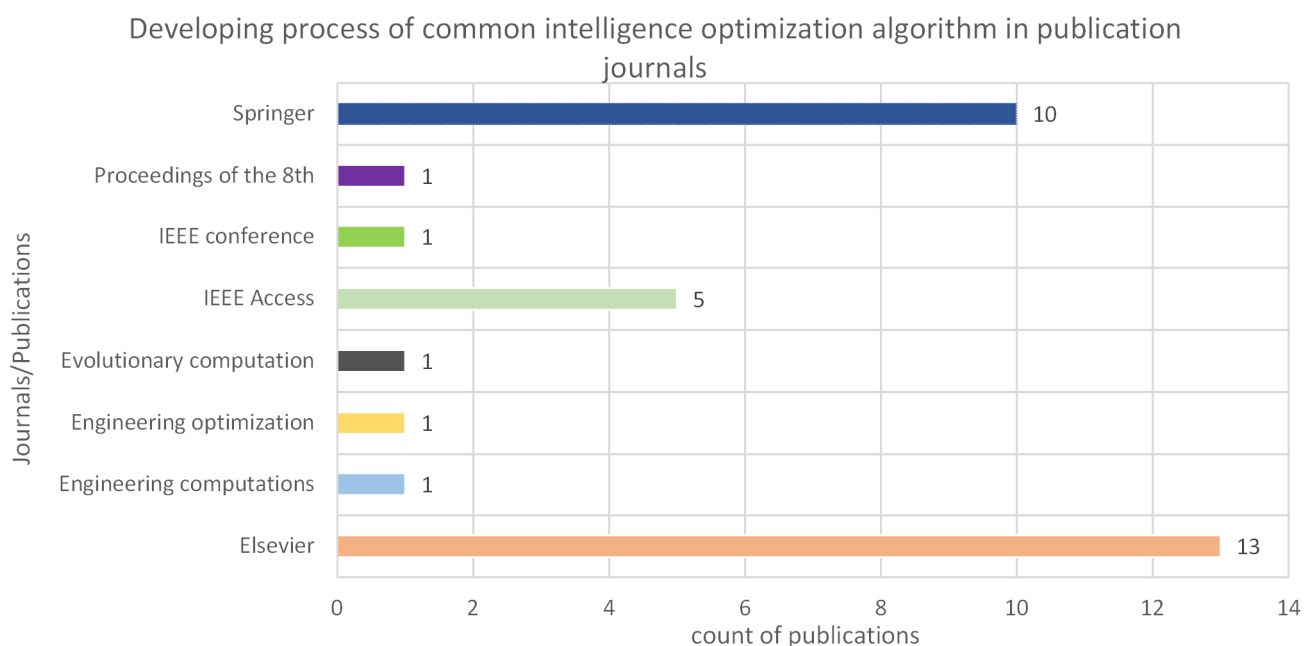


Figure 1. Development process of common intelligent optimization algorithms in publication journals.

The overview of the methodology of the literature review for single algorithms extended to multi-objective algorithms is briefly covered in the section that follows. The next section is a content analysis of literature and bibliometric review. Next, a brief discussion is held on a few single and extended multi-objective algorithms. In the section on multi-objective approaches and applications, the multi-objective algorithms and subsequent multi-objective algorithms are compared.

2. Methodology of Literature Review

2.1. Information Source

The literature review was conducted using databases such as IEEE Xplore, Springer-Link, and Elsevier ScienceDirect. Keywords like “multi-objective optimization”, “evolutionary algorithms”, “swarm algorithms”, and “behaviour-based algorithms” were used. The reviewed studies span from 2000 to 2023. Articles were selected based on relevance, citation count, and publication in high-impact journals. Initial screening involved a title and abstract review, followed by full-text analysis to ensure comprehensive coverage of the topic, as shown in Figure 2.

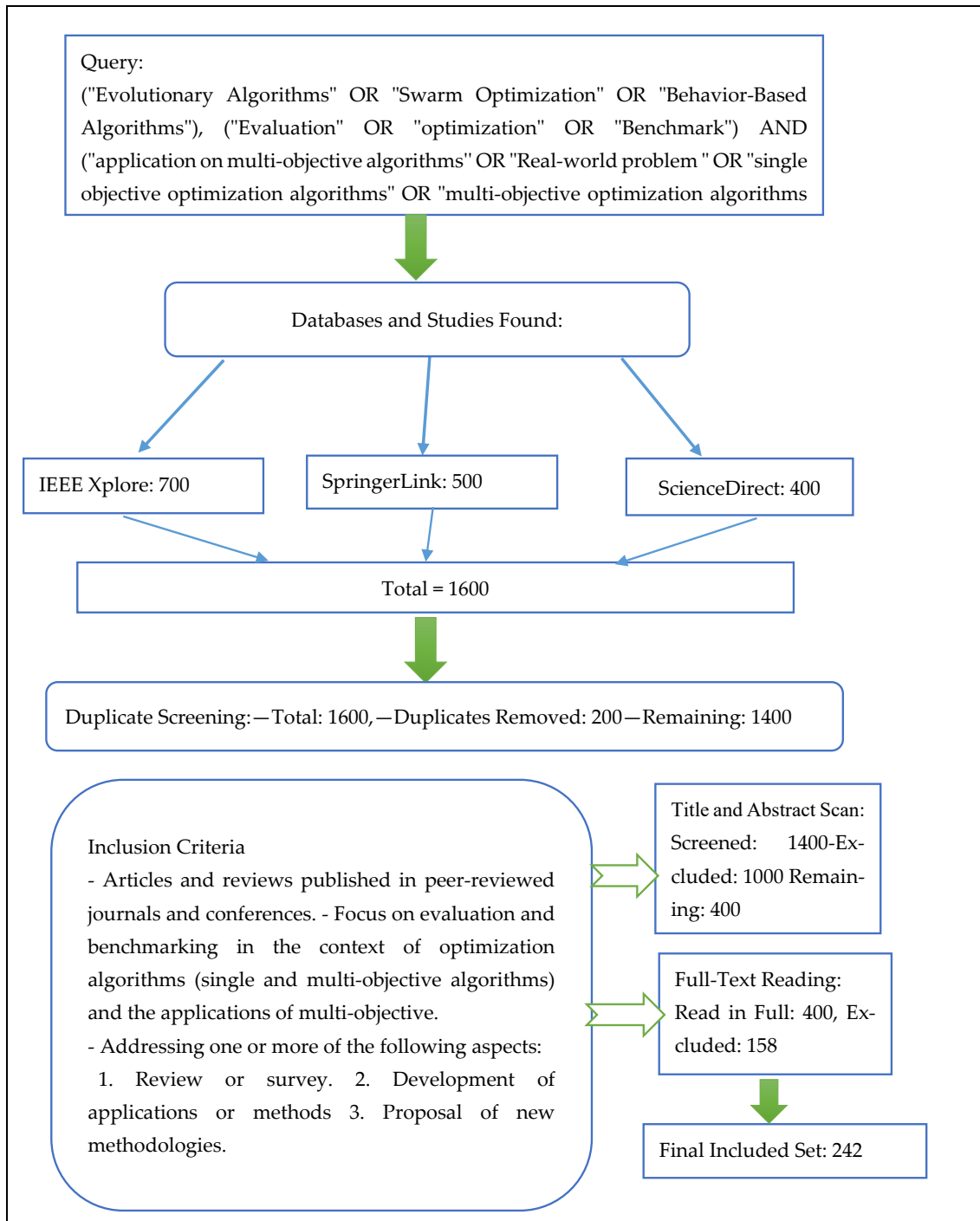


Figure 2. Study selection and screening process for literature review of optimization algorithms.

2.2. Database Selection

The databases used for this review were IEEE Xplore, SpringerLink, and ScienceDirect. These databases were chosen for their comprehensive coverage of the fields of computer science and optimization algorithms.

2.3. Time Frame

The review covered publications from the period 2000–2023. This period was selected to capture the significant advancements in optimization algorithms over the past two decades. The study selection procedure included an intensive search of relevant publications that depended on two iterations. Firstly, the titles and abstracts of the papers were scanned to exclude unrelated and duplicate papers. Secondly, after carefully reading the full text of the screened papers (from the first iteration step), researchers organized the papers.

2.4. Search Strategy

On 2 March 2023, we conducted a search query on the IEEE Xplore, SpringerLink, and ScienceDirect databases using their respective search boxes. Searches were performed in all databases by entering the terms as a search string.

The search strategy included using specific keywords such as “evolutionary algorithms”, “swarm optimization”, “behaviour-based algorithms”, and “multi-objective optimization”. Boolean operators were used to refine the search results.

The advanced search capabilities in each search engine were utilized to specifically filter for journals and conference papers while excluding book chapters and other document formats. We reviewed scholarly articles and conference proceedings that were likely to have been involved in current and relevant scientific research connected to our work. Table 1 displays the configurations employed to execute the search query.

Table 1. Settings of the search query in digital libraries.

Digital Library	Years	Language	Run On	Subject Areas	Date of Running Search String
IEEE Xplore	2000–2023	Only English	Full text	All available	2023
SpringerLink	2000–2023	Only English	Full text	All available	2023
ScienceDirect	2000–2023	Only English	Full text	Chemical engineering, computer science, engineering	2023

2.5. Inclusion and Exclusion Criteria

All papers that met the criteria presented in Figure 2 were included. The first purpose was to map the research landscape with single- and multi-objective algorithms. These groups were established based on an extensive pre-survey of the literature sources. After eliminating the duplicate articles, all the items that did not fulfil the requirements for qualifying were removed, as stated in Figure 2. The exclusion criteria were as follows:

- i. Inclusion criteria:
 - Articles and reviews published in peer-reviewed journals.
 - Conference papers presenting significant advancements or novel algorithms.
 - Studies focusing on single- and multi-objective optimization.
- ii. Exclusion criteria:
 - Non-peer-reviewed articles.
 - Studies not available in full text.
 - Publications not in English.

2.6. Data Extraction and Analysis

The selected studies were analyzed using bibliometric methods to assess the frequency of publications per year, types of works (scientific articles, reviews, conference materials), and key metrics like citation counts. Keyword analysis was conducted to identify prevalent themes and trends in the field. Excel was used to organize a list of all included articles that were compiled from various sources with their corresponding initial categories.

2.7. Summary Analysis of Single Algorithms Extended to Multi-Objective Algorithms

In Deb K. et al. (2002) [8], Non-Dominated Sorting Genetic Algorithm II (NSGA-II) was presented by the authors. A novel methodology is suggested for effectively solving MOO difficulties, which addresses the drawbacks of earlier multi-objective genetic algorithms (GAs), like their high computational complexity and non-elitism approach. A more recent iteration of NSGA-II, known as NSGA-III [22,23], was enhanced to handle numerous objective optimization problems with up to 15 goals inside by using a reference point [24]. In Cai X. et al. [2], constrained decomposition with grids (CDG) is a technique that was proposed to address the problems of sensitivity to Pareto front shapes and diversity loss which could occur in decomposition-based multi-objective evolutionary algorithms. For example, a unique differential evolution approach for MOO was introduced and proven by Fang et al. [18]. To replace the existing selection method, the research proposed a new one that makes it possible to apply differential evolution (DE) to the solution of both multi- and single-objective optimization problems. Comparing the trial population solution with its counterpart in the current population is a step in the selection process. The trial candidate will be chosen to live to the next generation and replace the present population vector if its dominance surpasses that of the current population member. If not, the current member of the population will be retained. They suggested rejecting the trial solution if it fails to meet any of the objectives of the target solution. Through its application to three multi-objective benchmark optimization problems, the strategy's validity was verified. The simulation results show that for the selected issues, the technique may produce an approximate Pareto front [14]. A variation of the generalized differential evolution (GDE) algorithm was presented by Lampinen and Kukkonen in the year 2005, and it might be applied to both multi- and single-objective problems. The original differential evolution algorithm was applied to unrestricted single-objective problems. To produce more efficient distributed solutions in MOO problems, GDE3, an improved version of GDE, was applied [18].

In Coello, C.A.C. and G.T. Pulido et al. (2005), to solve MOO problems, the authors used PSO [25]. This method involved using a separate set of particles to aid the other particles in their movements. Margarita [6] and colleagues (2006) presented the first proposal for a MOPSO, and offered a comprehensive analysis of several MOPSOs described in the specialized literature. We will categorize the approaches and list the salient features of each proposal as part of this study. The study's last part lists several topics in this field that we think may make good subjects for additional research. A hybrid method which combines the benefits of the PSO algorithm as well as genetic algorithms was proposed by Mousa et al. (2012).

In addition, a neighbor search engine has been utilized as a local search operator for improving the quality of the solution. This engine concentrates on examining the less populated area of the external archive to increase the quantity of non-dominated solutions. To provide more solutions close to the personal best, Qu et al. (2012) proposed incorporating a local search approach into the niching PSO. To particularly handle MOO problems, Yang, X.-S. et al. (2012) created the bat algorithm (BA). An adaptation of the flower pollination optimization approach was created by Yang X.-S. et al. (2014) to solve MOO problems.

Mirjalili S. et al. (2016) [3], for the first time, offered a cutting-edge method for effectively optimizing problems with multiple objectives, known as the multi-objective grey wolf optimizer (MOGWO). A fixed-sized external repository is combined into GWO. This helps in storing and getting back Pareto-optimal solutions. The process of gathering this data is then utilized to establish a social ranking and simulate hunting behavior among grey wolves

within multi-objective search situations. The new approach's performance was evaluated on ten multi-objective benchmark issues, and compared to two popular meta-heuristics: MOPSO and the multi-objective evolutionary algorithm based on decomposition (MOEA/D).

In Kumawat I.R. et al. (2017) [4], to tackle the difficulties caused by MOO problems, a multi-objective variant of the whale optimization algorithm (MOWOA) was created by the authors. In Li L. and W. Wang et al. (2017) [26], to deal with the difficulties brought about by multi-objective particle swarm optimization problems (MOP), the writers made a new version of PSO. Inside this edition, they introduce a new global margin ranking (GMR) that uses positional information from individuals in the objective space to define dominant borders across a population. In Chai R. et al. (2017) [7], the aero-assisted spacecraft trajectory optimization problem was dealt with by the authors through presenting a new multi-objective technique. This algorithm was created using an enhanced version of the NSGA-II algorithm, including a discretization method for formulation and parameterization.

In Mirjalili et al. (2017) [27], a new version of the ant lion optimizer (MOALO), which is based on how ant lions hunt and interact with ants, was presented by the writers. At first, the repository keeps all non-dominated Pareto-optimal solutions that have been discovered until now. In this case, we use a roulette wheel mechanism to select the best responses from this set. The roulette wheel considers the number of antlions linked to every solution for finding out which multi-objective search regions are most excellent. Several standard unconstrained and limited test functions are used to show the effectiveness of the method being studied. This methodology is applied to tackling various multi-objective engineering design problems.

In El Aziz et al. (2018) [10], the authors proposed tackling the multi-objective multilevel thresholding image segmentation task using the weighted optimization algorithm (WOA). Furthermore, there is the GWO. In Mirjalili et al. (2018) [28], to address MOO problems, the authors proposed integrating the grasshopper algorithm with the archive and goal selection approach. In Tawhid M.A. et al. (2019) [29], to handle multi-objective engineering design problems, the authors presented a multi-objective sine-cosine algorithm (MOSCA). To produce a range of non-dominated solutions and maintain the diversity among them, it makes use of elitist non-dominated sorting and crowding distance techniques. In Lai X. et al. (2019) [30], to solve MOO problems, the authors introduced the artificial sheep method, which combines an archive as well as leader selection method with Pareto-based theory. In Liang J. et al. (2019), the authors proposed a multi-modal multi-objective differential evolution (MMOEA) optimization algorithm [31] as a solution for Pareto multi-modality problems, where the Pareto set is composed of numerous distinct subsets. By allowing offspring outside the search space to experience additional mutations, MMOEA utilized a new mutation-constrained approach to reduce the concentration regarding individuals located at the search space boundaries. A similar analysis has been carried out by Cheng and Tian (2017) [32] to evaluate Pareto-optimum solutions on the Pareto-optimal fronts of different geometries. In Santiago A. et al. (2019) [33], for this objective, the authors used an enhanced version of the inverted generational distance indicator. Fuzzy logic is used in certain research in the literature to improve evolutionary MOO (EMO).

Via Eric S. Fraga et al. (2019) [34], several methods have been proposed to deal with such problems, which include using weighting methods for reducing multi-objective problems to only one objective. It is recommended to use an algorithm inspired by the nature of plant propagation to solve control and design problems in dynamic processes. Because it is population-based, it can concurrently find the Pareto-optimal frontier approximation. A representative sample from the literature is used to demonstrate how this problem could be expressed. We cover the formulation of the problem and how the Fresa system is used for MOO. The result shows how using a multi-objective method could lead to a better comprehension of control or design problems. The situation that is being portrayed was first thought to have a single goal. This may obscure information about different designs and how they might affect the functionality of the system. By giving more thorough information, the engineer can improve their design decision-making process by employing a multi-objective formulation.

In Hamid Afshari et al. (2019) [35], the authors suggest using a multi-objective approach to optimally design reinforced concrete beams, with an emphasis on determining the best course of action that strikes a compromise between cost and deflection. Next, they compare six popular MOO techniques—one of which is based only on random point selection—by looking at how well they solve the design challenge. The consistency and ranking of outcomes in a derivative-free optimization method help to understand its effectiveness. To identify the most effective algorithms for resolving the test problem, this study explores gradient-based and derivative-free optimization (DFO) techniques. So, in the case where we are comparing algorithms for structural problems, we consider different factors like efficiency, development and application. Deterministic methods are better than evolutionary algorithms because they do not have unpredictability in the final solution, and they show good efficiency.

In Inês Costa-Carrapiço et al. (2020) [22], to assess the possibility of combining GA with MOO to improve the development regarding retrofitting methods and its decision-making process, a thorough study of the literature was conducted. Of the 557 studies assessed, 57 were particularly investigated to evaluate the approach's potential, obstacles, and limitations, as well as the studies' conclusions and current trends. The main conclusions show that a wide range of building retrofit MOO problems can be effectively addressed. The strong outcomes attained, which demonstrate notable advancements in target achievement, corroborate this. Yet, due to time-consuming and efficacious issues, the results imply that to obtain the optimal retrofit solutions, modified GA or GA-mixed techniques might be required. Less focus has been placed on heritage buildings, which present a unique problem in identifying a qualitative objective function. The lack of a standardized systematic approach and the complex transition between the modeling, as well as the optimization environment, the need for a high level of expertise to perform MOO and operate the software, and the lack of confidence in the results obtained are further challenges. Even if more investigation is required to properly assess how well GA-based MOO supports building retrofit and its decision-making process, when paired with auxiliary techniques, it has a lot of potential.

In Mohammed et al. (2020) [23], for each objective in a multi-objective problem (MOP), the authors suggest a comparative study between single-objective evolutionary algorithms (SOEAs) and MOEAs, with particular attention to the bi-objective situation. The knapsack problem (KNP) and traveling salesman problem (TSP) are two optimization problems that have been studied extensively. Three MOEAs and two SOEAs form part of this experimental investigation. The SOEA treats each optimization objective separately. This helps us compare the multiple-objective solutions made by MOEAs with the best values for every objective. MOEAs, which stand for multi-objective evolutionary algorithms, can optimize two objectives at once. This idea is manageable because it is likely to evaluate the ideal sites for every objective separately by utilizing Pareto fronts created from MOEAs. The real outcomes demonstrate that MOEAs could compete with SOEAs and manage many objectives simultaneously, particularly when examples are large or strongly correlated.

In C et al. (2020) [14], the authors used multi-agent systems and Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) to solve the multi-objective optimization problem (MOOP) with a focus on logistics management. The first solution for MOPs was stochastic fractal search (SFS) [36] designed by Khalil Pourazari S. B. et al. in 2020. The validation of SFS was limited to the CEC 2009 benchmark test. So, we do not know how effective it is on other benchmark problems like GLT and DTLZ. In Grotti E. et al. (2020), to overcome suspension optimization problems, a unique multi-objective quantum particle optimizer based on an archive was formulated (MOQPSO) [37]. The basic equations of QPSO were adjusted and more methods to handle various goals were included, thus forming the MOQPSO algorithm. In addition, Chai et al. (2017) [7] presented an innovative multi-objective approach based on swarm intelligence to determine the optimal overtaking trajectory for autonomous ground vehicles. The purpose of this method was to enhance the time length of the gambit, visibility, and smoothness in the vehicle's path while considering mission-based limitations.

In Nianyin Zeng et al. (2021) [38], the authors recommended using a competitive method. A possible solution to MOO issues is given in the form of an integrated whale optimization algorithm (CMWOA). The study might be able to construct a stronger leader to control the update of whale groups by applying a new competitive procedure. This will be better for the algorithm's convergence and deliver better outcomes. Furthermore, it is worth mentioning that the competitive procedure uses a more advanced crowding distance computation where multiplication replaces the standard addition operation. This adjustment helps to represent population density more precisely. Differential evolution (DE) is a different method used to make the population diverse. We can use many changing techniques for optimizing important DE parameters, which improves overall efficiency. A thorough assessment is conducted of the suggested comprehensive multi-objective optimization algorithm (CMWOA) on various benchmark functions that show distinct versions of the true Pareto front. Based on the outcomes of various performance metrics, it is shown that in most situations, the suggested CMWOA method performs better than the other three methods. The discussion also includes implications of model parameters. Three real-world instances showed an effective application of the proposed CMWOA algorithm, which offers more evidence for its usability.

In Mohamed Abdel-Basset et al. (2021) [39], to lessen the drawbacks of addressing MOO problems, an improved and extended version of the whale optimization algorithm (WOA) is suggested. Their improvements included (i) modification of the standard WOA's distance control factor to include dynamically generated values rather than a fixed value; (ii) balancing the movement towards the opposite of the optimal solution and its original values depending upon a specific probability to prevent being stuck in local minima; and (iii) speeding up coverage and convergence using the Nelder–Mead method and the Pareto archived evolution strategy. By contrasting the suggested method with nine reliable multi-objective algorithms, its efficacy is assessed. Three benchmark multi-objective test functions (which are CEC2009, DTLZ, and GLT), totaling twenty-five test functions, are used in this evaluation. The experiments demonstrate how the suggested algorithm performs better than some of the existing multi-objective algorithms that have been published in the literature.

In Ryan H. Stewart et al. (2021) [40], the authors offer a thorough review of the existing body of research on MOO that has already been conducted. This covers the basics of optimization theory, commonly used MOO algorithms, and the field's recent developments in nuclear science and engineering. The techniques presented in this study are intended to investigate the user's design space to identify a set of best practices. Typically, such solutions are evenly distributed across the Pareto front and non-dominated. This gives the designer a variety of possibilities to pick from, none of which would improve one objective function without sacrificing another. Two different categories of MOO methods are presented: meta-heuristic and classical. Those groups are then divided into four different approaches (progressive, a priori, no preference, and a posteriori) which are used to determine which answers are the best. Every approach discussed has benefits and drawbacks that vary depending on variables including user participation volume, solution sets' efficacy, and implementation costs. This study gives academics a thorough understanding of the MOO algorithms that are presently available. Additionally, it provides a basic understanding of how these methods can be used in a broad range of problems in the disciplines of engineering and nuclear science. A thorough understanding of how to apply these techniques to a variety of problems in the disciplines of engineering and nuclear science.

In Todor Balabanov et al. (2021) [41], the authors described a computational technique that generates solutions for multi-objective problems inside the Pareto subset using the LibreOffice Calc NLP Solver. This work examines the performance of a solver intended for single-objective problems on multi-objective problems. Several solutions near the Pareto front are produced by giving the objectives random weights. Since the solver is inherently random, the suggested solutions are positioned close to the front rather than directly there.

This is not an acceptable scenario from a mathematical standpoint. Practically speaking, though, it is better to have even acceptable answers when no alternatives are available.

In Shanu Verma et al. (2021) [42], the authors provide a thorough examination of the popular MOO method NSGA-II for combinatorial optimization tasks. The assignment problem, vehicle routing problem, traveling salesman problem, allocation problem, scheduling problem, and knapsack problem are among the problems that NSGA-II has been utilized to solve. Three NSGA-II versions can be applied: Modified NSGA-II, Conventional NSGA-II, and Hybrid NSGA-II. In addition to analyzing the modifications made to NSGA-II, the researchers look at the many approaches taken to evaluate its effectiveness, including statistical testing, case studies, performance metrics, test instances, and benchmarking against other cutting-edge algorithms. This work also provides a succinct bibliometric overview of the research that was conducted.

In Baihao Qiao et al. (2021), to deal with the drawbacks of DEED situations, the authors suggested a method using proportional dynamic adjustment decision (PDAD) variables considering the unit's power-generating range. The dynamic slack variable (DSL) approach can be described as a new way that brings in better constraint management methods from the slack variable approach. Additionally, they introduce NSDESa_LS which is an original variation of the sorting differential evolution method. To effectively tackle the DEED challenges, the solution uses a local search operator along with a self-adaptive parameter operator. Lastly, how well the suggested approach works is compared against other current techniques on systems of 5, 10, and 40 units. The outcomes demonstrate the good performance of this NSDESa_LS-PDAD technique.

In Khodadadi et al. (2021) [43], motivated by the concept of constructing crystal structures, a multi-objective crystal structure algorithm (MOCryStAl) was introduced recently. The main three parts in the approach are the grid, choosing leaders, and archiving. The effectiveness of this method is determined by its use in real-world engineering design problems as well as mathematical ones. The outcomes demonstrate that employing the suggested methods could create significant enhancements in handling the complex problems under investigation.

In M. Premkumar et al. (2022) [44], the authors with complicated optimization problems; for instance, real-life engineering design optimization issues were handled by the researchers introducing a fresh multi-objective equilibrium optimizer (MOEO). One metaheuristic algorithm that was recently published and finds its base in physics is known as the equilibrium optimizer (EO). It takes inspiration from models which forecast dynamic and equilibrium states. The MOEO technique uses a similar process, including models in a new target search area. During the evolution of the search, crowding distance was applied to balance between exploring and exploiting stages. The MOEO algorithm keeps an important aspect in multi-objective metaheuristic algorithms, which is population variation through a non-dominated sorting technique. Optimal solutions are used for maintaining and improving Pareto's extent with the use of a repository with an update function. The evaluation of 33 contextual problems, comprising 12 unconstrained, 6 constrained, and 15 realistic constrained engineering design challenges, such as non-linear problems, confirmed the efficacy of MOEO. The performance of the suggested MOEO algorithm was compared to that of existing state-of-the-art MOO methods. The results show that the recommended MOEO produces more competitive outputs than the other algorithms, both in terms of number and quality. The findings obtained for all 33 benchmark optimization problems provide a comprehensive description and clarification of the robustness, efficiency, and exploratory potential of the MOEO algorithm for solving multi-objective problems.

In Yuan Mei et al. (2022) [45], the five ADMET properties were taken into account, MOO was used, a model was created to map the relation between molecular descriptors as well as their biological activities, and the process of choosing molecular descriptors was studied. To choose possible medication candidates for the treatment of breast cancer, a thorough process was created. More specifically, the next contributions are presented in this study. Different perspectives on unsupervised spectral clustering are used to

construct a new feature selection technique. The selected features demonstrate decreased redundancy and an improved ability to represent comprehensive information. The issue is resolved by enhancing and applying the AGE-MOEA algorithm¹⁴, which is founded on the examination of the interaction between conflicting optimization targets. With regard to search performance, the improved approach performs better than many existing algorithms. The process of choosing candidate compounds with anti-breast cancer characteristics has been aided by the establishment of a thorough framework.

Milica Petrović et al. (2022) [46] suggest the MOGWO, which is a novel method that can be used to efficiently organize material transport systems with a single, intelligent mobile robot. The optimization process that is suggested entails a careful analysis and the mathematical development of thirteen new fitness functions. The MOO problem is solved by coupling such functions to form a Pareto front. Furthermore, a novel approach to efficiently explore the search space of numerous objectives is shown. In addition, three advanced metaheuristic approaches (MOAOA, MOGA, and MOPSO) and the suggested enhanced MOGWO algorithm are quantitatively evaluated and compared for their efficiency on 25 benchmark problems using four metrics: inverted generational distance (IGD), maximum spread (MS), generational distance (GD), and spacing (SP). Results from two experimental settings showed that the improved MOGWO is superior in terms of the coverage, convergence, and robust solution generation when compared to previous algorithms. The results of testing provided clear indications of the effectiveness of the proposed approach for the scheduling of material transport jobs with one mobile robotic system function.

In Rahman and Rashid (2022) [47], the multi-objective learner performance-based behavior algorithm (MOBP), which is a new MOO algorithm, was presented by the authors to provide a performance-based learner behavior system for addressing engineering optimization problems with multiple objectives. They used five real engineering problems in total to evaluate the method and compared it against three other solvers designed for multiple objectives. The results indicate that the performance of this suggested algorithm was superior in terms of variety and accuracy compared to its counterparts.

In Abdullah and Rashid (2023) [48], the authors introduced the MOFDO algorithm, which is a fitness-dependent optimizer with multiple objectives. The study discussed and evaluated the MOFDO method on both artificial benchmark problems and real-life engineering situations. As per the findings, it appears that MOFDO shows improved performance in terms of convergence and solution variety compared with other optimization approaches. Furthermore, MOFDO is a handy tool to tackle engineering design problems in the actual world because it provides decision-makers with many feasible alternatives for selection.

Ahmed and Rashid (2023) [49] offer a multi-objective version of the cat swarm optimization algorithm, which is referred to as the grid-based multi-objective cat swarm optimization algorithm (GMCSO). Modern multi-objective algorithms primarily aim to produce robust results by preserving diversity and achieving convergence. They first replace the roulette wheel technique of the original CSO algorithm with the greedy strategy for achieving such objectives. The grid architecture and the twofold archive approach—two key tenets of the Pareto archived evolution approach algorithm (PAES)—are applied. Several test functions, as well as the pressure vessel design problem—a real-world scenario—are used for the evaluation of the efficacy of the proposed approach. In the experiment, some metrics, such as the spacing measure, spread metric and reversed generational distance, are used to compare the suggested method with other well-known algorithms. The optimization results showed the robustness of the suggested approach, and the use of statistical methods and graphical displays validates their validity even more.

3. Content Analysis of Literature and Bibliometric Review

This section integrates the findings from the bibliometric and content analysis to discuss the broader implications for the field of optimization algorithms. The trends observed in publication frequency and the types of work being published suggest a growing diversification in research topics and methods. Key contributors identified in the bibliometric

analysis are recognized for their pioneering work, providing foundational techniques that continue to influence new research. The keyword analysis highlights emerging themes, such as optimization algorithms and multi-objective optimization, signaling shifts in research focus that could dictate the direction of future studies.

3.1. Frequency of Publications

The review includes an analysis of the frequency of publications per year, providing insights into the trends and popularity of different optimization algorithms over time. This analysis helps in understanding how interest in various algorithms has evolved and can highlight periods of significant breakthroughs or shifts in research focus. An analysis of the number of publications per year shows a significant increase in research output from 2010 onwards, with a peak in 2020, as shown in Figure 1.

Figure 3 shows the temporal trends in both multi- and single-objective optimization algorithms published between 2000 and 2024. The provided data provide a comprehensive study of the number of publications, broken down by year and publication type (journals versus conference proceedings), in the areas of multi-objective algorithms, single algorithms, and related subjects. With a strong emphasis on recent years, notably 2020 and 2023, the dataset spans the years 2000 to the present. IEEE Access, Elsevier, and other IEEE Transactions are among the publications and conferences that are heavily represented in the dataset, indicating their significance as knowledge-sharing platforms in this subject. Moreover, Springer distinguishes itself as a noteworthy publisher that adds to the body of knowledge already available in the optimization algorithms field.

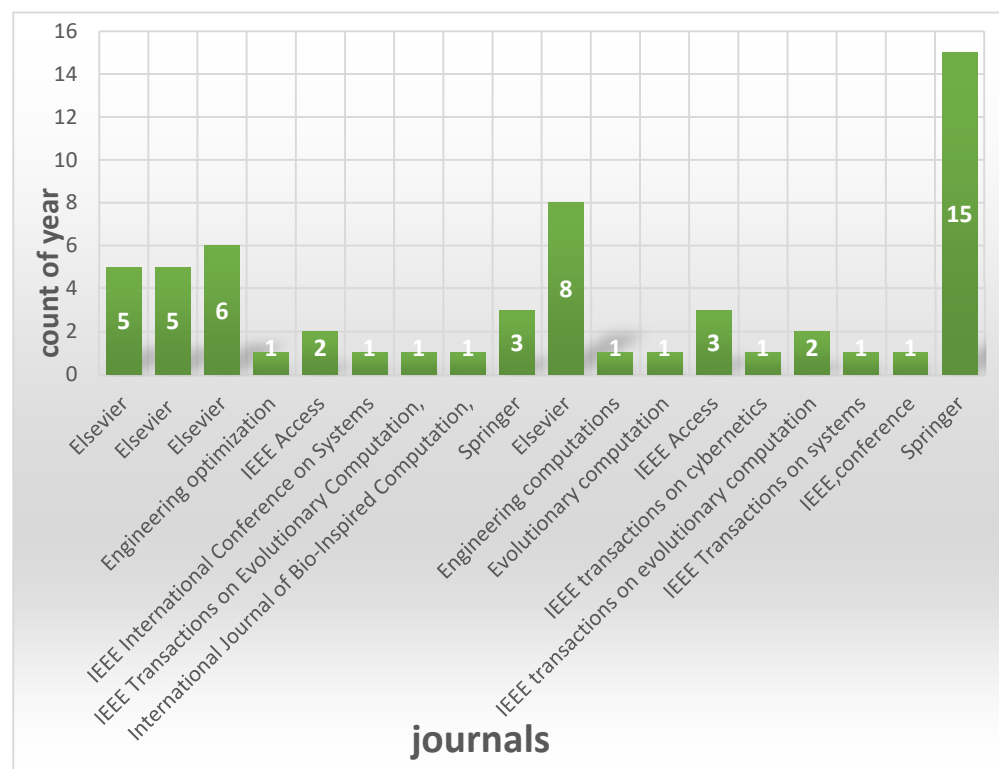


Figure 3. Temporal trends in both multi- and single optimization algorithms.

3.2. Types of Works Analyzed

The review includes a mix of scientific articles (60%), review papers (20%), and conference proceedings (20%). The analysis of academic publications spanning from 2019 to 2024, gathered from various academic journals and conferences. The chart provides a comprehensive summary of scholarly articles published during this period, encompassing a diverse array of subjects such as mathematics, computer science, engineering, and artificial

intelligence. Major contributing authors and their affiliations were identified, with a high concentration of research from institutions in the USA, Europe, and Asia, as shown in Figures 4 and 5.

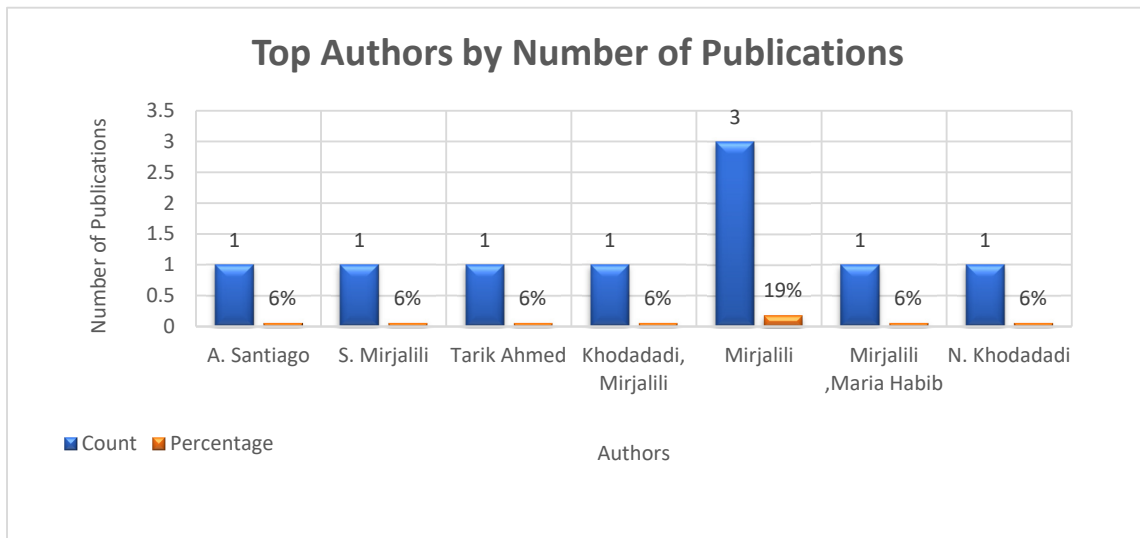


Figure 4. Analysis of publication trends: authors and chronological distribution in the field of single and MOO algorithm research.

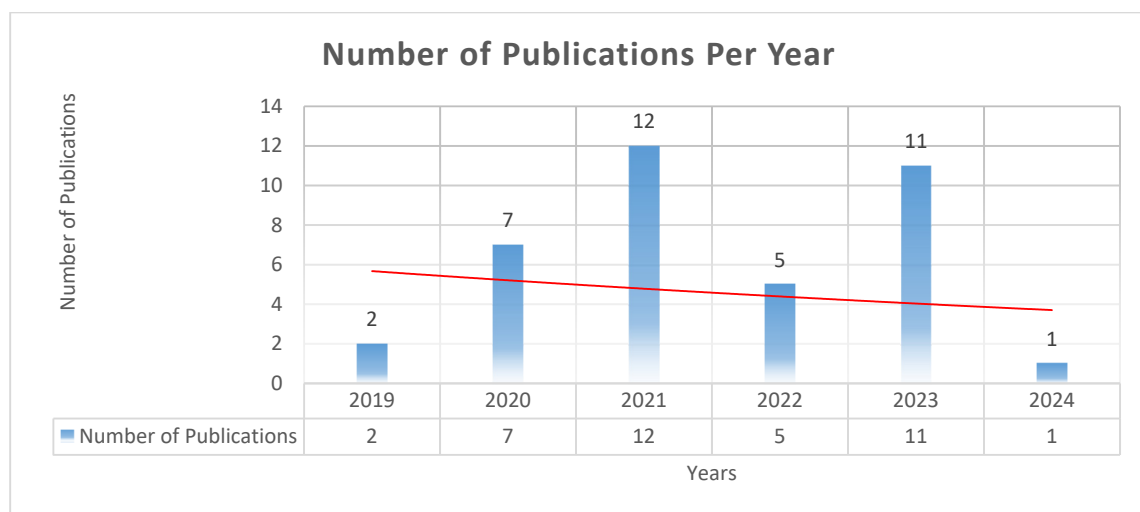


Figure 5. An analysis of academic publications across multiple journals and conferences spanning from 2019 to 2024.

3.3. Bibliometric Analysis

Bibliometric analysis was performed to identify key contributors, highly cited papers, and influential research groups within the field of optimization algorithms. This analysis provides a quantitative measure of the influence and impact within the academic community, helping to highlight the most significant contributors and their contributions to the development of optimization techniques.

3.4. Author Publication Analysis

Figure 4 depicts the allocation of publications among specific authors across different years, emphasizing their contributions to the area. Each bar in the diagram corresponds to the number of publications linked to a certain author in a specific year. Notably, authors such as S. Mirjalili stand out due to their numerous publications spanning various years,

which suggests a continuous and influential presence in the field. In recent times, authors like M. Abdel-Basset, R. Mohamed, Tarik Ahmed, and J.M. Abdullah have produced valuable contributions, indicating their active involvement in ongoing research efforts. S. Mirjalili consistently maintains a strong presence, with specific years, notably 2021, standing out as periods of increased scholastic activity. The data offer significant insights into the patterns of publication and the specific contributions of authors on the subject.

Figure 5 is an analysis of academic publications across multiple journals and conferences spanning from 2019 to 2024. This chart provides a thorough summary of scholarly articles published between 2019 and 2024, gathered from various academic journals and conferences. The publications encompass a diverse array of subjects, such as mathematics, computer science, engineering, and artificial intelligence. Each item provides the journal or conference name, publisher (if applicable), and the associated year of publication. The multitude of sources exemplifies the interdisciplinary character of the research field, emphasizing the changing patterns and regions of concentration within academia during this period.

Figure 6 illustrates the various types of optimization algorithms that could be implemented using single-objective or multi-objective methods.

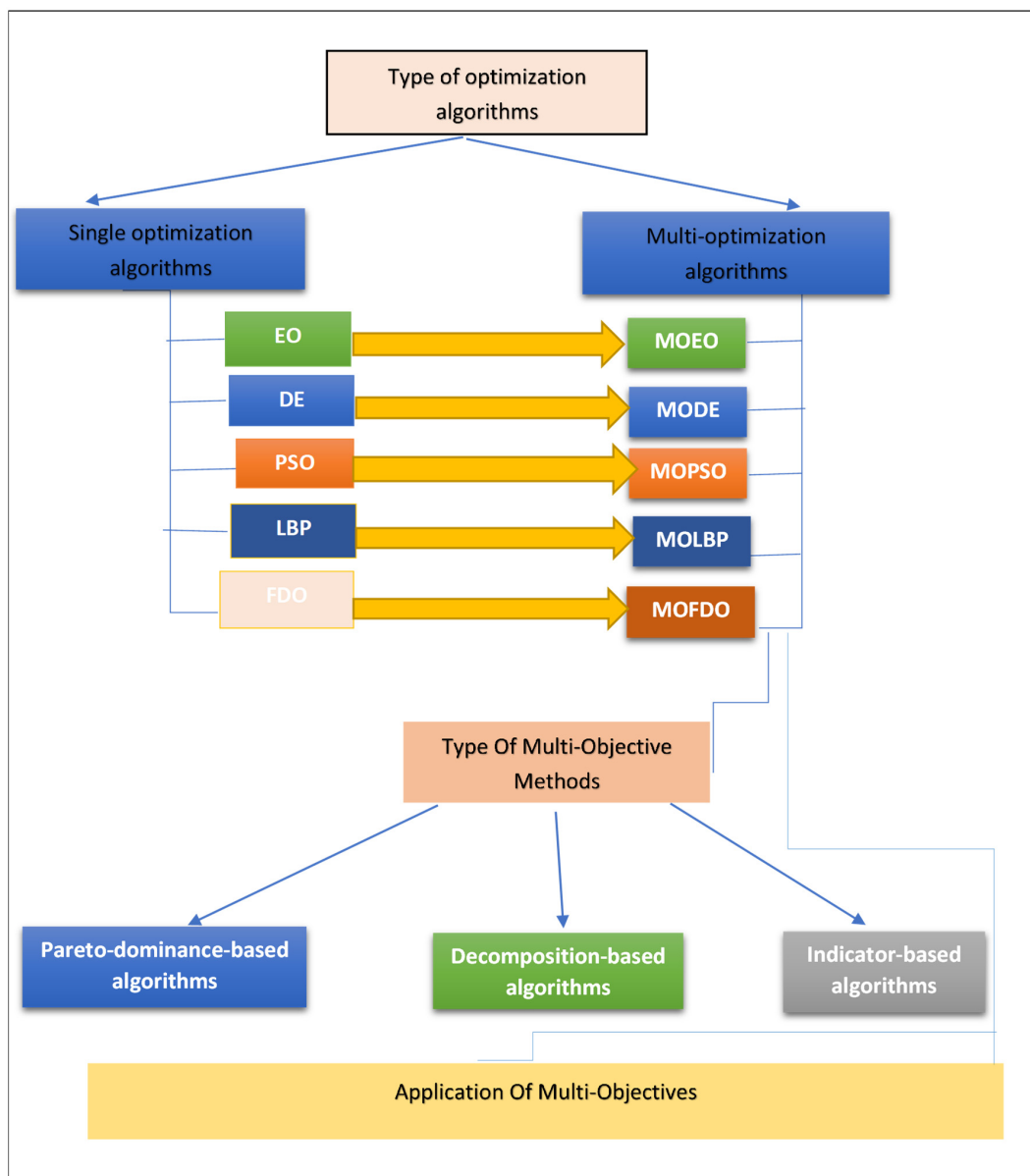


Figure 6. The type of optimization algorithm.

4. Analysis of Single Objective Algorithms Overview

4.1. Evolutionary Optimization

The primary ways that classical optimization approaches and evolutionary optimization concepts differ are explained in the following [8,50]. Gradient information is frequently not used by an evolutionary optimization technique during the search phase. EO methods are methods of direct search that are applicable to a large class of optimization problems. An evolutionary optimization process employs multiple solutions simultaneously in an iteration, contrasting with classical optimization algorithms that update only one solution per iteration. Utilizing a population has several benefits; an evolutionary optimization process employs stochastic operators, as opposed to deterministic operators commonly seen in traditional optimization methods. Operators aim to attain a certain result by favoring higher probabilities for favorable outcomes rather than relying on established and unchanging transition rules. EO algorithms can effectively navigate many optima and complications, offering a global perspective during the search process as seen in Algorithm 1.

Algorithm 1 EO

```
1: Start
2: Initialization population solutions randomly
3: Evaluated
4: For ( $i = 0, i \leq n, i++$ )
5: While (not termination condition stop)
6: Begin
7: Selection
8: Crossover
9: Mutation
10: Elite—preservation
11: End for
12: End while
13: Create new offspring
14: Update
15: End
16: Best optimal solution
```

The process of initialization frequently involves generating solutions at random. When generating the initial population, it is best to leverage information about known good solutions for a given problem. According to another source, a customized initialization can speed up the search process and be useful for handling complex real-world optimization problems [25]. To fill an intermediate mating pool, the selection operator selects solutions from the improved population that are above average and have a greater likelihood. For this aim, the literature on evolutionary optimization has a variety of stochastic selection operators. Essentially, two solutions could be randomly selected from the evaluated population, and the best of the two (depending on the assessed order) could be selected. This process is known as tournament selection. The “variation” operator is a set of operators which are used to produce a modified population, such as mutation and crossover. Through sharing information between the parent solutions, crossover operator creates one or more solutions through randomly selecting at least two solutions (i.e., parents) from the mating pool. When using the crossover operator, the fraction of the population that participates in the crossover operation is represented by the crossover probability ($pc \in [0, 1]$). The changed child population receives a direct replication of the remaining proportion of the population, which is equal to 1 minus pc . Every variable can be crossed separately in real-parameter optimization with n real-value variables as well as a crossover involving two parent solutions. Two new numerical values are often generated as offspring values positioned around the two parent values using a distribution of probability depending on the difference between values of the two parent variables. To preserve the relationship between variables

from parent solutions to the offspring solutions, vector-wise recombination operators are advised along with variable-wise recombination operators [1]. Every one of the child solutions is first created by a crossover operator and changed after that by a mutation operator [42,51]. With a chance of pm , which is often calculated as $1/n$ (the number of variables), each variable is mutated, leading to an average of one mutated variable per solution.

A simple Gaussian probability distribution with a set variance, centered on the value of the child variable, could be used for real-parameter optimization [1]. With the help of such an operator, an evolutionary optimization algorithm can investigate a solution's vicinity independently of other solutions' positions within the population. The operator of elitism combines the current population with the newly created population and selects the best solutions from the combined population. This process guarantees that an algorithm will always function at a high level without degrading. The authors of 90 proved that a specific EO technique, which includes elitism and mutation as important operators, has asymptotic convergence. Usually, a termination condition is applied after a particular number of generations. When faced with goal-related issues, an evolutionary algorithm could be terminated as soon as a target solution or solution that satisfies a predefined objective is found. Many studies compute the convergence rate using a termination criterion that compares the current population's statistics to those of the previous population. The satisfaction level of Karush–Kuhn–Tucker (KKT) conditions is one example of a theoretical optimality criterion that has been used in recent research to determine whether to terminate a real-parameter EO approach. While heuristics form the foundation of evolutionary algorithms (EAs), an EA's capacity to converge towards local optimal solutions could be assessed by applying theoretical optimality concepts to the algorithm [25].

4.2. Differential Evolution

Price and Storn (1995) proposed an evolutionary approach for solving problems inside a continuous domain. To balance exploration and exploitation, differential evolution modifies the search phase of the evolutionary process. The perturbation and differential are the two components that make up the mutation operator. The differential component directs other solutions toward improvement by using data from the population's best solution. On the other hand, random variation is introduced by the perturbation component and adapts throughout the evolutionary phase. Because parent individuals are so far apart, there is a great deal of disturbance early in the evolutionary process. As evolution proceeds, the disturbance diminishes and the population is constricted to a small area. By beginning with a large search step and subsequently refining the population with a smaller search step, the step of the adaptive search enables the evolution algorithm to carry out a global search. Depending on each parent and child's fitness scores, the selection operator in differential evolution selects the superior individual [15,16]. According to Feng, DE is similar to the (μ, λ) evolution technique, where a mutation is an important element. There are many iterations of the original differential evolution. The work of Sanderson and Joshi from 1999a and 1999b is followed by this specific one. The operators of mutation and selection are the main forces influencing evolution. There is a brief explanation in the paragraph that follows [16]. The process functions fundamentally in the same way as an evolutionary algorithm as seen in Algorithm 2.

Algorithm 2 DE

- 1: Initialization (initialize population with random members)
 - 2: Evaluated
 - 3: For ($i = 0, i \leq n, i++$)
 - 4: While (not termination condition stop)
 - 5: Begin
 - 6: Mutation calculate difference vector using Equation (1)
-

Algorithm 2 *Cont.*

```

7:   Recombine multi –point crossover using Equation (2)
8:   Replacement elitist replacement
9:   End for
10:  End while
11:  Create new offspring
12:  Update
13:  End
14:  Best optimal solution

```

After startup, each one of the members of the population experiences recombination and mutation. Following recombination, the new individual is assessed against the old individual, and the one with superior fitness is chosen to advance to the following generation (replacement policy). Among the current generation of members ($X_{i,G}$), we randomly select three distinct people from the present generation, ($X_{i,G}! = X_{r1,G}!, X_{r2,G}!, X_{r3,G}!$). We generate a mutant vector, or donor vector, using such member vectors, ($V_{i,G+1}$). In the following generation, the weighted difference of two vectors ($r3$ and $r2$) is added to the third vector ($r1$). Equation (1) illustrates this [51].

$$\left(V_{i,G+1} = X_{r1,G} + F(X_{r3,G} - X_{r2,G}) \right) \quad (1)$$

The new mutant vector finishes the mutation and is prepared now for recombination. Throughout such stage, multi-point crossover is carried out between mutant vector ($V_{i,G+1}$) and the original vector from the current generation ($X_{i,G}$), which involves incorporating components from our mutant vector with the original member for assessment of fitness. Equation (2) illustrates this procedure.

$$\left\{ \begin{array}{l} \text{if}(rand \leq CR) \text{ then } (V_{j,i,G+1}) \\ \text{if}(rand > CR) \text{ then } (V_{j,i,G}) \end{array} \right\} \quad (2)$$

After applying crossover to every vector element, the candidate vector that is produced is sent to the replacement phase. To contribute to the new vector, a random selection of one member and one mutant is made. Whether it is a new vector or one from the current generation, the vector with the greater level of fitness is permitted to advance to the following generation. To learn more, see [18,52].

4.3. Practical Swarm Optimization

PSO can be defined as a heuristic search approach that simulates movements of a swarm of birds seeking food. Its developers, refer to PSO as an evolutionary algorithm. Due to its population-based methodology and simple nature, PSO has been positioned as a good option for diversifying into MOO [6,40]. J. Kennedy and R. C. Eberhart first presented the PSO method for optimization. PSO is a population-based search algorithm that mimics the social behavior of a bird flock. PSO was originally developed as a global optimizer for neural network (NN) weight balancing [53]. It immediately became well liked, especially for problems requiring real number choice variables. There are two main distinctions between an evolutionary algorithm and PSO, according to Angeline. Parent representation, individual selection, and parameter fine-tuning are the three mechanisms that evolutionary algorithms rely on. In PSO, there are only two approaches that are used; an explicit selection function is not used. PSO uses leaders to guide the search to compensate for the absence of a selection mechanism. PSO is not related to the idea of offspring generation as an evolutionary algorithm.

The management of the individuals is another way that evolutionary algorithms and PSO differ from one another. Using an operator, PSO can determine a particle's velocity and give it a specific direction. This is a directional mutation operator that depends on

both the swarm's global best as well as the particle's personal best. There will be less investigation when the prospective directions have a modest angle in the case when the personal best direction coincides with the global best direction. On the other hand, a larger angle will enable more comprehensive research. Evolutionary algorithms use a mutation operator that can orient an individual in multiple ways, each with a different probability. Mutation operators similar to evolutionary algorithms have been adopted because of the directed mutation limitations of PSO [6]. There are two main characteristics that we believe have contributed to the popularity of PSO. First, the primary algorithm of PSO is simple due to its use of a single operator for generating new solutions, which sets it apart from most evolutionary algorithms, making its implementation uncomplicated. Moreover, a significant amount of PSO source code is accessible in the public domain. PSO has demonstrated high effectiveness across a diverse range of applications, yielding excellent outcomes with minimal computing expenses [12,53–55]. To learn more, see [6].

In PSO, particles move through a hyper-dimensional search space. Particle movements in the search space are driven by people's tendency, depending on social-psychological reasons, to imitate other people's success. Depending on its own experiences and those of its nearest neighbor, each particle adjusts its position using the equations in [6]. The position of particle p_i at time step t is represented by $X_i(t)$. Adding a velocity $V_i(t)$ to p_i 's current position modifies its position.

$$X_i(t) = X_i(t-1) + V_i(t) \quad (3)$$

The velocity vector, which is commonly defined as follows, depicts the information conveyed within society.

$$V_i(t) = W V_i(t-1) + c_1 r_1 (X_{pbest,i} - X_i(t)) + c_2 r_2 (X_{leader} - X_i(t)) \quad (4)$$

Random numbers between 0 and 1 make up c_2 , c_1 , r_2 , and r_1 . Particles are frequently affected by the accomplishments of those with whom they are associated. Instead of being close in parameter space, such neighbors are particles that are close to one another depending on a neighborhood topology that characterizes the social structure of the swarm.

For single optimization as seen in below Algorithm 3, the fundamental PSO algorithm functions. First, the swarm would be initialized. Velocities and locations are both initialized in this way. The leader is identified by choosing the gbest solution, and each particle's pbest is set. Each particle moves across the search space during a predetermined number of iterations, modifying its location in accordance with Equations (3) and (4), updating its own best, and after that updating the leader.

Algorithm 3 PSO

```

1: Begin
2: Swarm Locate leader Initialization
3: g = 0
4: While g < termination condition
5:   For every particle
6:     Update Position (Flight)
7:     Evaluation
8:     Update pbest
9:   EndFor
10:  Update leader
11:  g++
12: EndWhile
13: End

```

4.4. Fitness Dependent Optimizer

A population of artificial scouts is first randomly initialized inside the search space ($i = 1, 2, \dots, n$) through the algorithm; every point of a scout bee corresponds to a recently discovered hive (i.e., solution). In order to find excellent hives, scout bees randomly explore their surroundings. They ignore the hive they had previously found once they find a superior one. As a result, the previously found answer is ignored every time the algorithm finds a new and better one. Additionally, the artificial scout bee will continue on its previous trajectory in the hopes that it will lead it to a more ideal solution if its current movement fails to reach a better solution (hive). The current solution is the best choice found so far; if the previous path does not produce a better answer, the algorithm will go on to the current solution. In the wild, scout bees randomly look for hives. Using a random walk-in conjunction with a fitness weight mechanism, this system uses artificial scouts to first randomly scout the landscape. As such, when an artificial scout bee moves forward by speeding up from where it is now, it is doing so in order to explore a better choice. The following describes how artificial scout bees move:

$$X_{i,t+1} = X_{t,i} + pace \tag{5}$$

Assuming x is the artificial scout bee (search agent), t represents the current iteration, and i represents the current search agent, assuming $pace$ is the artificial scout bee's movement rate and direction. $Pace$ is mostly dependent upon physical fitness level. Nonetheless, the orientation of $Pace$ is dependent upon a stochastic mechanism. This is why it is possible to compute the objective function for minimization problems as follows:

$$fw = \left| \frac{*X_{i,t}fitness}{X_{i,t}fitness} \right| - wf \tag{6}$$

The fitness value of the optimal global solution discovered so far is denoted as the fitness function value $*X_{i,t}fitness$ of $X_{i,t}fitness$. The user's text is unclear and incomplete. The weight factor wf , whose value could just be 1 or 0, represents the fitness function regarding the current solution and is used to influence the fw . A minimum likelihood of coverage and a substantial degree of convergence are indicated if the value is equal to 1. On the other hand, if $wf = 0$, it has no effect on Equation (4) and can be ignored. The search is more stable when $wf = 0$. Because the fitness function value is dependent upon the optimization problem, there are times when the opposite occurs. Nonetheless, the fw value should be between 0 and 1. However, there are several situations in which fw could equal 1. This might happen, for instance, if the present solution is the global best or if the fitness value of the current solution and the global best solution are the same. Additionally, there is a possibility that the value of fw is equal to zero, which takes place in the case when the fitness of $*X_{i,t}fitness$ equals 0. In the case where X_i is equal to zero, it is crucial to avoid dividing by zero. Therefore, it is recommended to follow these guidelines:

$$\left\{ \begin{array}{l} fw = 1 \text{ or } fw = 0 \text{ or } fw = X_{i,t}fitness = 0 \text{ pace} = X_{i,t} * r \tag{7} \\ fw > 0 \text{ and } fw < 1 \left\{ \begin{array}{l} r < 0, \text{ pace} = (X_{i,t} - X_{i,t}^*) * fw * -1 \tag{8} \\ \{ r \geq 0, \text{ pace} = (X_{i,t} - X_{i,t}^*) * fw * \} \tag{9} \end{array} \right. \end{array} \right\}$$

within this context, r denotes a randomly generated integer that is between -1 and 1 . There are several ways to construct the random walk, yet Levy flight was chosen because of its well-distributed curve, which provides higher stability.

Fitness Dependent Optimizer Is a Technique Used to Solve Problems with a Single Aim in Optimization

The FDO with single-objective scout optimization problems (FDOSOOP) process begins with the placement of fake scouts at random positions on search landscape inside the designated higher and lower borders. The optimal solution discovered thus far is

selected as the global best solution at the end of every iteration. Next, Equation (2) is used for calculating the fitness value fw for each one of the artificial scout bees. The value of fw is then checked to see if it equals 1 or 0, and if it does, $xi, fitness$ also equals 0. The pace is produced using Equation (3). A random number r will be generated between -1 and 1 if fw is less than 1 and more than 0 . The speed is calculated using Equation (4) if the value of r is smaller than zero. fw is given a negative sign in this case. Equation (5) is used for calculating pace, however, if r is larger than or equal to zero. As a result, Zw is given a positive sign. A random assignment of a positive or negative sign to a fw will guarantee that the artificial bee searches in a random manner in every direction. In FDO, the randomization method influences both the direction and size of pace; in such cases, the pace size is dependent on the fw . In most other situations, the randomization procedure just impacts the pace direction. Moreover, the artificial scout bee uses the fitness function that has been shown in the pseudocode of single-objective FDO (refer to Algorithm 4) to determine whether the newly discovered solution is superior to the current one. The previous solution is ignored and the new answer is accepted if it is better. Moreover, a noteworthy feature of FDO is that the artificial scout bee will continue to pursue the previous path (using the previous $pace$ value if it exists) in the case when the new solution does not produce better results—that is, if it is beneficial. The scout bee is searching for a better solution. Moreover, the FDO method will stick with the current solution until the next iteration if using the previous position is not helping the scout bee discover a better one. With this method, the $Pace$ value is saved after each acceptable answer in case it is needed in a later iteration. The readers that are interested in learning more about single-objective FDO can refer to [20].

Algorithm 4 Single-objective FDO

```

1:   Initialize scout bee population randomly  $X_i$  ( $i = 1, 2, 3, \dots, N$ ).
2:   While (t) iteration limit not reached (m)
3:     or solution good enough.
4:     For every one of the artificial scout bee  $X_{t,i}$ 
5:       Finding the optimal artificial scout bee  $X_{t,best}$ 
6:       Generating random walk  $r$  in  $[-1, 1]$  range
7:       If ( $X_{t,i}$  fitness = 0)(for avoiding division by 0)
8:         Fitness weight = 0
9:       Otherwise
10:        fitness weight value is calculated with the use of Equation (6)
11:      EndIf
12:      If (fitness weight = 1 or the fitness weight = 0)
13:        Pace calculation with the use of Equation (7)
14:      Otherwise
15:        If (random number  $\geq 0$ )
16:          Pace calculation with the use of Equation (7)
17:          Pace calculation with the use of Equation (9)
18:        Otherwise
19:          Pace calculation with the use of Equation (8)
20:        EndIf
21:      EndIf
22:      Calculating  $X_{t+1,i}$  using Equation (5)
23:      If ( $X_{t+1,i}$  fitness <  $X_{t,i}$  fitness)
24:        Accepted and saved pace is moved
25:      Otherwise
26:        Calculate  $X_{t+1,i}$  with the use of Equation (5) with the preceding pace
27:        If ( $X_{t+1,i}$  fitness <  $X_{t,i}$  fitness)
28:          Accepted and saved pace is moved
29:        Otherwise
30:          The current position is maintained // (no move)

```

Algorithm 4 Cont.

```

31: EndIf
32: EndIf
33: EndFor
34: EndWhile

```

4.5. Learner Performance-Based Behavior

An innovative method of optimization is an algorithm depending on learner performance behavior. The LPB approach looks at student behaviors that affect their achievement in college by simulating the admissions process for different universities for recent high school graduates. The elements that could help students change their high school study habits that are ineffective for college-level work are also examined. In order to carry this out, multiple populations could be utilized to provide students with different GPAs. This ultimately results in a beneficial balance between exploitation and exploration. Depending on who in each category has the highest level of fitness, the population is divided into smaller groups. Subpopulations with the highest number of exceptional individuals are given precedence by the optimization technique. Newly formed individuals can have their structures changed using the crossover and mutation operators [21].

Learner Performance-Based Algorithm Steps

A population of the graduated learners M who want to randomly apply to a variety of departments in various universities is created as the first step in the algorithm. We also have an operator called division probability (dp). As stated in [21], learners with GPAs that meet or exceed the required minimum are admitted to all departments. Using the dp option, we first choose a random subset of elements from M according to a specified percentage. Next, we rank each of the chosen individuals based on their level of fitness. After that, we divide them into two groups: those with low fitness levels and those with high fitness levels. Those with a higher GPA are in the former category, and the other people are in the latter group. Then, the fitness of individuals of the main population M is calculated and further refined. People will be moved to the bad population in the case where their level of fitness is lower than or equal to the highest level (optimal level) in the bad population. The rest of the individuals will be divided into two groups. The ones who are either less fit than or as fit as the most fit in the good population will be transferred to a good population. The ideal population will receive those whose fitness values are higher than the maximal fitness in the ideal population. The department will then choose the predetermined number of learners from the good and ideal populations. There will be a minimal number of learners who meet the needed GPA if the population size of these two groups is less than the number of the specified students as has been decided by the department. In such cases, the department has to decide whether to accept more learners with lower GPAs or not. If they decide to accept more learners, the remaining applicants will be chosen from the disadvantaged population. As previously mentioned, it is clear that recent high school graduates might not have effective study habits when they are admitted to the departments [56].

Nevertheless, enhancing behaviours such as requesting assistance and collaborating in groups can greatly benefit individuals. Furthermore, as previously stated [56,57], individuals can exert an impact on the actions and conduct of their peers. For instance, when individuals collaborate in groups or seek assistance from one another, their studying behaviours will be influenced. The genetic algorithm employs the crossover operator to demonstrate this. Implementing a crossover operator enables individuals to interchange certain learning behaviours. As a result, the learner possesses a distinct set of studying behaviours that differ from the original studying behaviours of other learners. Consequently, the behaviours of both individuals will be influenced, resulting in the production of individuals with distinct behaviours. A learner's overall study behaviors are also strongly influenced by their level of metacognition, which is a sign of significant influence. A

learner's general study behaviors will be affected when their metacognitive capacity is randomly impacted [56,57]. As per [57], the learner's level of metacognition is impacted by the many methods they are trained in. This research does not involve the implementation of such solutions. As a result, the system could determine the rate at which learners' metacognition changes. As mentioned before, learners' behaviors may be randomly influenced by their level of metacognition. This could be achieved by updating the values of the learner's studying behaviours at random or by randomly changing the individual's behavior positions at a specific rate. The algorithm uses the genetic algorithm's mutation operator; for further information on the algorithm, see [21]. The LPB algorithm's pseudocode is shown in Algorithm 5.

Symbol definition:

M represents the initial random population

N represents the number of individuals in the new population

dp represents the percentage of the individuals that have been selected from M

O represents a subpopulation that has been selected from M based on dp operator.

GP represents the good population

BP represents the bad population

k represents a counter that is utilized for counting the number of individuals that have been newly created

PF represents the perfect population

Algorithm 5 LPB

```

1: [Initialization] A population M is randomly created
2: [parameters are specified]
3: The number of the required learners N for some departments, mutation rate and crossover rate are specified.
4: [Subpopulations are Created]
5: The dp parameter is utilized for the random selection of a percentage of the individuals O from M. The fitness of the individuals in O is evaluated based on their level of fitness, individuals are sorted in O (i.e., in a descending manner), one of the sorting methods is used, and O is divided into 2 good populations (individuals that have a high level of fitness) and bad (individuals that have a low level of fitness).
6: While the condition of termination isn't met
7: The dp parameter is utilized for randomly choosing a percentage of the individuals O from M
8: Individuals' fitness is assessed in O. Depending upon their level of fitness, individuals are sorted in O (in a descending manner), and one of the sorting approaches is used.
9: The group of O is divided into 2 populations, which are good (individuals that have a high level of fitness) and bad (individuals that have a low level of fitness)
10: Fitness is found for all of the individuals in M
11: Highest fitness is found in the bad as well as good populations
12: In the case where an individual from M has fitness  $\leq$  highest fitness in bad population
13: Then it is moved to BP
14: ElseIf
15: an individual from M has fitness  $\leq$  highest fitness in GP
16: then it is moved to GP
17: Else
18: It is moved PF
19: EndIf
20: while  $k \leq N$ 
21: in the case where PF isn't empty an individual is selected from PF

```

Algorithm 5 Cont.

```

22: ElseIf
23: GP isn't empty
24: Then an individual is selected from GP
    25: Else
    26: An individual is selected from BP
    27: EndIf
28: K++;
29: EndWhile
30: Crossover
31: Mutation
32: [Termination] the process is repeated from step 3 through termination condition
    is met.
33: EndWhile
34: [Optimal Solution] The optimal solution is selected from the perfect population.
    
```

4.6. A Comprehensive Analysis of the Benefits, Drawbacks, and Practical Uses of Single Algorithms

Within the optimization field, individual algorithms are essential tools with unique strengths and weaknesses. This section comprehensively analyzes the EO, DE, PSO, FDO, and LBP single algorithms, highlighting their advantages, disadvantages, and practical uses in various real-world optimization issues. Table 2 presents a thorough examination of individual algorithms, offering a full analysis of their benefits, limitations, and practical applications in different optimization scenarios. This table provides readers with significant insights into the effectiveness and relevance of different algorithms in solving complicated real-world problems by carefully examining their unique characteristics.

Table 2. The advantages, disadvantages, and applications of the single algorithms.

Single Algorithms	Advantage	Disadvantage	Applications
EO	<ul style="list-style-type: none"> - It provides an EO with parallel processing power that achieves computationally fast overall search - It provides an EO to find several optimal solutions, thus leading to the facilitation of MOO and multi-modal problem solution - It presents the EO with the capability of normalizing decision variables (in addition to constraint and objective functions) within the evolving population with the use of population-best min. and max. values. 	<ul style="list-style-type: none"> - Slow convergence speeds - Premature convergence 	<ul style="list-style-type: none"> - It is used for solving a very large number of real-world optimization tasks [8].
DE	<ul style="list-style-type: none"> - Efficiency, simplicity, and real coding, easy to use - Works with 2 populations, which are new generation and old generation of same population. - The population size is adjusted by NP parameter. - Local searching property and speediness 	<ul style="list-style-type: none"> - Slow convergence speeds - Premature convergence - Instability performance 	<ul style="list-style-type: none"> - It is used for solving a very large number of real-world optimization problems [16,18].

Table 2. Cont.

Single Algorithms	Advantage	Disadvantage	Applications
PSO	<ul style="list-style-type: none"> - Has high effectiveness in many different applications, which have the capability of producing excellent results - Relatively simple to implement - Very low computational - Low cost 	<ul style="list-style-type: none"> - Prone to falling into the local optimum 	<ul style="list-style-type: none"> - Used for solving a very large number of real-world optimization problems [54,58,59].
FDO	<ul style="list-style-type: none"> - Using a fitness function for the generation of proper weights that are helpful for the algorithm in the exploration phase as well as the exploitation phase - It has important feature in which the past search agent pace (i.e., velocity) is stored for possible re-use in future - Similar to PSO mechanism in updating agents' position - Fast convergence to global optimal - Proven effectiveness to solve real-world problem 	<ul style="list-style-type: none"> - Consuming more time and space - The performance of the algorithm depending on the number of search agents 	<ul style="list-style-type: none"> - Aperiodic antenna array designs - Frequency-modulated sound waves - GAP, which stands for Generalized Assignment Problem, is a well-known combinatorial optimization problem that is classified as NP-hard [20].
LPB	<ul style="list-style-type: none"> - Provides good balance between the phases of exploitation and exploration - Convergence speed - The ability to avoiding local optima better than other algorithms such as PSO - Using mutation and crossover to make good change in the structure of individual - Proven effectiveness to solve real-world problem - Improving the randomness and time processing - A percentage of population is separated. - Population is divided into multiple subpopulations. 	<ul style="list-style-type: none"> - Sizes for combinatorial optimization 	GAP [21].

5. Analysis of Multi-Objective Algorithms

This discussion will only address MOO among the several kinds of optimizations [25]. In MOO, the best values for several desirable objectives are found. Given that many real-world problems lend themselves to modeling with many competing aims, it has great practical value (Ming, 2019; Arora, 2004; Jain and 2019, Eriksson) [60–62]. MOO is used because it simplifies the optimization process by doing away with the necessity for intricate formulae. The capability to come to a compromise or trade-off on opposing issues is a difficult decision-making task in MOOs. Numerous difficult applications of multiple objective problems exist, including staff scheduling (Clarke, 2015) [63], workshop scheduling (Huang et al., 2018) [64], transportation challenges (Samy and Elkhoully, 2021) [65], energy efficiency, and other applications [66]. A thorough examination of the current multi-criteria decision-making models and their real-world applications is given in the paper by Pereira

(2022) [67]. The prevailing approach to such problems typically focuses on combining many goals into a single objective and takes group dynamics into account [41].

5.1. Multi-Objective Basic

It was Vilfredo Pareto who invented MOO. A vector that represents the objective function is contained in MOO. The vectors of the goal function rely on the vector of the solution. There are various options that could be taken into consideration within the context of MOO, rather than a single, universally applicable best solution [68]. The MOO problem's mathematical representation is expressed in Equation (10) (Deb, 2001) [8].

$$\begin{aligned} &Max/Min f_1(x), f_2(x), f_3(x), f_4(x) \dots \dots \dots f_n(x) \\ &Subject\ to: X \in U \end{aligned} \tag{10}$$

Assuming U is the feasible set, $f_n(x)$ is the n to nth objective function, x represents the solution, n represents the number of the objective functions, and min/max represent the total number of object operations [8]. The objective function vector as well as the choice variable space of the solution vector are both contained in the multidimensional space that makes up the MOO. There is a matching point in the space of the objective function for every solution x in the space of choice variables (Deb, 2001) [1,8]. Since we are only concerned with maximizing one function, finding the best solution in the setting of a single-objective problem is simple. If and only if solution A achieves a smaller objective value than solution B in a minimization problem, then solution B is preferable to solution A. In MOP, on the other hand, we seek to find a solution that concurrently maximizes each of the objectives. The relational operators cannot be used to compare the solutions. Unfortunately, finding a single solution capable of optimizing all the target functions at the same time is a difficult undertaking [39] (Mohamed Abdel-Basset). If and only if solution A achieves a higher value in at least one of the objective functions, whereas the values of other objectives stay the same, then solution A is deemed superior or dominant to solution B in the context of MOP. Given that it denotes the perfect balance between the objectives, the Pareto-optimal solution is the best possible solution for a multi-objective problem [15,69,70].

5.2. Multi-Objective Methods

Evolutionary algorithms are used in the MOO paradigm. The research topic of evolutionary MOO [67,71,72] encompasses the theoretical framework, real-world applications, and computational techniques. A variety of multi-objective evolutionary algorithms that were particularly created to solve MOO problems are described in the literature. Several characteristics allow for the classification of these algorithms. The following groups make up one often-used classification for multi-objective evolutionary algorithms [73,74]:

- (a) Pareto-dominance-based algorithms: NSGA has been used to find the Pareto front for MOO problems since its invention in 1994. Following this, several intelligent optimization algorithms that relied on Pareto-dominated methods surfaced. Non-convex optimization problems were handled through intelligent algorithms that preserved each goal's distinctive qualities on their own, without relying on the others. A growing number of studies on intelligent optimization algorithms have been carried out by researchers to offer Pareto-dominated solutions for MOO problems. The Pareto dominance relationship is applied, which entails choosing a partner from the population of individuals that it dominates for non-dominated individuals. Amongst well-known algorithms of this kind are Pareto Envelope-based Selection Algorithm II (PESA-II) [25,74], Strength Pareto Evolutionary Algorithm 2 (SPEA-2) [75], and Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [42,76].
- (b) Decomposition-based algorithms transform MOP into a set of SOPs utilizing the scalarizing functions: After that, concurrent solutions are found for these resultant single-objective problems. MOGLS (i.e., multi-objective genetic local search algorithm) [77], the cellular multi-objective genetic algorithm (C-MOGA), and the multi-

objective evolutionary algorithm based on decomposition (MOEA/D) [2] are a few examples of algorithms of that method.

- (c) Indicator-based algorithms utilize indicator function for the assessment of a set of solutions' quality through utilizing a measure that takes into account the objective function space's convergence and diversity. Those algorithms' objective is to determine, depending on the performance indicator, the optimal subset of the Pareto non-dominated solution. There are other variations of such algorithms, including the Fast Hypervolume MOO Algorithm, S-Metric Selection Evolutionary MOO Algorithm (SMS-EMOA), and Indicator Based-Selection Evolutionary Algorithm (IBEA) [78].

5.3. Multi-Objective Algorithms Overview

There are several objective functions in a MOO problem that need to either be maximized or minimized. Similar to a single-objective optimization problem, any conceivable solution—including all of the optimal solutions—must fulfil several constraints in the MOO problem. Objectives could be maximized or minimized, respectively. The extremist concept of seeking an optimum solution cannot be applied to a single objective in the context of MOO when other objectives are significant as well. Trade-offs—conflicting consequences between aims—between various objectives might result from different solutions. A compromise must be made about other objectives to achieve an extreme (in a better sense) solution for one goal. One cannot select a solution that is ideal for only one objective as a result. This indicates two optimal MOO goals [8]:

1. Find a set of solutions lying on the Pareto-optimal front;
2. Find a set of solutions diverse enough to represent the whole Pareto-optimal front range.

5.3.1. Evolutionary MOO

The aforementioned concepts are what evolutionary MOO approaches seek to uphold. From a practical standpoint, an optimization issue, whether multi- or single-objective, only needs one solution, even though multiple- and single-objective optimizations differ in the cardinality of the optimum set. Finding a set of ideal solutions that successfully balance each objective is the aim of MOO. When a user finds several trade-off options, they can employ more sophisticated qualitative factors to help them decide. To identify a group of non-dominated solutions in MOO, EMO approaches are appropriate since they manage a population of solutions in every one of the iterations [8,25]. When handling MOO problems, an evolutionary multi-objective technique functions based on the following notion [25]:

Step 1: Identify many non-dominated places along the Pareto-optimal front that exhibit broad trade-offs across the objectives.

Step 2: Select one of the acquired points based on more advanced data.

Figure 7 roughly illustrates the ideas used in an EMO approach. EMO processes, being heuristic, do not ensure the discovery of Pareto-optimal points, unlike theoretically proven optimization methods designed for situations that are linear or convex.

Significant operators are employed by EMO approaches for improving evolving non-dominated points, emphasizing both convergence and variety, much like how artificial and natural evolving systems constantly enhance their solutions. Given that, one of the recent simulation studies [1,8] showed that starting from random non-optimal solutions, a specific EMO method could iteratively progress toward the theoretical Karush–Kuhn–Tucker (KKT) points in real-value MOO problems. Finding many trade-off solutions in a single simulation is one benefit of using an evolutionary MOO technique. Figure 6 shows the identification of many non-dominated points reflecting trade-offs vertically downward during Step 1 of the EMO-based MOO method. In Step 2, one of the trade-off points resulting from the task displayed horizontally to the right is chosen using higher-level information.

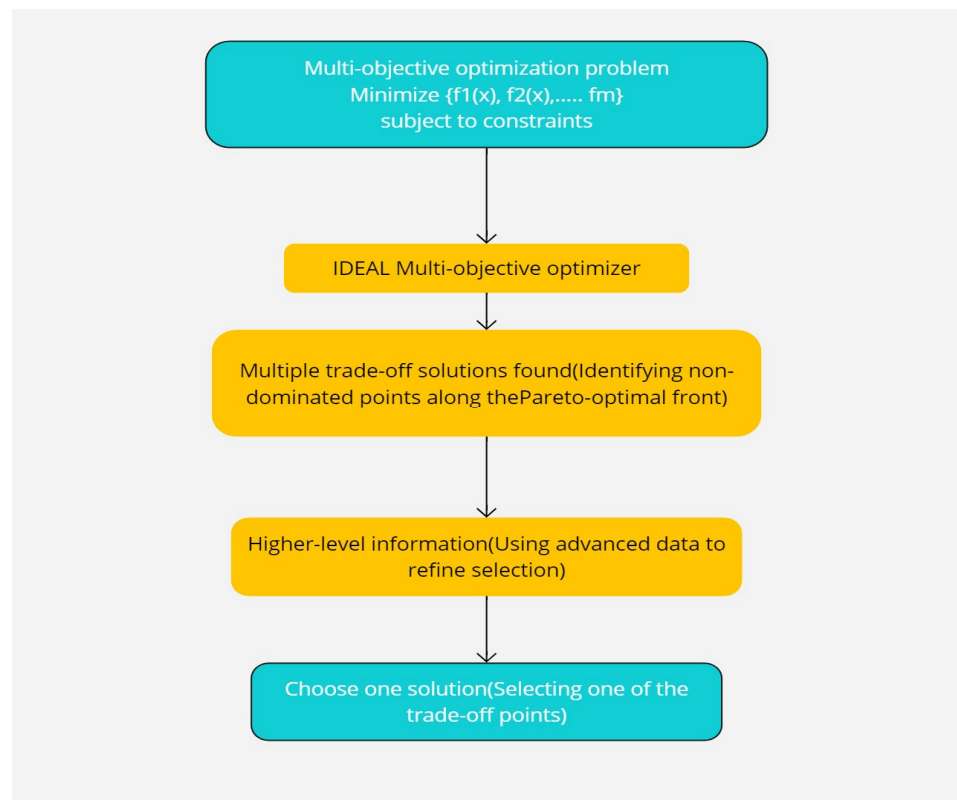


Figure 7. Diagram illustrating a two-step MOO process.

In this case, when applying this dual-tasking capability to single-objective optimization tasks, it becomes intriguing. According to detailed explanations in other studies, a single-objective optimization is a degenerate version of MMO [8]. Ideally, Step 1 should find the unique globally optimal solution in single-objective optimization, such that Step 2 is not necessary. It is crucial to first identify all or a few global optima in single-objective optimization with numerous global optima before choosing a solution based on more in-depth knowledge of the issue. Although it seems specifically designed for MOO, Figure 7's framework can be thought of as a universal principle that applies to both single- and MOO. By concentrating on many non-dominated and unique solutions, an EMO seeks to find multiple Pareto-optimal solutions inside a single simulation. Later, we will look at a few EMO techniques that clarify how this dual emphasis is accomplished. The NSGA-II approach is a popular EMO method which finds many Pareto-optimal solutions to an MOO problem [42]. It has the following three characteristics:

1. It employs an elitist concept.
2. It employs a specific strategy to maintain diversity.
3. It highlights non-dominated solutions.

Applying crowded sorting on the points in the final front yields the points with the largest diversity. Points are selected starting at the top of the list and going down in descending order based on their crowding distance values. The unoccupied objective space surrounding point i in the population is measured by the crowding distance d_i of point i . We compute d_i by estimating the crowding distance, which is the perimeter of the cuboid formed with the use of the closest neighbors as vertices in the objective space.

5.3.2. Multi-Objective Differential Evolution

As a general evolutionary algorithm, the MODE method is made up of three primary parts: selection, Pareto-based evaluation, and mutation. This section [16] goes into great detail about these elements. Algorithm 6 describes the steps of MODE

Mutation Operator

Two types of vectors should be established: the perturbation vector and the differential vector, to duplicate the mutation operator in the differential evolution method that was described in Section 2. For a single-objective problem in the DE, the vector that depicts the difference between the best individual and the individual undergoing operation is known as the differential vector. The individual with the highest value of fitness within the population is usually the best [51]. The Pareto-optimum solution group is the goal of the evolutionary approach when applied to a multi-objective domain. The original objective of finding every Pareto solution is to identify them all, not to direct individuals toward a single option. In every generation of the evolutionary process, the proposed MODE presents a Pareto-based method for choosing the best individual for mutation procedure identification regarding non-dominated solutions (Pareto-optimum solutions) in D across the population. We should determine whether an individual P_i is dominated before performing the mutation operation on them. We need to look into whether or not that specific individual is being dominated. The “best” answer for identifying a dominated individual is to identify the group of non-dominated individuals, D_i , that dominate this individual. A value is chosen at random from the set D_i , called P_{best} . For the mutation process, the vector that lies between points P_{best} and P_i is the differential vector. P_{best} will be the individual itself if they are already non-dominated. In this case, only perturbation vectors are affected and the differential vector equals zero. In contrast to single objective DE, multi-objective differential evolution changes rather than stays the same to reproduce all of the individuals in a population. This is consistent with the main objective of finding the entire Pareto-optimal set. A random selection of individual pairs from the parent population yields perturbation vectors. A structure akin to the single-objective DE could be used for the mutation process once differential vectors as well as perturbation vectors are determined. If one uses the representation of the natural chromosome, each allele is associated with a single choice variable. They conduct the experiments detailed in their study using this form of representation in their implementation of MODE. So, given a mutation probability of P_m , each allele of an individual will undergo a specific process. An individual could be thought of as a vector, and every element of the vector will be subject to the operation, as shown in Equation (11) [16].

$$\mathcal{P}_i = \begin{cases} P_i + F * \sum_{k=1}^K (P_a^k - P_b^k) & \text{if } P_i \text{ is non-dominated} \\ \gamma * P_{best} + (1 - \gamma)P_{best} + F * \sum_{k=1}^K (P_a^k - P_b^k) & \text{else} \end{cases} \quad (11)$$

where P_{best} is the best Pareto-selected individual from the parent population, $\gamma \in [0, 1]$ denotes the operator’s greediness, K is number of perturbation vectors, F is the factor that determines the perturbation’s scale, P_a^k and P_b^k are mutually distinct individuals from the parent population that are randomly selected, and \mathcal{P}_i is the offspring.

The boundary used to determine how many individuals are inside the predefined sharing radius is the radius. An individual’s fitness value is penalized more heavily as the number of individuals in their border grows. The difficulty is in figuring out the right fitness-sharing radius, which is contingent on the properties of the problem’s objective or the choice space [16,18].

Selection Operator

A $(\mu + \lambda)$ selection method is used in the original NSGA-II, where parents and offspring are combined in order to compete for membership in the following generation. One of the approaches used in this manner is to keep the elite in place while the candidates are being chosen. Individuals are first compared according to their Pareto rankings. Higher-ranking individuals are selected for the following population. In the event of a tie in Pareto ranks, the individuals who will make up the next generation of the population are ascertained using the crowd distance metric. Research by Xue et al. (2003) [16] has demonstrated that using a strong elitism approach does not produce desirable results. Any

solution that ranks higher will be chosen using this fitness evaluation and selection method, regardless of how dense it is within that rank. Goel and Deb (2001) [79] underline the importance of retaining diversity within ranks to allow those in lower positions to advance to the following generation.

To determine the solution's proximity to nearby solutions in the objective space and ultimately reducing its fitness to a minimal value, the MODE incorporates an additional parameter called σ crowd. This method might prevent individuals who are closely related from reproduction, which could lead to early convergence. To put it simply, the MODE creates the following generation by selecting the top N individuals solely based on their fitness rating from parents as well as offspring that have been created by the reproduction operator. When compared with fitness sharing algorithms presented by Goldberg (1989) [50], it is possible to compute crowd distance metrics without the specification of any parameters beforehand. In the proposal, a new parameter (σ crowd) is proposed. Since the major objective of MOEA is to avoid very similar individuals, the sensitivity of MODE is not as important as the sharing radius in terms of how well it functions. In studies, a low value of (σ crowd) is advantageous in many cases [15].

Algorithm 6 MODE

```

1: Initialization (initialize population with random members)
2:   Evaluated
3:   For( $i = 0, I \leq n, i++$ )
4:   While (not termination condition stop)
5:   Begin
6:     Implement a based method for the selection of optimal individual
7:     For the mutation operation using Equation (11)
8:     Apply Pareto rank assignment to evaluate individuals
9:     Using a sharing radius as a boundary to count members of the individuals
10:    Using crowding distance metrics for estimating the density of the solutions
11:    around certain individuals and panelized fitness of the individual
12:    selection operation
13:    to reduce the fitness of the solution to a very small value used parameter  $\sigma$  crowd
14:    based only on fitness ranking for the selection of optimal  $N$  individuals as the following generation from the two parents
    and offspring that have been produced by the operator of reproduction
15:   End for
16:   End while
17:   Create new offspring
18:   Update
19: End
20: Best set of solutions

```

5.3.3. Multi-Objective Particle Swarm Optimization

In Section 4.3, we noted that, in contrast to global optimization, the solution set of a problem with multiple objectives is not a single solution, indicating that the original approach should be modified to apply the PSO method for tackling MOO problems. Finding several solutions known as the Pareto-optimum set is the aim of MOO. When addressing a multi-objective problem, there are usually three main objectives to emphasize:

1. Increase the quantity of elements in the Pareto-optimum set.
2. Reduce the difference between the algorithm-generated Pareto front and the actual, global Pareto front (assuming we are aware of its position).
3. Maximize the dissemination of solutions to achieve a smooth and uniform distribution of vectors.

As seen in the preceding section, once a neighborhood topology is created, each particle's leader for updating its position is identified while solving single-objective optimization problems. Every particle in MOO problems might have several leaders to pick from, yet just one could be chosen to update the particle's location. Usually, the leaders' group is kept

in a location that is referred to as an external archive that is distinct from a swarm [12,58]. The non-dominated solutions that are discovered are kept in this repository. The solutions that are kept in the external archive act as a reference to rearrange the particle positions within the swarm. Furthermore, the final result of the algorithm is usually supplied as the contents of the external archive [54,59]. Nothing in the way of material is offered. The workings of a general MO-PSO algorithm are demonstrated by Algorithm 7. The distinct features that set this algorithm apart from the conventional PSO method for single-objective optimization are highlighted by the underlined procedures. First, the swarm is organized. The non-dominated particles in a swarm are used in order to form a group of leaders. As mentioned before, the group of leaders is usually stored in an external repository. After that, a quality metric is computed for every leader to choose one leader for every swarm particle. Every particle has a leader selected throughout each generation, and the flight is carried out. A mutation operator is incorporated into many MOPSO algorithms after the flight operation. The particle is then evaluated, and the corresponding personal best value is updated. In the case when the former is either dominated or both particles are incomparable, which means they are not dominated concerning one another, a new particle frequently replaces its personal best particle. The set of leaders is likewise updated when all particles are updated. This approach is usually repeated for a specific and updated quality measure of the group of leaders at the end [6,80].

Algorithm 7 MOPSO

```

1:  Begin
2:  Swarm Initialization
3:  Initialization of leaders in external archive
4:  Quality(leaders)
5:  g = 0
6:  While gmax > g
7:  For every particle Select leader
8:  Update Position (Flight)
9:  Mutation
10: Evaluation
11:  pbest Update
12:  EndFor
13: Update leaders in external archive Quality(leaders)
14:  g = g + 1;
15: EndWhile
16: The report leads to external archive
17: End
  
```

5.3.4. Multi-Objective Fitness Dependent Optimizer

In MOFDO, in order to determine the pace, it is necessary to take into consideration conditions that are outlined in Equations (13)–(15), which rely on the fitness weight (fw) value, which may be determined by applying Equation (16) to the problem's cost function values. There is importance in noting that in MOFDO, the pace denotes both domain and historical knowledge:

$$X_{i,t+1} = X_{t,i} + \text{pace} \quad (12)$$

In this case, x denotes the individual itself, t denotes current iteration, i represents current individual number, and pace is the direction and rate of movement, remembering that fw determines the pace value most of the time. On the other hand, the pace's direction (value sign) is solely determined randomly.

$$\left\{ \begin{array}{l} fw = 1 \text{ or } fw = 0 \text{ or } fw = X_{i,t} \text{ fitness} = 0 \text{ pace} = X_{i,t} * r \text{ (13)} \\ fw > 0 \text{ and } fw < 1 \left\{ \begin{array}{l} r < 0, \text{ pace} = (X_{i,t} - X_{i,t^*}) * fw * -1 \text{ (14)} \\ r \geq 0, \text{ pace} = (X_{i,t} - X_{i,t^*}) * fw * \text{ (15)} \end{array} \right. \end{array} \right\}$$

Equations (13)–(15) contain two different conditions. Initially, if fw equals 0, or, if fw equals 1, or if $\sum_{o=1}^n Xi,tfitness = 0$, then the pace should be estimated as Equation (12). Second, a random number r is created in the $[-1, 1]$ range if the fw value is between zero and one. If r is negative, Equation (13) will be applied; if not, Equation (15) will be utilized in order to determine the pace fw , which is calculable with the use of Equation (16).

$$fw = \left| \frac{\sum_{o=1}^n *Xi,tfitness}{\sum_{o=1}^n Xi,tfitness} \right| - wf \quad (16)$$

where $\sum_{o=1}^n *Xi,tfitness$ represents a summation of cost function of the global best individual, n represents the number of objectives, and $o = \{1, 2, 3, \dots, n\}$; the $\sum_{o=1}^n Xi,tfitness$ represents the summation of current individual's cost function, again n represents number of the objectives, and $o = \{1, 2, 3, \dots, n\}$. Lastly, the weight factor wf in Equation (16) has a value of either 0 or 1. It is possible to observe that $wf = 0$ has no effect on the equation and can be disregarded. Readers who are interested in learning more about single-objective FDO are directed to [20]. The algorithm structure of MOFDO as seen in Algorithm 8 is somewhat similar to that of a single objective FDO, but it also has a few more advantages, which are as follows: for the goal of holding Pareto front solutions throughout optimization, an archive (repository) is utilized, since this is the practice of much published research [73,81]. Pareto front solutions undergo a polynomial mutation prior to the non-dominated solution being added to the archive. As a variation operator, the polynomial mutation was used in MOEAs [67].

Algorithm 8 MOFDO

- 1: Initialize the artificial scout bee population randomly $X_{t,i}$ ($i = 1, 2, 3, \dots, n, t = 1, 2, 3, \dots, m$).
 - 2: Creating an archive for non-dominated solutions with specific sizes.
 - 3: Generate Hybrid cube Grid
 - 4: While (t) iteration limit not reached (m) or solution good enough.
 - 5: For every one of the artificial scout bee $X_{t,i}$
 - 6: Find the ideal artificial scout bee $X_{t,best}$
 - 7: Generation of the random walk r in $[-1, 1]$ range
 - 8: Estimate weight fitness value Using Equation (14) //checking Equations (12)–(14) conditions
 - 9: if (fitness weight ≥ 1 or fitness weight ≤ -1 or summation of $X_{t,best}$ fitness = 0)
 - 10: fitness weight = r
 - 11: EndIf
 - 12: Calculate $X_{t+1,i}$ using Equation (12)
 - 13: If ($X_{t+1,i}$ fitness dominate on $X_{t,i}$ fitnesses)
 - 14: Otherwise
 - 15: Calculation of $X_{t+1,i}$ using Equation (12) with previous pace saved
 - 16: If ($X_{i,t+1}$ fitness dominates on $X_{i,t}$ fitnesses)
 - 17: Move accepted, and saved pace
 - 18: Otherwise
 - 19: The current position is maintained // (do not move)
 - 20: EndIf
 - 21: Apply polynomial mutation
 - 22: Add non-dominated ants (solutions) to the archive.
 - 23: Keep only non-dominated members in the archive
 - 24: Update Hypercube Grid indices
 - 25: End for
 - 26: End while
-

More information is included in the method (8), yet MOFDO begins by allocating the search individuals at random across the search space. Hypercube grids are constructed, and an archive with a specific size is established. Line (4) will then initiate the main algorithm loop, which is mostly dependent on a certain number of iterations or to the point where a predetermined condition is satisfied. Line (5) states that, based on the number of individuals, individual (artificial scout bee) procedures from Line 6 to Line 27 will be repeated for each search. The activities that are stated are determining the global best search agent; determining fw with the use of Equation (15); calculating a new search agent position utilizing Equation (14); and Line 10–Line 12, applying conditions from Equations (12)–(14) to compute pace. The method always determines if the new result (cost function) outperforms the old result when a new search agent is found (14). If so, as indicated in Line (15), the new position will be acknowledged and the pace will be saved for possible future use. If not, the previously saved pace will be utilized in place of the new one in the hopes of producing a better result—that is, if the search agent doesn't maintain the present position [see Lines (17–22)]. Line (24) will see the application of the polynomial mutation to obtain further variant solutions. Lines 25 and 26 will then verify whether or not the solution fits inside the archive. Hypercube grid indices are constantly updated in Line (27) by changes in the search landscape for more detailed about MOFDO see in[48].

5.3.5. Multi-Objective Learner Performance-Based Algorithm

To convert a learner performance-based behavior algorithm into a successful MOO algorithm (in other words, the learner with the best skills), we need to reinterpret its essential elements. The answers with the lowest value for a specific objective function are chosen to represent the best learner. The evaluation of numerous objectives that must be maximized or decreased is included in MOPs, nevertheless. Consequently, in order to select learners (individuals) from subgroups, the algorithm ought to utilize an extra criterion. In this instance, the best non-dominated solutions are chosen using the crowding distance from reference [42]. As was shown in previous sections, the crowding distance approach is a useful tool for quantifying the diversity of non-dominated solutions surrounding a single non-dominated solution. The distribution of solutions inside a specific location is more uniform when the crowding distance value is lower. Crowding distance can be utilized only in the objective space, or in the objective space and the parameter space. We arrange all non-dominated solutions based on the result of one of the objectives so that we can use it in the goal space. The search is directed toward a very promising location by using the dp operator to split the main population into multiple subpopulations and give priority to the subpopulation with the most remarkable individuals. The optimal method for the selection of solutions in subsequent iterations is to select individuals as best they can from the top-performing subpopulation, and after that select the next best individuals. This method is thought to be an essential part of the MOLPB algorithm as seen in Algorithm 9. For every iteration, the crowding distance method is applied to all of the non-dominated solutions to preserve a diverse range of solutions and a high rate of convergence. After that, the non-dominated solutions chosen based on the crowding distance are used to reconstitute the subpopulations. As such, the newly generated subpopulations are made up of individuals whose crowding distance value is smaller. The crowding distance technique has been employed by the NSGA-II algorithm to efficiently maintain a high degree of variety [47].

Algorithm 9 MOLPB

- 1: **Start**
 - 2: Step 1 Initialization of operators (size of population, Mutation, Crossover, dp).
 - 3: Step 2 Random generation of the initial population.
 - 4: Step 3 Random selection of multiple individuals (i.e., learners) with the use of the dp operator.
 - 5: Step 4 Chosen individuals are run the previous step through the fitness function.
-

Algorithm 9 *Cont.*

- 6: Step 5 non-dominated method and crowding distance are used for choosing half the population from Step 3, and that group is referred to as the good Population. The second half is renamed as bad Population.
 - a. The crossover and mutation are carried out between half the elements in good Population, and for the second half, partners are brought from the main population. It should be noted that before selecting partners from the main population, the following is performed:
 - b. The optimal element is found (which is the non-dominated element compared with the rest of the elements) in a bad Population.
 - c. elements are removed from the population dominated by optimal elements that had been found in (i).
 - d. In the main population, individuals with a fitness value that is lower than or equal to optimal fitness in the bad Population go to the bad Population.
 - e. The rest of the individuals in the main population have been divided into 2 subpopulations, which are:
 - f. Perfect Population: containing individuals with fitness values that are higher than the optimal fitness in good Population.
 - g. good Population: containing the individuals with the fitness values that are lower smaller than or equal to best fitness in good Population.
 - h. After data filtering, crossover and mutation are performed between the remaining individuals in good Population and elements in perfect Population.
 - i. In the case where individuals from perfect Population have not been enough, the individuals are brought from bad Population.
 - j. non-dominated solutions are stored in external archive.
 - k. crowding distance is found for non-dominated solutions in external archive.
 - l. In all cases where the external archive is full, crowding distance method is utilized for the removal of dominated solutions (the ones with low values of crowding distance) in the archive and non-dominated solutions are stored.
 - m. In the case of meeting the stop condition, this algorithm will stop, or else, go to step 3
- 7: **End**

5.4. A Comprehensive Analysis of the Benefits, Drawbacks, and Practical Uses of Multi-Objective Algorithms

Single- and multi-objective algorithms are essential instruments in the quest for optimized solutions across several areas. Single-objective algorithms focus exclusively on a given optimization target, while multi-objective algorithms aim to optimize many competing objectives in a simultaneous manner, which results in a range of Pareto-optimal solutions.

Knowing the delicate advantages, limitations, and practical uses of these algorithms is paramount to handle complex problems with skill and create efficient solutions. The Table 3 below gives a detailed analysis of the benefits, constraints, and practical uses of MEO, MODE, MOPSO, MOLPB, and MOFDO algorithms. By making clear their usefulness and applicability in various optimization situations, it offers an essential understanding of how well they work for real problem-solving efforts.

Table 3. The advantages, disadvantages, and applications for the multi-objective algorithms.

Multi-Objective Algorithms	Advantage	Disadvantage	Methods	Applications
MOEO	- Useful in the solution of real-world problems	- Memory-less	- Multiple Pareto-optimal solutions - Crowding distance	- Spacecraft trajectory design - Still to be applied to a wide range of science and engineering areas

Table 3. Cont.

Multi-Objective Algorithms	Advantage	Disadvantage	Methods	Applications
MODE	<ul style="list-style-type: none"> - Fast convergence - More effective computational result compared to other algorithms - More accuracy to represent the real Pareto front 	<ul style="list-style-type: none"> - The issue is determining the value of fitness sharing radius 	<ul style="list-style-type: none"> - Parato approach - Crowding distance 	<ul style="list-style-type: none"> - Four-bar truss design problem, - Two constrained problems (disk brake design problem [81,82], welded beam design problem—feature selection [83,84])
MOPSO	<ul style="list-style-type: none"> - Very useful for real-world applications - Uses maximum fitness function to determine Pareto front - Provides good diversity - No need for niching or clustering techniques - Leader is stored in an external archive selected randomly - Different leaders choosing each particle for each decision variable to conform to a single global best 	<ul style="list-style-type: none"> - Easy to fall into local optimum in the high-dimensional space - Low rate of convergence 	<ul style="list-style-type: none"> - Parato (non-dominated) approach - External Archive 	<ul style="list-style-type: none"> - Multi-objective 0/1-knapsack problem. - Pseudo-Boolean discrete problems - Multi-objective combinatorial optimization problems [12,23,53,85]
MOFDO	<ul style="list-style-type: none"> - Uses fitness weight and weight factor parameter to increasing the convergence and coverage in the algorithm - Provides good distribution solutions, enabling the decision-makers to have more variant options to take under consideration. - Time and space will increase linearly 	<ul style="list-style-type: none"> - Complex computation - The performance of the problem depending on the natural order of the problem like other algorithms 	<ul style="list-style-type: none"> - Parato (non-dominated) approach - External Archive - Polynomial mutation - Hypercube grids 	<ul style="list-style-type: none"> - Welded beam design problem [48]
MOLPB	<ul style="list-style-type: none"> - Provides better competition - Good diversity - Capability of optimizing the different real-world engineering problems - Proper method for providing Pareto-optimal solution for a variety of multi-objective problems 	<ul style="list-style-type: none"> - Complex computation - The performance of the problem depending on the natural order of the problem like other algorithms 	<ul style="list-style-type: none"> - Parato (non-dominated) approach and crowding distance - External Archive 	<ul style="list-style-type: none"> - The 4-bar truss design issue - The coil compression spring design problem - The pressure vessel design problem [47] - The car side-impact design problem - The speed reducer design problem

5.5. Multi-Objective Applications

MOO, which stands for multi-objective optimization, is a strong method used in various areas to handle complex decision-making problems with clashing objectives. The use of MOO techniques has been rising in popularity because they can effectively find the best solutions that balance many different criteria at once. In the table, it is clear that MOO applications are spread across many areas like engineering design, systems for renewable energy, healthcare, and finance. Every field of application has its own problems and objectives which need specific optimization techniques and algorithms to be applied. Complex systems and processes can be optimized by researchers or practitioners through the use of MOO approaches. These methods consider multiple contradictory standards in

order to efficiently achieve the desired outputs. The references provided in this table are significant and influential works, as well as contemporary contributions to the research, within each particular field of application.

These studies utilize diverse MOO methods, including NSGA-II, MOEA, and SPEA-2, which have been customized to address the individual demands and intricacies of each issue area. Table 4 provides readers with a comprehensive view of the various applications of MOO and the distinct approaches employed to tackle intricate decision-making problems in diverse domains. Furthermore, it emphasizes the significance of utilizing advanced optimization approaches to better decision-making processes, enhance system performance, and stimulate innovation in many fields.

Table 4. Summary of MOO applications across several areas.

Application Domain	Description	References	Method/Algorithm Used
Engineering design	Optimal design of mechanical, electrical, and structural systems considering multiple objectives such as cost, performance, and reliability.	[27,28,39,48,54,58,86]	NSGA-II, MOEA, multi-objective differential evolution, MOFDO, MOLPB, MOPSO, MOEO multi-objective ant lion optimizer (MOALO), MODA
Renewable energy systems	Design and optimization of renewable energy systems (solar, wind, hydro) considering cost, efficiency, and environmental impact simultaneously.	[5,61–63,65,66,70,72,87–94]	MOPSO, NSGA-II, SPEA2, I-MODE
Supply chain management	Optimization of supply chain networks to minimize cost, lead time, and inventory while maximizing customer satisfaction and resilience to disruptions.	[95–112]	NSGA-II, MOGA, Multi-Objective Genetic Algorithm (MOGA), MOPSO, SPEA2
Healthcare	Patient treatment planning, scheduling, and resource allocation in healthcare systems consider objectives like cost, patient outcomes, and resource utilization.	[45,54,69,71,104,113–140]	NSGA-II, NSGA-III, Multi-Objective Simulated Annealing (MOSA), ϵ -constraint approach, MOPSO, SPEA2, lexicographical method, multi-objective grey wolf optimizer
Environmental management	Conservation planning, land use optimization, and biodiversity conservation consider conflicting objectives like habitat preservation, economic development, and ecosystem services.	[13,72,77,108,111,129,141–156]	MOEA, NSGA-II, Multi-Objective Genetic Algorithm (MOGA) SPEA2, improved multi-objective antlion optimization algorithm, IMOALO
Finance and investment	Portfolio optimization, risk management, and asset allocation considering objectives such as return on investment, risk exposure, and liquidity.	[9,157–176]	NSGA-II, SPEA2, multi-objective particle swarm optimization (MOPSO), dynamic weighted multi-objective planning models

Table 4. Cont.

Application Domain	Description	References	Method/Algorithm Used
Transportation systems	Route optimization, vehicle routing, and traffic management considering objectives like travel time, fuel consumption, emissions, and congestion reduction.	[58,76,99,136,152,177–206]	NSGA-II, MOGA, genetic algorithm (GA) SPEA2, lexicographical method, multi-objective residential DR optimization model, MOQPSO
Water resource management	Allocation of water resources for irrigation, urban supply, and ecosystem conservation considering objectives like water availability, economic value, and environmental sustainability.	[76,194–197,207–214]	NSGA-II, MOEA, MOGWO, self-adaptive multi-objective cuckoo search (SAMOCSA)
Manufacturing processes	Optimization of manufacturing processes and production scheduling considering objectives like cost, throughput, energy consumption, and quality.	[24,64,74,75,92,135,137,145, 153–155,199,215–232]	NSGA-II, differential evolution, multi-objective genetic algorithm (MOGA), lexicographical method SPEA2, multi-objective grey wolf optimizer, improved multi-objective antlion optimization algorithm, IMOALO, mixed integer linear programming
Urban planning	Land use planning, urban infrastructure development, and smart city design consider objectives like livability, accessibility, environmental sustainability, and economic growth.	[24,78,93,138,146,177,181,199, 219,223–225,233–242]	NSGA-II, SPEA2, multi-objective ant colony optimization (MOACO), MOPSO, MOGA, HMOGA, fuzzy decision-making, multi-objective residential DR optimization model, improved multi-objective ant lion optimization algorithm, IMOALO, mixed integer linear programming,

6. Research Directions

Research Gaps

1. Performance Evaluation: Although this paper gives a thorough overview of numerous optimization algorithms, it lacks specific metrics for performance evaluation and comparisons of the algorithms, especially regarding scalability, convergence speed, and solution quality.
2. Real-World Applications: While this paper covers using optimization algorithms in various fields, it falls short in discussing particular real-world case studies or applications where these algorithms have been effectively implemented, as well as the difficulties and practical ramifications of doing so.
3. Hybrid Approaches: A few hybrid optimization approaches are mentioned in passing in this study; however, further research and analysis is needed to determine whether hybridization techniques might enhance the performance of optimization algorithms, especially when tackling challenging real-world issues.
4. Parameter Tuning: This is a crucial step in many optimization algorithms as it has a substantial impact on their performance. However, this review lacks extensive examination of the strategies of parameter tuning in addition to their impact on the behavior of the algorithm as well as the quality of the solution.

5. The focus should be directed towards the enhancement of optimization algorithms' steadiness and dependability in the case of being faced with uncertainty or noise. In addition to that, their efficiency in the handling of the scenarios of the dynamic optimization must be evaluated as detailed as possible.
6. Parallel and Distributed Computing: Parallel and distributed computing approaches' effectiveness in the acceleration of the process of optimization, in addition to the handling of cases with vast amounts of data, is not explored in detail. Such a knowledge gap exists although there are growing requirements for dealing with the complicated problems of optimization at large scales.

7. Future Directions

There are numerous possibilities for future research and development in the optimization area. The advent of algorithms of hybrid optimization offered a possibility for combining benefits from several methods, which potentially leads to stronger and more efficient solutions. The experts may enhance their capability of solving problems that are related to complex real-world situations through the blending of various tactics such as the merging of the evolutionary algorithms with approaches that are based upon the gradients or uniting swarm intelligence, in addition to the programming of the innovative structures of optimization.

In addition to all that, including certain data and problem limits is highly significant in the approaches of optimization. The combination of knowledge from a variety of areas, such as engineering, finance, healthcare, or logistics, with the algorithms to give solutions that are not just perfect but also suited for particular needs and restrictions of the scenario, has potential for improving the quality of the solution, at the same time as encouraging the innovations within particular domains, in addition to promoting collaborations in a variety of the areas. In addition to that, there is a high importance in focusing on making the criteria of evaluation and processes of benchmarking in optimization research more standardized. Through the formation of uniform evaluation standards and datasets that are standard for a variety of the problems of optimization, it is possible to encourage equal comparisons between the algorithms, simplify result duplication, and speed up the advancements. In the case of setting up a standard structure for the evaluation of the efficiency of an algorithm, the researchers can then provide an accurate understanding of the strong points and weak areas of the different strategies, which will be helpful in pushing for better means of algorithm optimization (Liu et al., 2022).

To sum up, the consideration of scalability problems in MOO is very important for solving more difficult problems with big decision spaces and high-dimensional goals. As optimization issues get more complex and computing resources become limited, it is crucial to make scalable algorithms that can efficiently explore large search areas. Studies should focus on advancing methods for parallel and distributed optimization, metaheuristic algorithms with better convergence properties, and adaptive strategies to handle high-dimensional or changing optimization problems.

8. Conclusions

In conclusion, this review paper provides a thorough study of the latest advancements in single- and MOO algorithms. It covers traditional methods as well as contemporary metaheuristic techniques. The examination of various algorithms along with their basic ideas, benefits, and downsides is presented to make clear the theoretical structure and practical application of these approaches. The importance of utilizing some specialized algorithms, such as FDO, DE, PSO, and LPB, is highlighted for the MOO alongside their extensions. These famous widespread EO algorithms are analyzed in detail in this study, which offers a valuable understanding of their success in the case of dealing with a variety of difficulties that are related to optimization within the distinct problem domains. This paper discusses the comparison studies, recent progressions, and upcoming patterns as well, which is helpful in the comprehension of changes in the optimization scope research in

a more detailed sense. The present review provides a detailed MOO application summary in numerous areas, such as engineering design, healthcare, renewable energy systems, and finance. Since each one of the application areas has its own issues and goals, demanding utilizing certain algorithms and techniques of optimization, MOO methods are helpful for practitioners and researchers in the optimization of intricate processes and systems.

In addition to that, this study highlighted that the algorithmic diversity, standardized evaluation metrics, and performance evaluation methodologies, in addition to the strict protocols of benchmarking, are crucial for the progression of this field. It emphasizes the way that the ongoing teamwork between the industrial and academic sectors is vital for handling actual issues. It has been proposed that future studies research hybridization strategies, incorporate certain field knowledge, and tackle issues of scalability in the MOO. In conclusion, the present study represents a valuable aid for individuals who implement MOO algorithms or study them. The detailed examination, recognition of the key research areas requiring attention, and outline of future objectives have the aim of inspiring additional progress and innovation in the field of optimization.

Author Contributions: Conceptualization, N.A.R.; methodology, N.A.R.; validation, N.A.R.; formal analysis N.A.R., Y.H.A. and T.A.R.; investigation, Y.H.A. and T.A.R.; resources, N.A.R.; data curation, N.A.R.; writing—original draft preparation, N.A.R.; writing—review and editing N.A.R., Y.H.A. and T.A.R.; visualization, N.A.R.; supervision, Y.H.A. and T.A.R.; funding acquisition, N.A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

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

References

1. Deb, K.; Anand, A.; Joshi, D. A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization. *Evol. Comput.* **2002**, *10*, 371–395. [[CrossRef](#)] [[PubMed](#)]
2. Cai, X.; Mei, Z.; Fan, Z.; Zhang, Q. A constrained decomposition approach with grids for evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **2017**, *22*, 564–577. [[CrossRef](#)]
3. Mirjalili, S.; Saremi, S.; Mirjalili, S.M.; Coelho, L.d.S. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Syst. Appl.* **2016**, *47*, 106–119. [[CrossRef](#)]
4. Kumawat, I.R.; Nanda, S.J.; Maddila, R.K. Multi-objective whale optimization. In Proceedings of the TENCON 2017—2017 IEEE Region 10 Conference, Penang, Malaysia, 5–8 November 2017; pp. 2747–2752.
5. Wang, J.; Liu, J.; Wang, H.; Mei, C. Approaches to multi-objective optimization and assessment of green infrastructure and their multi-functional effectiveness: A review. *Water* **2020**, *12*, 2714. [[CrossRef](#)]
6. Reyes-Sierra, M.; Coello, C.C. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *Int. J. Comput. Intell. Res.* **2006**, *2*, 287–308.
7. Chai, R.; Savvaris, A.; Tsourdos, A.; Chai, S. Solving multi-objective aeroassisted spacecraft trajectory optimization problems using extended NSGA-II. In Proceedings of the AIAA SPACE and Astronautics Forum and Exposition, Orlando, FL, USA, 12–14 September 2017; p. 5193.
8. Deb, K. *Multiobjective Optimization Using Evolutionary Algorithms*; Wiley: New York, NY, USA, 2001.
9. Saborido, R.; Ruiz, A.B.; Bermúdez, J.D.; Vercher, E.; Luque, M. Evolutionary multi-objective optimization algorithms for fuzzy portfolio selection. *Appl. Soft Comput.* **2016**, *39*, 48–63. [[CrossRef](#)]
10. El Aziz, M.A.; Ewees, A.A.; Hassanien, A.E.; Mudhsh, M.; Xiong, S. Multi-objective whale optimization algorithm for multilevel thresholding segmentation. In *Advances in Soft Computing and Machine Learning in Image Processing*; Springer: Cham, Switzerland, 2018; pp. 23–39.
11. Oesterle, J.; Amodeo, L.; Yalaoui, F. A comparative study of multi-objective algorithms for the assembly line balancing and equipment selection problem under consideration of product design alternatives. *J. Intell. Manuf.* **2019**, *30*, 1021–1046. [[CrossRef](#)]
12. Roy, R.; Dehuri, S.; Cho, S.B. A novel particle swarm optimization algorithm for multi-objective combinatorial optimization problem. *Int. J. Appl. Metaheuristic Comput. (IJAMC)* **2011**, *2*, 41–57. [[CrossRef](#)]
13. Chen, Y.; Li, H.; He, B.; Wang, P.; Jin, K. Multi-objective genetic algorithm based innovative wind farm layout optimization method. *Energy Convers. Manag.* **2015**, *105*, 1318–1327. [[CrossRef](#)]
14. Pennada, V.S.T. Solving Multiple Objective Optimization Problem Using Multi-Agent Systems: A case in Logistics Management. 2020. Available online: <https://www.diva-portal.org/smash/get/diva2:1501741/FULLTEXT01.pdf> (accessed on 25 August 2024).

15. Abbass, H.A.; Sarker, R.; Newton, C. PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), Seoul, Republic of Korea, 27–30 May 2001; pp. 971–978.
16. Xue, F.; Sanderson, A.C.; Graves, R.J. Pareto-based multi-objective differential evolution. In Proceedings of the 2003 Congress on Evolutionary Computation, 2003. CEC'03, Canberra, ACT, Australia, 8–12 December 2003; pp. 862–869.
17. Cui, Y.; Geng, Z.; Zhu, Q.; Han, Y. Multi-objective optimization methods and application in energy saving. *Energy* **2017**, *125*, 681–704. [[CrossRef](#)]
18. Xue, F.; Sanderson, A.C.; Graves, R.J. Multi-objective differential evolution-algorithm, convergence analysis, and applications. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005; pp. 743–750.
19. Liang, J.; Qiao, K.; Yue, C.; Yu, K.; Qu, B.; Xu, R.; Li, Z.; Hu, Y. A clustering-based differential evolution algorithm for solving multimodal multi-objective optimization problems. *Swarm Evol. Comput.* **2021**, *60*, 100788. [[CrossRef](#)]
20. Abdullah, J.M.; Ahmed, T. Fitness dependent optimizer: Inspired by the bee swarming reproductive process. *IEEE Access* **2019**, *7*, 43473–43486. [[CrossRef](#)]
21. Rahman, C.M.; Rashid, T.A. A new evolutionary algorithm: Learner performance based behavior algorithm. *Egypt. Inform. J.* **2021**, *22*, 213–223. [[CrossRef](#)]
22. Costa-Carrapiço, I.; Raslan, R.; González, J.N. A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency. *Energy Build.* **2020**, *210*, 109690. [[CrossRef](#)]
23. Mahrach, M.; Miranda, G.; León, C.; Segredo, E. Comparison between single and multi-objective evolutionary algorithms to solve the knapsack problem and the travelling salesman problem. *Mathematics* **2020**, *8*, 2018. [[CrossRef](#)]
24. Zupančič, J.; Filipič, B.; Gams, M. Genetic-programming-based multi-objective optimization of strategies for home energy-management systems. *Energy* **2020**, *203*, 117769. [[CrossRef](#)]
25. Coello, C. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Springer: New York, NY, USA, 2007.
26. Li, L.; Wang, W.; Xu, X. Multi-objective particle swarm optimization based on global margin ranking. *Inf. Sci.* **2017**, *375*, 30–47. [[CrossRef](#)]
27. Mirjalili, S.; Jangir, P.; Saremi, S. Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems. *Appl. Intell.* **2017**, *46*, 79–95. [[CrossRef](#)]
28. Mirjalili, S.Z.; Mirjalili, S.; Saremi, S.; Faris, H.; Aljarah, I. Grasshopper optimization algorithm for multi-objective optimization problems. *Appl. Intell.* **2018**, *48*, 805–820. [[CrossRef](#)]
29. Tawhid, M.A.; Savsani, V. Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems. *Neural Comput. Appl.* **2019**, *31*, 915–929. [[CrossRef](#)]
30. Lai, X.; Li, C.; Zhang, N.; Zhou, J. A multi-objective artificial sheep algorithm. *Neural Comput. Appl.* **2019**, *31*, 4049–4083. [[CrossRef](#)]
31. Liang, J.; Xu, W.; Yue, C.; Yu, K.; Song, H.; Crisalle, O.D.; Qu, B. Multimodal multiobjective optimization with differential evolution. *Swarm Evol. Comput.* **2019**, *44*, 1028–1059. [[CrossRef](#)]
32. Tian, Y.; Cheng, R.; Zhang, X.; Cheng, F.; Jin, Y. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Trans. Evol. Comput.* **2017**, *22*, 609–622. [[CrossRef](#)]
33. Santiago, A.; Dorronsoro, B.; Nebro, A.J.; Durillo, J.J.; Castillo, O.; Fraire, H.J. A novel multi-objective evolutionary algorithm with fuzzy logic based adaptive selection of operators: FAME. *Inf. Sci.* **2019**, *471*, 233–251. [[CrossRef](#)]
34. Fraga, E.S. An example of multi-objective optimization for dynamic processes. *Chem. Eng. Trans.* **2019**, *74*, 601–606.
35. Afshari, H.; Hare, W.; Tesfamariam, S. Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures. *Appl. Soft Comput.* **2019**, *83*, 105631. [[CrossRef](#)]
36. Khalilpourazari, S.; Naderi, B.; Khalilpourazary, S. Multi-objective stochastic fractal search: A powerful algorithm for solving complex multi-objective optimization problems. *Soft Comput.* **2020**, *24*, 3037–3066. [[CrossRef](#)]
37. Grotti, E.; Mizushima, D.M.; Backes, A.D.; de Freitas Awruch, M.D.; Gomes, H.M. A novel multi-objective quantum particle swarm algorithm for suspension optimization. *Comput. Appl. Math.* **2020**, *39*, 105. [[CrossRef](#)]
38. Zeng, N.; Song, D.; Li, H.; You, Y.; Liu, Y.; Alsaadi, F.E. A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution. *Neurocomputing* **2021**, *432*, 170–182. [[CrossRef](#)]
39. Abdel-Basset, M.; Mohamed, R.; Mirjalili, S. A novel whale optimization algorithm integrated with Nelder–Mead simplex for multi-objective optimization problems. *Knowl.-Based Syst.* **2021**, *212*, 106619. [[CrossRef](#)]
40. Stewart, R.H.; Palmer, T.S.; DuPont, B. A survey of multi-objective optimization methods and their applications for nuclear scientists and engineers. *Prog. Nucl. Energy* **2021**, *138*, 103830. [[CrossRef](#)]
41. Balabanov, T. Solving Multi-Objective Problems by Means of Single Objective Solver. *Probl. Eng. Cybern. Robot.* **2021**, *76*, 63–70. [[CrossRef](#)]
42. Verma, S.; Pant, M.; Snasel, V. A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. *IEEE Access* **2021**, *9*, 57757–57791. [[CrossRef](#)]
43. Khodadadi, N.; Azizi, M.; Talatahari, S.; Sareh, P. Multi-objective crystal structure algorithm (MOCryStAl): Introduction and performance evaluation. *IEEE Access* **2021**, *9*, 117795–117812. [[CrossRef](#)]
44. Premkumar, M.; Jangir, P.; Sowmya, R.; Alhelou, H.H.; Mirjalili, S.; Kumar, B.S. Multi-objective equilibrium optimizer: Framework and development for solving multi-objective optimization problems. *J. Comput. Des. Eng.* **2022**, *9*, 24–50. [[CrossRef](#)]

45. Mei, Y.; Wu, K. Application of multi-objective optimization in the study of anti-breast cancer candidate drugs. *Sci. Rep.* **2022**, *12*, 19347. [[CrossRef](#)]
46. Petrović, M.; Jokić, A.; Miljković, Z.; Kulesza, Z. Multi-objective scheduling of a single mobile robot based on the grey wolf optimization algorithm. *Appl. Soft Comput.* **2022**, *131*, 109784. [[CrossRef](#)]
47. Rahman, C.M.; Rashid, T.A.; Ahmed, A.M.; Mirjalili, S. Multi-objective learner performance-based behavior algorithm with five multi-objective real-world engineering problems. *Neural Comput. Appl.* **2022**, *34*, 6307–6329. [[CrossRef](#)]
48. Abdullah, J.M.; Rashid, T.A.; Maarroof, B.B.; Mirjalili, S. Multi-objective fitness-dependent optimizer algorithm. *Neural Comput. Appl.* **2023**, *35*, 11969–11987. [[CrossRef](#)]
49. Ahmed, A.M.; Rashid, T.A.; Soran, A.M.S.; Noori, K.A.; Hassan, B.A.; Rahman, C.M.; Ahmed, O.H.; Umar, S.U.; Yaseen, Z.M. GMOCSSO: Multi-objective Cat Swarm Optimization Algorithm based on a Grid System. *Preprint* **2023**. [[CrossRef](#)]
50. Goldberg, D.E. Genetic Algorithms in Search. In *Optimization, Machine Learning*; Addison-Wesley Professional: Reading, PA, USA, 1989.
51. Talbi, E. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 2, pp. 268–308.
52. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2010**, *15*, 4–31. [[CrossRef](#)]
53. Wang, B.; Sun, Y.; Xue, B.; Zhang, M. Evolving deep neural networks by multi-objective particle swarm optimization for image classification. In Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 13–17 July 2019; pp. 490–498.
54. Habib, M.; Aljarah, I.; Faris, H.; Mirjalili, S. Multi-objective particle swarm optimization: Theory, literature review, and application in feature selection for medical diagnosis. In *Evolutionary Machine Learning Techniques: Algorithms and Applications*; Springer: Singapore, 2020; pp. 175–201.
55. Engelbrecht, A.P. *Fundamentals of Computational Swarm Intelligence*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2006.
56. Cao, L.; Nietfeld, J.L. College Students' Metacognitive Awareness of Difficulties in Learning the Class Content Does Not Automatically Lead to Adjustment of Study Strategies. *Aust. J. Educ. Dev. Psychol.* **2007**, *7*, 31–46.
57. Qayyum, A. Student help-seeking attitudes and behaviors in a digital era. *Int. J. Educ. Technol. High. Educ.* **2018**, *15*, 17. [[CrossRef](#)]
58. Afrasyabi, P.; Mesgari, M.S.; Khodadai, N.; Kaveh, M. CBMODPSO: Crossover based multi-objective discrete particle swarm optimization for solving multi-modal routing problem. *Preprint* **2022**. [[CrossRef](#)]
59. Mishra, S.K.; Panda, G.; Meher, S. Multi-objective particle swarm optimization approach to portfolio optimization. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 1612–1615.
60. Marler, R.T.; Arora, J.S. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* **2004**, *26*, 369–395. [[CrossRef](#)]
61. Ghiasi, M. Detailed study, multi-objective optimization, and design of an AC-DC smart microgrid with hybrid renewable energy resources. *Energy* **2019**, *169*, 496–507. [[CrossRef](#)]
62. Ming, M.; Wang, R.; Zha, Y.; Zhang, T. Multi-objective optimization of hybrid renewable energy system using an enhanced multi-objective evolutionary algorithm. *Energies* **2017**, *10*, 674. [[CrossRef](#)]
63. Clarke, D.P.; Al-Abdeli, Y.M.; Kothapalli, G. Multi-objective optimisation of renewable hybrid energy systems with desalination. *Energy* **2015**, *88*, 457–468. [[CrossRef](#)]
64. Huang, X.; Guan, Z.; Yang, L. An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem. *Adv. Mech. Eng.* **2018**, *10*, 1687814018801442. [[CrossRef](#)]
65. Samy, M.; Elkhoully, H.I.; Barakat, S. Multi-objective optimization of hybrid renewable energy system based on biomass and fuel cells. *Int. J. Energy Res.* **2021**, *45*, 8214–8230. [[CrossRef](#)]
66. Thirunavukkarasu, M.; Sawle, Y.; Lala, H. A comprehensive review on optimization of hybrid renewable energy systems using various optimization techniques. *Renew. Sustain. Energy Rev.* **2023**, *176*, 113192. [[CrossRef](#)]
67. Pereira, J.L.J.; Oliver, G.A.; Francisco, M.B.; Cunha, S.S., Jr.; Gomes, G.F. A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems. *Arch. Comput. Methods Eng.* **2022**, *29*, 2285–2308. [[CrossRef](#)]
68. Pandya, S.B.; Ravichandran, S.; Manoharan, P.; Jangir, P.; Alhelou, H.H. Multi-objective optimization framework for optimal power flow problem of hybrid power systems considering security constraints. *IEEE Access* **2022**, *10*, 103509–103528. [[CrossRef](#)]
69. Fathollahi-Fard, A.M.; Ahmadi, A.; Karimi, B. Multi-objective optimization of home healthcare with working-time balancing and care continuity. *Sustainability* **2021**, *13*, 12431. [[CrossRef](#)]
70. Dowlatshahi, M.; Hashemi, A. Multi-objective Optimization for Feature Selection: A Review. In *Applied Multi-objective Optimization*; Springer: Singapore, 2024; pp. 155–170.
71. Feng, Z.; Huang, J.; Jin, S.; Wang, G.; Chen, Y. Artificial intelligence-based multi-objective optimisation for proton exchange membrane fuel cell: A literature review. *J. Power Sources* **2022**, *520*, 230808. [[CrossRef](#)]
72. Al-Shahri, O.A.; Ismail, F.B.; Hannan, M.; Lipu, M.H.; Al-Shetwi, A.Q.; Begum, R.; Al-Muhsen, N.F.; Soujeri, E. Solar photovoltaic energy optimization methods, challenges and issues: A comprehensive review. *J. Clean. Prod.* **2021**, *284*, 125465. [[CrossRef](#)]
73. Zhang, J.; Wei, L.; Guo, Z.; Sun, H.; Hu, Z. A survey of meta-heuristic algorithms in optimization of space scale expansion. *Swarm Evol. Comput.* **2024**, *84*, 101462. [[CrossRef](#)]

74. Xu, W.; Wang, X.; Guo, Q.; Song, X.; Zhao, R.; Zhao, G.; Yang, Y.; Xu, T.; He, D. Evolutionary process for engineering optimization in manufacturing applications: Fine brushworks of single-objective to multi-objective/many-objective optimization. *Processes* **2023**, *11*, 693. [[CrossRef](#)]
75. Harkare, V.; Mangrulkar, R.; Thorat, O.; Jain, S.R. Evolutionary Approaches for Multi-objective Optimization and Pareto-Optimal Solution Selection in Data Analytics. In *Applied Multi-objective Optimization*; Springer: Singapore, 2024; pp. 67–94.
76. Jain, P.; Khare, R. Multi-objective Rao algorithm in resilience-based optimal design of water distribution networks. *Water Supply* **2022**, *22*, 4346–4360. [[CrossRef](#)]
77. Biswas, P.P.; Suganthan, P.N.; Amaratunga, G.A. Decomposition based multi-objective evolutionary algorithm for windfarm layout optimization. *Renew. Energy* **2018**, *115*, 326–337. [[CrossRef](#)]
78. Song, Z.; Guan, X.; Cheng, M. Multi-objective optimization strategy for home energy management system including PV and battery energy storage. *Energy Rep.* **2022**, *8*, 5396–5411. [[CrossRef](#)]
79. Deb, K.; Goel, T. Controlled elitist non-dominated sorting genetic algorithms for better convergence. In *Evolutionary Multi-Criterion Optimization*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 67–81.
80. Zhang, J.; Cho, H.; Mago, P.J.; Zhang, H.; Yang, F. Multi-objective particle swarm optimization (MOPSO) for a distributed energy system integrated with energy storage. *J. Therm. Sci.* **2019**, *28*, 1221–1235. [[CrossRef](#)]
81. Nadimi-Shahraki, M.H.; Taghian, S.; Mirjalili, S.; Faris, H. MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems. *Appl. Soft Comput.* **2020**, *97*, 106761. [[CrossRef](#)]
82. Carvalho, J.P.G.; Carvalho, É.C.; Vargas, D.E.; Hallak, P.H.; Lima, B.S.; Lemonge, A.C. Multi-objective optimum design of truss structures using differential evolution algorithms. *Comput. Struct.* **2021**, *252*, 106544. [[CrossRef](#)]
83. De Melo, V.V.; Carosio, G.L. Investigating multi-view differential evolution for solving constrained engineering design problems. *Expert Syst. Appl.* **2013**, *40*, 3370–3377. [[CrossRef](#)]
84. Ali, M.; Siarry, P.; Pant, M. An efficient differential evolution based algorithm for solving multi-objective optimization problems. *Eur. J. Oper. Res.* **2012**, *217*, 404–416. [[CrossRef](#)]
85. Abdollahzadeh, B.; Gharehchopogh, F.S. A multi-objective optimization algorithm for feature selection problems. *Eng. Comput.* **2022**, *38*, 1845–1863. [[CrossRef](#)]
86. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [[CrossRef](#)]
87. León Gómez, J.C.; De León Aldaco, S.E.; Aguayo Alquicira, J. A review of hybrid renewable energy systems: Architectures, battery systems, and optimization techniques. *Eng* **2023**, *4*, 1446–1467. [[CrossRef](#)]
88. Hernández-Pérez, L.G.; Alsuhaibani, A.S.; Ponce-Ortega, J.M.; El-Halwagi, M.M. Multi-objective optimization of ammonia and methanol production processes considering uncertain feedstock compositions of shale/natural gas. *Chem. Eng. Res. Des.* **2022**, *187*, 27–40. [[CrossRef](#)]
89. Fonseca, J.D.; Commenge, J.-M.; Camargo, M.; Falk, L.; Gil, I.D. Sustainability analysis for the design of distributed energy systems: A multi-objective optimization approach. *Appl. Energy* **2021**, *290*, 116746. [[CrossRef](#)]
90. Wang, X.-h.; Zhang, Y.; Sun, X.-y.; Wang, Y.-l.; Du, C.-h. Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size. *Appl. Soft Comput.* **2020**, *88*, 106041. [[CrossRef](#)]
91. Eriksson, E.; Gray, E.M. Optimization of renewable hybrid energy systems—A multi-objective approach. *Renewable energy* **2019**, *133*, 971–999. [[CrossRef](#)]
92. Ullah, K.; Hafeez, G.; Khan, I.; Jan, S.; Javaid, N. A multi-objective energy optimization in smart grid with high penetration of renewable energy sources. *Appl. Energy* **2021**, *299*, 117104. [[CrossRef](#)]
93. Alzahrani, A.; Rahman, M.U.; Hafeez, G.; Rukh, G.; Ali, S.; Murawwat, S.; Iftikhar, F.; Haider, S.I.; Khan, M.I.; Abed, A.M. A strategy for multi-objective energy optimization in smart grid considering renewable energy and batteries energy storage system. *IEEE Access* **2023**, *11*, 33872–33886. [[CrossRef](#)]
94. Heydari, A.; Nezhad, M.M.; Keynia, F.; Fekih, A.; Shahsavari-Pour, N.; Garcia, D.A.; Piras, G. A combined multi-objective intelligent optimization approach considering techno-economic and reliability factors for hybrid-renewable microgrid systems. *J. Clean. Prod.* **2023**, *383*, 135249. [[CrossRef](#)]
95. Jayarathna, C.P.; Agdas, D.; Dawes, L.; Yigitcanlar, T. Multi-objective optimization for sustainable supply chain and logistics: A review. *Sustainability* **2021**, *13*, 13617. [[CrossRef](#)]
96. Hasani, A.; Mokhtari, H.; Fattahi, M. A multi-objective optimization approach for green and resilient supply chain network design: A real-life case study. *J. Clean. Prod.* **2021**, *278*, 123199. [[CrossRef](#)]
97. Zamanian, M.R.; Sadeh, E.; Amini Sabegh, Z.; Ehtesham Rasi, R. A multi-objective optimization model for the resilience and sustainable supply chain: A case study. *Int. J. Supply Oper. Manag.* **2020**, *7*, 51–75.
98. Mohammed, A.; Harris, I.; Soroka, A.; Nujoom, R. A hybrid MCDM-fuzzy multi-objective programming approach for a G-resilient supply chain network design. *Comput. Ind. Eng.* **2019**, *127*, 297–312. [[CrossRef](#)]
99. Ehtesham Rasi, R.; Sohanian, M. A multi-objective optimization model for sustainable supply chain network with using genetic algorithm. *J. Model. Manag.* **2021**, *16*, 714–727. [[CrossRef](#)]
100. Nayeri, S.; Torabi, S.A.; Tavakoli, M.; Sazvar, Z. A multi-objective fuzzy robust stochastic model for designing a sustainable-resilient-responsive supply chain network. *J. Clean. Prod.* **2021**, *311*, 127691. [[CrossRef](#)]

101. Bortolini, M.; Calabrese, F.; Galizia, F.G.; Mora, C. A three-objective optimization model for mid-term sustainable supply chain network design. *Comput. Ind. Eng.* **2022**, *168*, 108131. [[CrossRef](#)]
102. Nayeri, S.; Paydar, M.M.; Asadi-Gangraj, E.; Emami, S. Multi-objective fuzzy robust optimization approach to sustainable closed-loop supply chain network design. *Comput. Ind. Eng.* **2020**, *148*, 106716. [[CrossRef](#)]
103. Sharifi, E.; Amin, S.H.; Fang, L. Designing a sustainable, resilient, and responsive wheat supply chain under mixed uncertainty: A multi-objective approach. *J. Clean. Prod.* **2024**, *434*, 140076. [[CrossRef](#)]
104. Ala, A.; Goli, A.; Mirjalili, S.; Simic, V. A fuzzy multi-objective optimization model for sustainable healthcare supply chain network design. *Appl. Soft Comput.* **2024**, *150*, 111012. [[CrossRef](#)]
105. Khalili-Fard, A.; Parsaee, S.; Bakhshi, A.; Yazdani, M.; Aghsami, A.; Rabbani, M. Multi-objective optimization of closed-loop supply chains to achieve sustainable development goals in uncertain environments. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108052. [[CrossRef](#)]
106. Alizadeh-Meghbrazi, M.; Tosarkani, B.M.; Amin, S.H.; Popovic, M.R.; Ahi, P. Design and optimization of a sustainable and resilient mask supply chain during the COVID-19 pandemic: A multi-objective approach. *Environ. Dev. Sustain.* **2022**, 1–46. [[CrossRef](#)]
107. Belamkar, P.; Biswas, S.; Baidya, A.; Majumder, P.; Bera, U.K. Multi-objective optimization of agro-food supply chain networking problem integrating economic viability and environmental sustainability through type-2 fuzzy-based decision making. *J. Clean. Prod.* **2023**, *421*, 138294. [[CrossRef](#)]
108. Vieira, A.A.; Figueira, J.R.; Fragoso, R. A multi-objective simulation-based decision support tool for wine supply chain design and risk management under sustainability goals. *Expert Syst. Appl.* **2023**, *232*, 120757. [[CrossRef](#)]
109. Al-Ashhab, M. A multi-objective optimization modelling for design and planning a robust closed-loop supply chain network under supplying disruption due to crises. *Ain Shams Eng. J.* **2023**, *14*, 101909. [[CrossRef](#)]
110. Yadegari, M.; Sahebi, H.; Razm, S.; Ashayeri, J. A sustainable multi-objective optimization model for the design of hybrid power supply networks under uncertainty. *Renew. Energy* **2023**, *219*, 119443. [[CrossRef](#)]
111. Wang, C.-H.; Chen, N. A multi-objective optimization approach to balancing economic efficiency and equity in accessibility to multi-use paths. *Transportation* **2021**, *48*, 1967–1986. [[CrossRef](#)]
112. Zhang, S.; Zhuang, Y.; Tao, R.; Liu, L.; Zhang, L.; Du, J. Multi-objective optimization for the deployment of carbon capture utilization and storage supply chain considering economic and environmental performance. *J. Clean. Prod.* **2020**, *270*, 122481. [[CrossRef](#)]
113. Zhang, W.; Cao, K.; Liu, S.; Huang, B. A multi-objective optimization approach for health-care facility location-allocation problems in highly developed cities such as Hong Kong. *Comput. Environ. Urban Syst.* **2016**, *59*, 220–230. [[CrossRef](#)]
114. Pourrezaie-Khaligh, P.; Bozorgi-Amiri, A.; Yousefi-Babadi, A.; Moon, I. Fix-and-optimize approach for a healthcare facility location/network design problem considering equity and accessibility: A case study. *Appl. Math. Model.* **2022**, *102*, 243–267. [[CrossRef](#)]
115. Wang, Z.; Cao, K.; Chiu, Y.L.M.; Feng, Q. Spatial multi-objective optimization of primary healthcare facilities: A case study in Singapore. *Trans. GIS* **2024**, *28*, 564–581. [[CrossRef](#)]
116. Delgado, E.J.; Cabezas, X.; Martin-Barreiro, C.; Leiva, V.; Rojas, F. An equity-based optimization model to solve the location problem for healthcare centers applied to hospital beds and COVID-19 vaccination. *Mathematics* **2022**, *10*, 1825. [[CrossRef](#)]
117. Zhou, X.; Chen, Y.; Li, Y.; Liu, B.; Yu, Z. Spatiotemporal Data-Driven Multiperiod Relocation Optimization of Emergency Medical Services: Maximum Equality Objective. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 269. [[CrossRef](#)]
118. Agrawal, N.; Rabiee, M.; Jabbari, M. Contextual relationships in Juran’s quality principles for business sustainable growth under circular economy perspective: A decision support system approach. *Ann. Oper. Res.* **2023**, 1–31. [[CrossRef](#)]
119. Maleki Rastaghi, M.; Barzinpour, F.; Pishvae, M. A multi-objective hierarchical location-allocation model for the healthcare network design considering a referral system. *Int. J. Eng.* **2018**, *31*, 365–373.
120. Khodaparasti, S.; Maleki, H.; Jahedi, S.; Bruni, M.E.; Beraldi, P. Enhancing community based health programs in Iran: A multi-objective location-allocation model. *Health Care Manag. Sci.* **2017**, *20*, 485–499. [[CrossRef](#)] [[PubMed](#)]
121. Zhang, H.; Zhang, K.; Chen, Y.; Ma, L. Multi-objective two-level medical facility location problem and tabu search algorithm. *Inf. Sci.* **2022**, *608*, 734–756. [[CrossRef](#)]
122. AbdelAziz, A.M.; Alarabi, L.; Basalamah, S.; Hendawi, A. A multi-objective optimization method for hospital admission problem—A case study on COVID-19 patients. *Algorithms* **2021**, *14*, 38. [[CrossRef](#)]
123. Chiang, A.J.; Jeang, A.; Chiang, P.C.; Chiang, P.S.; Chung, C.-P. Multi-objective optimization for simultaneous operating room and nursing unit scheduling. *Int. J. Eng. Bus. Manag.* **2019**, *11*, 1847979019891022. [[CrossRef](#)]
124. Song, J.; Li, X.; Mango, J. Location optimization of urban emergency medical service stations: A hierarchical multi-objective model with a new encoding method of genetic algorithm solution. In Proceedings of the Web and Wireless Geographical Information Systems: 18th International Symposium, W2GIS 2020, Wuhan, China, 13–14 November 2020; pp. 68–82.
125. Zhong, G.; Lu, Y.; Chen, W.; Zhai, G. Multi-objective optimization approach of shelter location with maximum equity: An empirical study in Xin Jiekou district of Nanjing, China. *Geomat. Nat. Hazards Risk* **2023**, *14*, 2165973. [[CrossRef](#)]
126. Wei, F.; Xu, W.; Hua, C. A Multi-Objective Optimization of Physical Activity Spaces. *Land* **2022**, *11*, 1991. [[CrossRef](#)]
127. Singh, D.; Kumar, V.; Vaishali; Kaur, M. Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks. *Eur. J. Clin. Microbiol. Infect. Dis.* **2020**, *39*, 1379–1389. [[CrossRef](#)]

128. Kaur, M.; Gianey, H.K.; Singh, D.; Sabharwal, M. Multi-objective differential evolution based random forest for e-health applications. *Mod. Phys. Lett. B* **2019**, *33*, 1950022. [[CrossRef](#)]
129. Torkayesh, A.E.; Vandchali, H.R.; Tirkolaee, E.B. Multi-objective optimization for healthcare waste management network design with sustainability perspective. *Sustainability* **2021**, *13*, 8279. [[CrossRef](#)]
130. Datta, S.; Kapoor, R.; Mehta, P. A multi-objective optimization model for outpatient care delivery with service fairness. *Bus. Process Manag. J.* **2023**, *29*, 630–652. [[CrossRef](#)]
131. Tanantong, T.; Pannakkong, W.; Chemkomnerd, N. Resource management framework using simulation modeling and multi-objective optimization: A case study of a front-end department of a public hospital in Thailand. *BMC Med. Inform. Decis. Mak.* **2022**, *22*, 10. [[CrossRef](#)] [[PubMed](#)]
132. Wen, T.; Zhang, Z.; Qiu, M.; Wu, Q.; Li, C. A multi-objective optimization method for emergency medical resources allocation. *J. Med. Imaging Health Inform.* **2017**, *7*, 393–399. [[CrossRef](#)]
133. Wang, L.; Shi, H.; Gan, L. Healthcare facility location-allocation optimization for China's developing cities utilizing a multi-objective decision support approach. *Sustainability* **2018**, *10*, 4580. [[CrossRef](#)]
134. Alkaabneh, F.; Diabat, A. A multi-objective home healthcare delivery model and its solution using a branch-and-price algorithm and a two-stage meta-heuristic algorithm. *Transp. Res. Part C Emerg. Technol.* **2023**, *147*, 103838. [[CrossRef](#)]
135. Yang, M.; Ni, Y.; Yang, L. A multi-objective consistent home healthcare routing and scheduling problem in an uncertain environment. *Comput. Ind. Eng.* **2021**, *160*, 107560. [[CrossRef](#)]
136. Yadav, N.; Tanksale, A. A multi-objective approach for reducing Patient's inconvenience in a generalized home healthcare delivery setup. *Expert Syst. Appl.* **2023**, *219*, 119657. [[CrossRef](#)]
137. Belhor, M.; El-Amraoui, A.; Jemai, A.; Delmotte, F. Multi-objective evolutionary approach based on K-means clustering for home health care routing and scheduling problem. *Expert Syst. Appl.* **2023**, *213*, 119035. [[CrossRef](#)]
138. Eriskin, L.; Karatas, M.; Zheng, Y.-J. A robust multi-objective model for healthcare resource management and location planning during pandemics. *Ann. Oper. Res.* **2024**, *335*, 1471–1518. [[CrossRef](#)]
139. Aslani, B.; Rabiee, M.; Jabbari, M.; Delen, D. RETRACTED ARTICLE: An optimization framework for the sustainable healthcare facility location problem using a hierarchical conflict resolution approach. *Ann. Oper. Res.* **2024**, *337*, 31–32. [[CrossRef](#)]
140. Dehghanimohammadabadi, M.; Kabadayi, N. A two-stage AHP multi-objective simulation optimization approach in healthcare. *Int. J. Anal. Hierarchy Process* **2020**, *12*, 117–135.
141. Ramli, M.A.; Bouchekara, H.R.; Milyani, A.H. Wind farm layout optimization using a multi-objective electric charged particles optimization and a variable reduction approach. *Energy Strategy Rev.* **2023**, *45*, 101016. [[CrossRef](#)]
142. Gong, Y.; Abdel-Aty, M.; Yuan, J.; Cai, Q. Multi-objective reinforcement learning approach for improving safety at intersections with adaptive traffic signal control. *Accid. Anal. Prev.* **2020**, *144*, 105655. [[CrossRef](#)]
143. Akyol, G.; Göncü, S.; Silgu, M.A. Multi-objective Optimization Framework for Trade-Off Among Pedestrian Delays and Vehicular Emissions at Signal-Controlled Intersections. *Arab. J. Sci. Eng.* **2024**, *49*, 14117–14130. [[CrossRef](#)]
144. Mallaki, M.; Najibi, S.; Najafi, M.; Shirazi, N.C. Smart grid resiliency improvement using a multi-objective optimization approach. *Sustain. Energy Grids Netw.* **2022**, *32*, 100886. [[CrossRef](#)]
145. Ali, S.; Ullah, K.; Hafeez, G.; Khan, I.; Albogamy, F.R.; Haider, S.I. Solving day-ahead scheduling problem with multi-objective energy optimization for demand side management in smart grid. *Eng. Sci. Technol. Int. J.* **2022**, *36*, 101135. [[CrossRef](#)]
146. Ramezani, M.; Bahmanyar, D.; Razmjoooy, N. A new optimal energy management strategy based on improved multi-objective antlion optimization algorithm: Applications in smart home. *SN Appl. Sci.* **2020**, *2*, 2075. [[CrossRef](#)]
147. Ionescu, C.; Diaz, R.A.C.; Zhao, S.; Ghita, M.; Ghita, M.; Copot, D. A low computational cost, prioritized, multi-objective optimization procedure for predictive control towards cyber physical systems. *IEEE Access* **2020**, *8*, 128152–128166. [[CrossRef](#)]
148. Shi, M.; He, H.; Li, J.; Han, M.; Jia, C. Multi-objective tradeoff optimization of predictive adaptive cruising control for autonomous electric buses: A cyber-physical-energy system approach. *Appl. Energy* **2021**, *300*, 117385. [[CrossRef](#)]
149. Xie, D.; Qiu, Y.; Huang, J. Multi-objective optimization for green logistics planning and operations management: From economic to environmental perspective. *Comput. Ind. Eng.* **2024**, *189*, 109988. [[CrossRef](#)]
150. Trivedi, V.; Varshney, P.; Ramteke, M. A simplified multi-objective particle swarm optimization algorithm. *Swarm Intell.* **2020**, *14*, 83–116. [[CrossRef](#)]
151. Mahto, P.K.; Das, P.P.; Diyaley, S.; Kundu, B. Parametric optimization of solar air heaters with dimples on absorber plates using metaheuristic approaches. *Appl. Therm. Eng.* **2024**, *242*, 122537. [[CrossRef](#)]
152. Tang, J.; Liu, G.; Pan, Q. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1627–1643. [[CrossRef](#)]
153. Liu, C.; Wu, Z.; Zhang, Y.; Wang, Y.; Guo, F.; Wang, Y. Optimizing emergency supply location selection in urban areas: A multi-objective planning model and algorithm. *J. Urban Dev. Manag.* **2023**, *2*, 34–46. [[CrossRef](#)]
154. Liu, H.; Li, Y.; Duan, Z.; Chen, C. A review on multi-objective optimization framework in wind energy forecasting techniques and applications. *Energy Convers. Manag.* **2020**, *224*, 113324. [[CrossRef](#)]
155. Jabber, S.A.; Hashem, S.H.; Jafer, S.H. Task Scheduling and Resource Allocation in Cloud Computing: A Review and Analysis. In Proceedings of the 2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA), Taiz, Yemen, 10–11 October 2023; pp. 01–08.

156. Moreno, S.R.; Pierezan, J.; dos Santos Coelho, L.; Mariani, V.C. Multi-objective lightning search algorithm applied to wind farm layout optimization. *Energy* **2021**, *216*, 119214. [[CrossRef](#)]
157. García García, F.; González-Bueno, J.; Guijarro, F.; Oliver-Muncharaz, J.; Tamosiuniene, R. Multiobjective approach to portfolio optimization in the light of the credibility theory. *Technol. Econ. Dev. Econ. (Online)* **2020**, *26*, 1165–1186. [[CrossRef](#)]
158. Zheng, Y.; Zheng, J. A novel portfolio optimization model via combining multi-objective optimization and multi-attribute decision making. *Appl. Intell.* **2022**, *52*, 5684–5695. [[CrossRef](#)]
159. Doumpos, M.; Zopounidis, C. Multi-objective optimization models in finance and investments. *J. Glob. Optim.* **2020**, *76*, 243–244. [[CrossRef](#)]
160. Jalota, H.; Mandal, P.K.; Thakur, M.; Mittal, G. A novel approach to incorporate investor's preference in fuzzy multi-objective portfolio selection problem using credibility measure. *Expert Syst. Appl.* **2023**, *212*, 118583. [[CrossRef](#)]
161. Shouheng, T.; Hong, H. DEaf-MOPS/D: An improved differential evolution algorithm for solving complex multi-objective portfolio selection problems based on decomposition. *Econ. Comput. Econ. Cybernet. Stud. Res.* **2019**, *53*, 151–167.
162. Silva, Y.L.T.; Herthel, A.B.; Subramanian, A. A multi-objective evolutionary algorithm for a class of mean-variance portfolio selection problems. *Expert Syst. Appl.* **2019**, *133*, 225–241. [[CrossRef](#)]
163. Meghwani, S.S.; Thakur, M. Multi-objective heuristic algorithms for practical portfolio optimization and rebalancing with transaction cost. *Appl. Soft Comput.* **2018**, *67*, 865–894. [[CrossRef](#)]
164. Radulescu, M.; Radulescu, C.Z. A multi-objective approach to multi-period: Portfolio optimization with transaction costs. In *Financial Decision Aid Using Multiple Criteria: Recent Models and Applications*; Springer: Cham, Switzerland, 2018; pp. 93–112.
165. Saïb, M.N.-U.-D.E.; Gopaul, A.; Cheeneebash, J. A squirrel search algorithm for the multi-objective portfolio optimisation with transaction costs. *Sci. Afr.* **2024**, *24*, e02166.
166. Wang, Z.; Zhang, X.; Zhang, Z.; Sheng, D. Credit portfolio optimization: A multi-objective genetic algorithm approach. *Borsa Istanbul Rev.* **2022**, *22*, 69–76. [[CrossRef](#)]
167. Ruan, S.-X.; Zhang, X.-B.; Luo, Z.-H. Investigation and optimization of polyolefin elastomers polymerization processes using multi-objective genetic algorithm. *Chem. Eng. Res. Des.* **2023**, *193*, 383–393. [[CrossRef](#)]
168. Kumar, A.; Yadav, S.; Gupta, P.; Mehlawat, M.K. A credibilistic multiobjective multiperiod efficient portfolio selection approach using data envelopment analysis. *IEEE Trans. Eng. Manag.* **2021**, *70*, 2334–2348. [[CrossRef](#)]
169. Peykani, P.; Nouri, M.; Eshghi, F.; Khamechian, M.; Farrokhi-Asl, H. A novel mathematical approach for fuzzy multi-period multi-objective portfolio optimization problem under uncertain environment and practical constraints. *J. Fuzzy Ext. Appl.* **2021**, *2*, 191–203.
170. Wu, Q.; Liu, X.; Qin, J.; Zhou, L.; Mardani, A.; Deveci, M. An integrated multi-criteria decision-making and multi-objective optimization model for socially responsible portfolio selection. *Technol. Forecast. Soc. Chang.* **2022**, *184*, 121977. [[CrossRef](#)]
171. Yadav, S.; Kumar, A.; Mehlawat, M.K.; Gupta, P.; Charles, V. A multi-objective sustainable financial portfolio selection approach under an intuitionistic fuzzy framework. *Inf. Sci.* **2023**, *646*, 119379. [[CrossRef](#)]
172. Ruiz, A.B.; Saborido, R.; Bermúdez, J.D.; Luque, M.; Vercher, E. Preference-based evolutionary multi-objective optimization for portfolio selection: A new credibilistic model under investor preferences. *J. Glob. Optim.* **2020**, *76*, 295–315. [[CrossRef](#)]
173. Wu, Y. Portfolio Research Based on SVM-GARCH and Dynamic Weighted Multi-Objective Planning Models—An Example of Gold and Bitcoin. *Financ. Eng. Risk Manag.* **2023**, *6*, 93–106.
174. Zhao, H.; Chen, Z.-G.; Zhan, Z.-H.; Kwong, S.; Zhang, J. Multiple populations co-evolutionary particle swarm optimization for multi-objective cardinality constrained portfolio optimization problem. *Neurocomputing* **2021**, *430*, 58–70. [[CrossRef](#)]
175. Krink, T.; Paterlini, S. Multiobjective optimization using differential evolution for real-world portfolio optimization. *Comput. Manag. Sci.* **2011**, *8*, 157–179. [[CrossRef](#)]
176. Nouri, M.; Pishvae, M.S.; Mohammadi, E. A Novel Multi-Stage Stochastic Portfolio Optimization Model Under Transaction Costs. *Decis. Mak. Theory Pract.* **2023**, *1*, 13–27.
177. Xiao, M.; Chen, L.; Feng, H.; Peng, Z.; Long, Q. Smart City Public Transportation Route Planning Based on Multi-objective Optimization: A Review. *Arch. Comput. Methods Eng.* **2024**, *31*, 3351–3375. [[CrossRef](#)]
178. Alshammari, N.F.; Samy, M.M.; Barakat, S. Comprehensive analysis of multi-objective optimization algorithms for sustainable hybrid electric vehicle charging systems. *Mathematics* **2023**, *11*, 1741. [[CrossRef](#)]
179. Zhang, Z.; Zhu, H.; Zhang, W.; Cai, Z.; Zhu, L.; Li, Z. Multi-Objective Optimization of Traffic Signal Timing at Typical Junctions Based on Genetic Algorithms. *Comput. Syst. Sci. Eng.* **2023**, *47*, 1901–1917. [[CrossRef](#)]
180. Amer, A.; Shaban, K.; Gaouda, A.; Massoud, A. Home energy management system embedded with a multi-objective demand response optimization model to benefit customers and operators. *Energies* **2021**, *14*, 257. [[CrossRef](#)]
181. Gupta, P.; Mehlawat, M.K.; Aggarwal, U.; Charles, V. An integrated AHP-DEA multi-objective optimization model for sustainable transportation in mining industry. *Resour. Policy* **2021**, *74*, 101180. [[CrossRef](#)]
182. Akkad, M.Z.; Bányai, T. Multi-objective approach for optimization of city logistics considering energy efficiency. *Sustainability* **2020**, *12*, 7366. [[CrossRef](#)]
183. Brands, T.; Van Berkum, E.C. Performance of a genetic algorithm for solving the multi-objective, multimodal transportation network design problem. *Int. J. Transp.* **2014**, *2*, 1–20. [[CrossRef](#)]
184. Baykasoğlu, A.; Subulan, K. A multi-objective sustainable load planning model for intermodal transportation networks with a real-life application. *Transp. Res. Part E Logist. Transp. Rev.* **2016**, *95*, 207–247. [[CrossRef](#)]

185. Durmaz, Y.G.; Bilgen, B. Multi-objective optimization of sustainable biomass supply chain network design. *Appl. Energy* **2020**, *272*, 115259. [\[CrossRef\]](#)
186. Wu, Z.; Wu, J.; Chen, Y.; Liu, K.; Feng, L. Network rebalance and operational efficiency of sharing transportation system: Multi-objective optimization and model predictive control approaches. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17119–17129. [\[CrossRef\]](#)
187. Seydanlou, P.; Jolai, F.; Tavakkoli-Moghaddam, R.; Fathollahi-Fard, A.M. A multi-objective optimization framework for a sustainable closed-loop supply chain network in the olive industry: Hybrid meta-heuristic algorithms. *Expert Syst. Appl.* **2022**, *203*, 117566. [\[CrossRef\]](#)
188. Mrabti, N.; Hamani, N.; Boulaksil, Y.; Gargouri, M.A.; Delahoche, L. A multi-objective optimization model for the problems of sustainable collaborative hub location and cost sharing. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *164*, 102821. [\[CrossRef\]](#)
189. Men, J.; Chen, G.; Zhou, L.; Chen, P. A pareto-based multi-objective network design approach for mitigating the risk of hazardous materials transportation. *Process Saf. Environ. Prot.* **2022**, *161*, 860–875. [\[CrossRef\]](#)
190. Farahmand-Tabar, S.; Afrasyabi, P. Multi-modal Routing in Urban Transportation Network Using Multi-objective Quantum Particle Swarm Optimization. In *Applied Multi-Objective Optimization*; Springer: Singapore, 2024; pp. 133–154.
191. Chung, J.-H.; Bae, Y.K.; Kim, J. Optimal sustainable road plans using multi-objective optimization approach. *Transp. Policy* **2016**, *49*, 105–113. [\[CrossRef\]](#)
192. Yang, L.; Zhang, C.; Wu, X. Multi-objective path optimization of highway-railway multimodal transport considering carbon emissions. *Appl. Sci.* **2023**, *13*, 4731. [\[CrossRef\]](#)
193. Alolaiwy, M.; Hawsawi, T.; Zohdy, M.; Kaur, A.; Louis, S. Multi-objective routing optimization in electric and flying vehicles: A genetic algorithm perspective. *Appl. Sci.* **2023**, *13*, 10427. [\[CrossRef\]](#)
194. Cunha, M.; Magini, R.; Marques, J. Multi-Objective Optimization Models for the Design of Water Distribution Networks by Exploring Scenario-Based Approaches. *Water Resour. Res.* **2023**, *59*, e2023WR034867. [\[CrossRef\]](#)
195. Zhang, C.; Liu, H.; Pei, S.; Zhao, M.; Zhou, H. Multi-objective operational optimization toward improved resilience in water distribution systems. *AQUA—Water Infrastruct. Ecosyst. Soc.* **2022**, *71*, 593–607. [\[CrossRef\]](#)
196. Palod, N.; Prasad, V.; Khare, R. A new multi-objective evolutionary algorithm for the optimization of water distribution networks. *Water Supply* **2022**, *22*, 8972–8987. [\[CrossRef\]](#)
197. Ulusoy, A.-J.; Mahmoud, H.A.; Pecci, F.; Keedwell, E.C.; Stoianov, I. Bi-objective design-for-control for improving the pressure management and resilience of water distribution networks. *Water Res.* **2022**, *222*, 118914. [\[CrossRef\]](#)
198. Jafari, S.; Zahiri, A.; Bozorg-Haddad, O.; Tabari, M. Development of multi-objective optimization model for water distribution network using a new reliability index. *Int. J. Environ. Sci. Technol.* **2022**, *19*, 9757–9774. [\[CrossRef\]](#)
199. Al-Sahlawi, A.A.K.; Ayob, S.M.; Tan, C.W.; Ridha, H.M.; Hachim, D.M. Optimal Design of Grid-Connected Hybrid Renewable Energy System Considering Electric Vehicle Station Using Improved Multi-Objective Optimization: Techno-Economic Perspectives. *Sustainability* **2024**, *16*, 2491. [\[CrossRef\]](#)
200. Zhang, X.; Fan, X.; Yu, S.; Shan, A.; Fan, S.; Xiao, Y.; Dang, F. Intersection signal timing optimization: A multi-objective evolutionary algorithm. *Sustainability* **2022**, *14*, 1506. [\[CrossRef\]](#)
201. Zhang, X.; Fan, X.; Yu, S.; Shan, A.; Men, R. Multi-Objective Optimization Method for Signalized Intersections in Intelligent Traffic Network. *Sensors* **2023**, *23*, 6303. [\[CrossRef\]](#)
202. Dinçer, A.E.; Demir, A.; Yılmaz, K. Multi-objective turbine allocation on a wind farm site. *Appl. Energy* **2024**, *355*, 122346. [\[CrossRef\]](#)
203. Ma, W.; Wan, L.; Yu, C.; Zou, L.; Zheng, J. Multi-objective optimization of traffic signals based on vehicle trajectory data at isolated intersections. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102821. [\[CrossRef\]](#)
204. Reyad, P.; Sayed, T. Real-Time multi-objective optimization of safety and mobility at signalized intersections. *Transp. B Transp. Dyn.* **2023**, *11*, 847–868. [\[CrossRef\]](#)
205. Barakat, S.; Osman, A.I.; Tag-Eldin, E.; Telba, A.A.; Mageed, H.M.A.; Samy, M. Achieving green mobility: Multi-objective optimization for sustainable electric vehicle charging. *Energy Strategy Rev.* **2024**, *53*, 101351. [\[CrossRef\]](#)
206. Chupradit, S.; Tashtoush, M.A.; Al-Muttar, M.Y.O.; Mahmudiono, T.; Dwijendra, N.K.A.; Chaudhary, P.; Ali, M.H.; Alkhayyat, A. A multi-objective mathematical model for the population-based transportation network planning. *Ind. Eng. Manag. Syst.* **2022**, *21*, 322–331. [\[CrossRef\]](#)
207. Gupta, R.S.; Hamilton, A.L.; Reed, P.M.; Characklis, G.W. Can modern multi-objective evolutionary algorithms discover high-dimensional financial risk portfolio tradeoffs for snow-dominated water-energy systems? *Adv. Water Resour.* **2020**, *145*, 103718. [\[CrossRef\]](#)
208. Sarbu, I.; Popa-Albu, S.; Tokar, A. Multi-objective optimization of water distribution networks: An overview. *Int. J. Adv. Appl. Sci.* **2020**, *7*, 74–86.
209. Kidanu, R.A.; Cunha, M.; Salomons, E.; Ostfeld, A. Improving multi-objective optimization methods of water distribution networks. *Water* **2023**, *15*, 2561. [\[CrossRef\]](#)
210. Nyahora, P.P.; Babel, M.S.; Ferras, D.; Emen, A. Multi-objective optimization for improving equity and reliability in intermittent water supply systems. *Water Supply* **2020**, *20*, 1592–1603. [\[CrossRef\]](#)
211. Ramani, K.; Rudraswamy, G.; Umamahesh, N.V. Optimal Design of Intermittent Water Distribution Network Considering Network Resilience and Equity in Water Supply. *Water* **2023**, *15*, 3265. [\[CrossRef\]](#)

212. Ferdowsi, A.; Singh, V.P.; Ehteram, M.; Mirjalili, S. Multi-objective optimization approaches for design, planning, and management of water resource systems. In *Essential Tools for Water Resources Analysis, Planning, and Management*; Springer: Singapore, 2021; pp. 275–303.
213. Assad, A.; Moselhi, O.; Zayed, T. A new metric for assessing resilience of water distribution networks. *Water* **2019**, *11*, 1701. [[CrossRef](#)]
214. Choi, Y.H.; Kim, J.H. Development of multi-objective optimal redundant design approach for multiple pipe failure in water distribution system. *Water* **2019**, *11*, 553. [[CrossRef](#)]
215. Xu, W.; Wang, L.; Liu, D.; Tang, H.; Li, Y. Solving distributed low carbon scheduling problem for large complex equipment manufacturing using an improved hybrid artificial bee colony algorithm. *J. Intell. Fuzzy Syst.* **2023**, *45*, 147–175. [[CrossRef](#)]
216. Huang, K.; Li, R.; Gong, W.; Bian, W.; Wang, R. Competitive and cooperative-based strength Pareto evolutionary algorithm for green distributed heterogeneous flow shop scheduling. *Intell. Autom. Soft Comput.* **2023**, *37*, 2077–2101. [[CrossRef](#)]
217. Mohammadi, Y.; Shakouri, H.; Kazemi, A. A multi-objective fuzzy optimization model for electricity generation and consumption management in a micro smart grid. *Sustain. Cities Soc.* **2022**, *86*, 104119. [[CrossRef](#)]
218. Ashrafi, R.; Amirahmadi, M.; Tolou-Askari, M.; Ghods, V. Multi-objective resilience enhancement program in smart grids during extreme weather conditions. *Int. J. Electr. Power Energy Syst.* **2021**, *129*, 106824. [[CrossRef](#)]
219. Alhasnawi, B.N.; Jasim, B.H.; Jasim, A.M.; Bureš, V.; Alhasnawi, A.N.; Homod, R.Z.; Alsemawai, M.R.M.; Abbassi, R.; Sedhom, B.E. A multi-objective improved cockroach swarm algorithm approach for apartment energy management systems. *Information* **2023**, *14*, 521. [[CrossRef](#)]
220. Malik, M.Z.; Shaikh, P.H.; Khatri, S.A.; Shaikh, M.S.; Baloch, M.H.; Shaikh, F. Analysis of multi-objective optimization: A technical proposal for energy and comfort management in buildings. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e12736. [[CrossRef](#)]
221. Gheouany, S.; Ouadi, H.; El Bakali, S. Energy demand management in a residential building using multi-objective optimization algorithms. In *Proceedings of the International Conference on Advanced Intelligent Systems for Sustainable Development*, Rabat, Morocco, 22–27 May 2022; Springer: Cham, Switzerland, 2023; pp. 368–377.
222. Quan, Z.; Wang, Y.; Ji, Z. Multi-objective optimization scheduling for manufacturing process based on virtual workflow models. *Appl. Soft Comput.* **2022**, *122*, 108786. [[CrossRef](#)]
223. Barakat, S.; Ibrahim, H.; Elbaset, A.A. Multi-objective optimization of grid-connected PV-wind hybrid system considering reliability, cost, and environmental aspects. *Sustain. Cities Soc.* **2020**, *60*, 102178. [[CrossRef](#)]
224. Soares, J.; Ghazvini, M.A.F.; Vale, Z.; de Moura Oliveira, P. A multi-objective model for the day-ahead energy resource scheduling of a smart grid with high penetration of sensitive loads. *Appl. Energy* **2016**, *162*, 1074–1088. [[CrossRef](#)]
225. Yin, W.; Mavaluru, D.; Ahmed, M.; Abbas, M.; Darvishan, A. Application of new multi-objective optimization algorithm for EV scheduling in smart grid through the uncertainties. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 2071–2103. [[CrossRef](#)]
226. Sousa, T.; Morais, H.; Vale, Z.; Castro, R. A multi-objective optimization of the active and reactive resource scheduling at a distribution level in a smart grid context. *Energy* **2015**, *85*, 236–250. [[CrossRef](#)]
227. Tostado-Véliz, M.; Gurung, S.; Jurado, F. Efficient solution of many-objective Home Energy Management systems. *Int. J. Electr. Power Energy Syst.* **2022**, *136*, 107666. [[CrossRef](#)]
228. Chatterjee, A.; Paul, S.; Ganguly, B. Multi-objective energy management of a smart home in real time environment. *IEEE Trans. Ind. Appl.* **2022**, *59*, 138–147. [[CrossRef](#)]
229. Li, F.; Zhang, L.; Liao, T.W.; Liu, Y. Multi-objective optimisation of multi-task scheduling in cloud manufacturing. *Int. J. Prod. Res.* **2019**, *57*, 3847–3863. [[CrossRef](#)]
230. Qu, Y.; Ming, X.; Liu, Z.; Zhang, X.; Hou, Z. Smart manufacturing systems: State of the art and future trends. *Int. J. Adv. Manuf. Technol.* **2019**, *103*, 3751–3768. [[CrossRef](#)]
231. Shaikh, P.H.; Nor, N.B.M.; Nallagownden, P.; Elamvazuthi, I.; Ibrahim, T. Intelligent multi-objective control and management for smart energy efficient buildings. *Int. J. Electr. Power Energy Syst.* **2016**, *74*, 403–409. [[CrossRef](#)]
232. Veras, J.M.; Silva, I.R.S.; Pinheiro, P.R.; Rabêlo, R.A.; Veloso, A.F.S.; Borges, F.A.S.; Rodrigues, J.J. A multi-objective demand response optimization model for scheduling loads in a home energy management system. *Sensors* **2018**, *18*, 3207. [[CrossRef](#)]
233. Dorahaki, S.; MollahassaniPour, M.; Rashidinejad, M. Optimizing energy payment, user satisfaction, and self-sufficiency in flexibility-constrained smart home energy management: A multi-objective optimization approach. *e-Prime-Adv. Electr. Eng. Electron. Energy* **2023**, *6*, 100385. [[CrossRef](#)]
234. Shaikh, P.H.; Nor, N.B.M.; Nallagownden, P.; Elamvazuthi, I. Intelligent multi-objective optimization for building energy and comfort management. *J. King Saud Univ.-Eng. Sci.* **2018**, *30*, 195–204. [[CrossRef](#)]
235. Mansouri, S.A.; Ahmarinejad, A.; Nematbakhsh, E.; Javadi, M.S.; Jordehi, A.R.; Catalao, J.P. Energy management in microgrids including smart homes: A multi-objective approach. *Sustain. Cities Soc.* **2021**, *69*, 102852. [[CrossRef](#)]
236. Zhang, Y.; Zeng, P.; Li, S.; Zang, C.; Li, H. A novel multiobjective optimization algorithm for home energy management system in smart grid. *Math. Probl. Eng.* **2015**, *2015*, 807527. [[CrossRef](#)]
237. Kumar, G.; Kumar, L.; Kumar, S. Multi-objective control-based home energy management system with smart energy meter. *Electr. Eng.* **2023**, *105*, 2095–2105. [[CrossRef](#)]
238. Gheouany, S.; Ouadi, H.; El Bakali, S. Hybrid-integer algorithm for a multi-objective optimal home energy management system. *Clean Energy* **2023**, *7*, 375–388. [[CrossRef](#)]

239. Bahmanyar, D.; Razmjoooy, N.; Mirjalili, S. Multi-objective scheduling of IoT-enabled smart homes for energy management based on Arithmetic Optimization Algorithm: A Node-RED and NodeMCU module-based technique. *Knowl.-Based Syst.* **2022**, *247*, 108762. [[CrossRef](#)]
240. Ullah, I.; Hussain, S. Time-constrained nature-inspired optimization algorithms for an efficient energy management system in smart homes and buildings. *Appl. Sci.* **2019**, *9*, 792. [[CrossRef](#)]
241. Mrazek, M.; Honc, D.; Riva Sanseverino, E.; Zizzo, G. Simplified Energy Model and Multi-Objective Energy Consumption Optimization of a Residential House. *Appl. Sci.* **2022**, *12*, 10212. [[CrossRef](#)]
242. Abdelaziz, F.B.; Alaya, H.; Dey, P.K. A multi-objective particle swarm optimization algorithm for business sustainability analysis of small and medium sized enterprises. *Ann. Oper. Res.* **2020**, *293*, 557–586. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.