mathematics

MDPI

Review

Nature-Inspired Metaheuristic Techniques for Combinatorial Optimization Problems: Overview and Recent Advances

Md Ashikur Rahman [1,*], Rajalingam Sokkalingam [1], Mahmod Othman [1], Kallol Biswas [2], Lazim Abdullah [3] and Evizal Abdul Kadir [4]

[1] Fundamental and Applied Sciences Department, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; raja.sokkalingam@utp.edu.my (R.S.); mahmod.othman@utp.edu.my (M.O.)
[2] Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; kallolbiseee@gmail.com
[3] School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, Kuala Terengganu 21030, Malaysia; lazim_m@umt.edu.my
[4] Department of Informatics Engineering, Universitas Islam Riau, Pekanbaru 28284, Indonesia; evizal@eng.uir.ac.id
* Correspondence: rahman.ashik.eng@gmail.com

Abstract: Combinatorial optimization problems are often considered NP-hard problems in the field of decision science and the industrial revolution. As a successful transformation to tackle complex dimensional problems, metaheuristic algorithms have been implemented in a wide area of combinatorial optimization problems. Metaheuristic algorithms have been evolved and modified with respect to the problem nature since it was recommended for the first time. As there is a growing interest in incorporating necessary methods to develop metaheuristics, there is a need to rediscover the recent advancement of metaheuristics in combinatorial optimization. From the authors' point of view, there is still a lack of comprehensive surveys on current research directions. Therefore, a substantial part of this paper is devoted to analyzing and discussing the modern age metaheuristic algorithms that gained popular use in mostly cited combinatorial optimization problems such as vehicle routing problems, traveling salesman problems, and supply chain network design problems. A survey of seven different metaheuristic algorithms (which are proposed after 2000) for combinatorial optimization problems is carried out in this study, apart from conventional metaheuristics like simulated annealing, particle swarm optimization, and tabu search. These metaheuristics have been filtered through some key factors like easy parameter handling, the scope of hybridization as well as performance efficiency. In this study, a concise description of the framework of the selected algorithm is included. Finally, a technical analysis of the recent trends of implementation is discussed, along with the impacts of algorithm modification on performance, constraint handling strategy, the handling of multi-objective situations using hybridization, and future research opportunities.

Keywords: combinatorial optimization; metaheuristic optimization; vehicle routing problems; travelling salesman problems; supply chain design optimization; constraint handling

Citation: Rahman, M.A.; Sokkalingam, R.; Othman, M.; Biswas, K.; Abdullah, L.; Abdul Kadir, E. Nature-Inspired Metaheuristic Techniques for Combinatorial Optimization Problems: Overview and Recent Advances. *Mathematics* **2021**, *9*, 2633. https://doi.org/10.3390/math9202633

Academic Editors: Víctor Yepes and José Moreno-Jiménez

Received: 13 August 2021
Accepted: 21 September 2021
Published: 19 October 2021

1. Introduction

In the last two decades, combinatorial optimization (CO) has become a hub of vast research prospects from applied mathematics to operational research. As definition [1] stated that combinatorial optimization is a field that looks to find optimal area among a finite set of items. The core purpose of this research field is to adjust the real-world complex problem into a comprehensive output. In the real world, most problems have large data sizes and multiple objectives even though contradictory and complex constraints. In application prospect most of the combinatorial optimization problems can be categorized into three branches; for example, vehicle routing problems (VRP), traveling salesman problems (TSP), and facility location/supply-chain design problems (SCND) and Figure 1.

represents a schematic diagram of the popular branches of CO problems. As combinatorial optimization is considered as a NP-hard (non-deterministic polynomial time hardness) problem, it takes a significant amount of work to achieve any outcome for a certain problem.

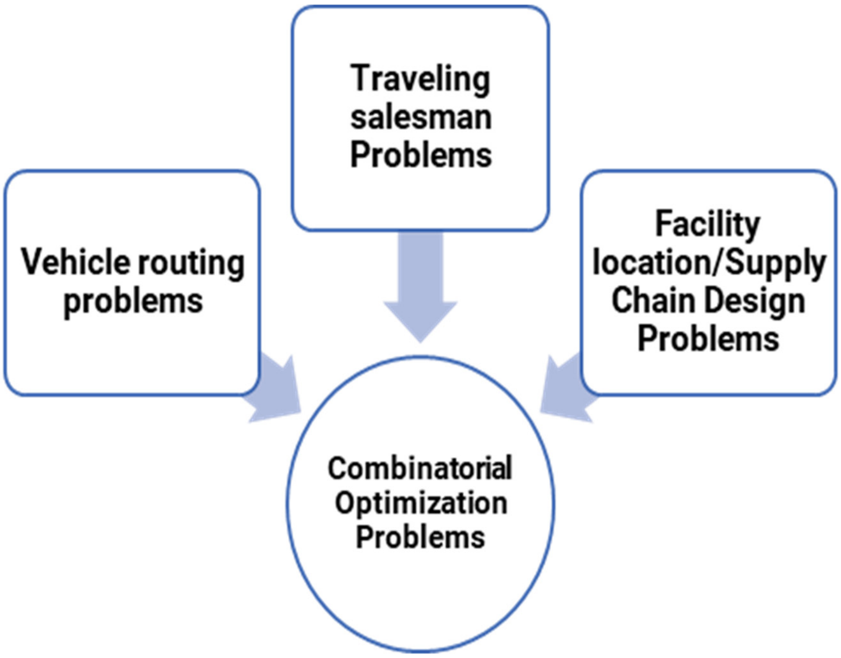
Traveling salesman Problems
Vehicle routing problems
Facility location/Supply Chain Design Problems
Combinatorial Optimization Problems

Figure 1. Selected combinatorial optimization problems.

Considering the complexity, the problem structure for NP-hard problems, researchers have adopted two types of strategy, such as exact methods and metaheuristic methods [2]. Generically, subsidiaries of exact methods like branch and bound and dynamic programming have successfully solved small instance problems. Where the problem has a large number of decision variables, it is observed that exact methods take a large computational time for execution [3]. To tackle this issue, researchers implemented a new method of approximation called metaheuristics. Metaheuristics are the special branch of heuristic methods that can explore a whole problem space with different heuristic approaches by a replicated intelligent iterative process. Normally, metaheuristics stimulate the natural evolution strategy to produce convergence, which needs less parameters to tune for any certain problem. This is why metaheuristic has been implemented to large instance NP-hard problems, which leads to a similar optimization like the exact methods [4].

As the industrial revolution has continued for the last two decades, the use of metaheuristics in combinatorial optimization has been a topic of great interest for researchers and practitioners. From the Web of Science database, the chart in Figure 2 can demonstrate the implementation trend of metaheuristics in combinatorial optimization.

Metaheuristic methods have been developed or modified before 2000, and they can be considered state-of-the-art in techniques such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) [5]. Eventually, after 2000, researchers adopted newly invented metaheuristics in many areas in operational research, so there has been a large prospect to survey the application scope of metaheuristics in combinatorial optimization. In this research, authors segregated the review of the metaheuristic application by their first implementation, parameter tuning capability, exploitation, and exploration scope techniques in combinatorial optimization.

The selected metaheuristics for this survey are the Firefly Algorithm (FA) [6], Harmony search (HS) [7], Bat Algorithm (BA) [8], Cuckoo Search (CS) [9], Artificial Bee Colony (ABC) [10], Bacteria Foraging Algorithm (BFA) [11] and Non-dominated Sorting Genetic Algorithm-2 (NSGA-2) [12]. These mentioned algorithms were filtered through their wide areas of implementation, citation index and hybridization capability in combinatorial optimization. Initially, every algorithm's basic framework would be discussed and later on their implementations in vehicle routing problems, travelling salesman problems and supply chain design problems are included in this survey.

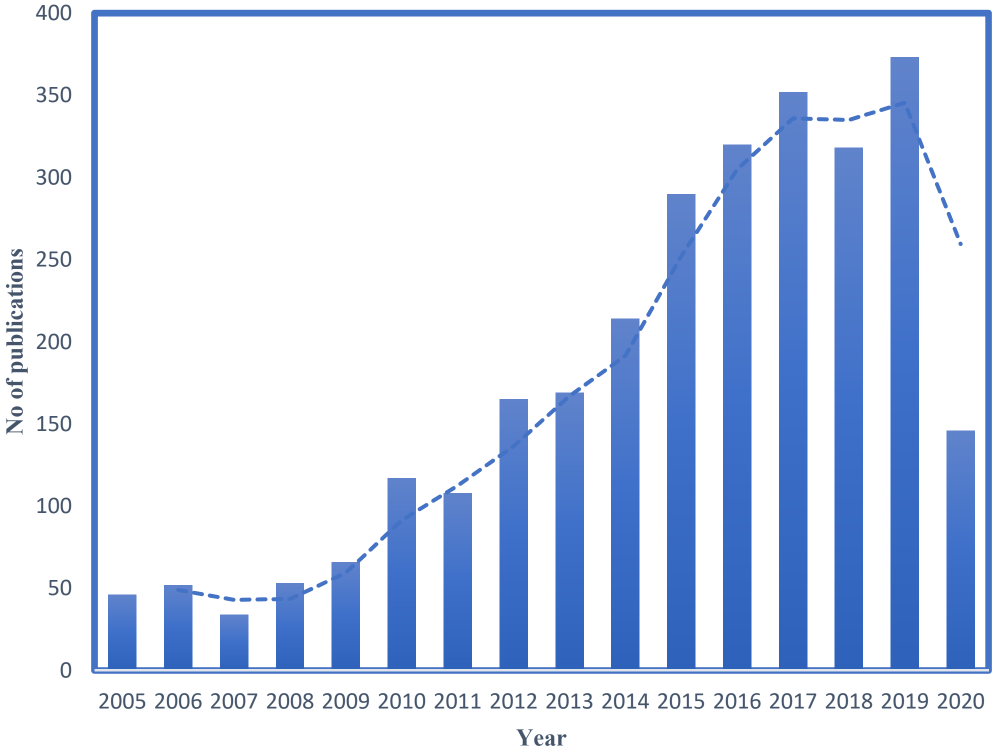

Figure 2. Citation reports of the papers related to the metaheuristics application in CO from 2005–2020 (Web of Science).

Table 1 represents the list of relevant review papers published in the span of 2005–2020. Ramos-Figueroa et al. [13] described only a short review of group combinatorial problems and it only focused on classical evolutionary metaheuristics. However, at present times numerous new generation swarm metaheuristics are being evolved. Rachih et al. [14] and Eskandarpour et al. [15] constructed a review on supply chain design problems and reverse logistics with swarm classical metaheuristics implementation. Analyzing the previous review papers, the authors found only one paper that described the new generation of metaheuristic applications. Dokeroglu et al. [5] illustrated the recent trend of metaheuristic implementation, parameter tuning and hybridization opportunities with a basic framework for global optimization. To our knowledge, the review paper list in the table describes the fact that there is no publication that concentrated only on the modification strategies of metaheuristic application in vehicle routing, travelling salesman problems, and supply chain design problems.

Table 1. Recent literature on combinatorial optimization problems.

Authors	Scope of the Study	Publisher
[13]	Group combinatorial problems and metaheuristic	Elsevier
[14]	Metaheuristics for reverse logistics	Elsevier
[5]	Review on recent trend of metaheuristic	Elsevier
[15]	VRP and hybrid metaheuristic	Elsevier
[16]	SCND and optimization-oriented review	Elsevier
[17]	Location routing problem and metaheuristics	Elsevier

The motivation to conduct this research is that most reviews did not include a detailed analysis of the way in which metaheuristic algorithms are customized to address complex combinatorial optimization issues in terms of a diversification strategy, limit handling techniques, local searching techniques, as well as hybridization techniques. Considering the lack of proper study on current advancement of metaheuristics in combinatorial optimization problems, the main contributions of the research are as follows:

- Provide insight about the recent advancement on the selected seven metaheuristic algorithms with algorithm framework, parameter tuning methods.
- Conduct a comprehensive analysis of mentioned algorithms in vehicle routing, travelling salesman problem and supply chain problems with initial population strategy, and cost function description.
- Finally, detail a critical discussion and observation about the impact of the local search technique, constraint handling strategy, hybridization on optimization performance and performance evaluation with classical metaheuristics.

The rest of this paper can be demonstrated as follows: systematic literature review and a background of combinatorial optimization are described in Sections 2 and 3, consecutively. In Section 4, insights in terms of the metaheuristic application in combinatorial optimization are included. In Section 5, observations and discussions of this study are outlined. Finally, Section 6 describes the conclusion and future research prospects of this research.

2. Systematic Literature Review

The main goal of this section to represent a systematic review process through which the published literature has been refined for application of metaheuristics for combinatorial optimization.

As the first phase, it is required to sort out the literature relevant to the main topic. We applied a handful of keywords e.g., “metaheuristics” AND “combinatorial optimization”, “Algorithm name (such as FA, BA etc.)” AND “VRP”, “Algorithm name (such as FA, BA etc.)” AND “TSP”, “Algorithm name (such as FA, BA etc.)” AND “SCND”. As a search engine, we used the popular journal core databases Scopus (www.scopus.com accessed on 10 August 2021) and Web of Science (http://apps.webofknowledge.com accessed on 10 August 2021) related to the fields of abstract, title and keywords. Consequently, we limited our literature with the search fields of language (English), paper type (article or review) and the span of 2005–2020. In addition, we searched the literature on trusted and widely used journal in operational research such, as The European Journal of Operational Research; Computers and Industrial Engineering; the Journal of Production Research; Computers and Operations Research; the Journal of Cleaner Production. The whole process is illustrated in Figure 3.

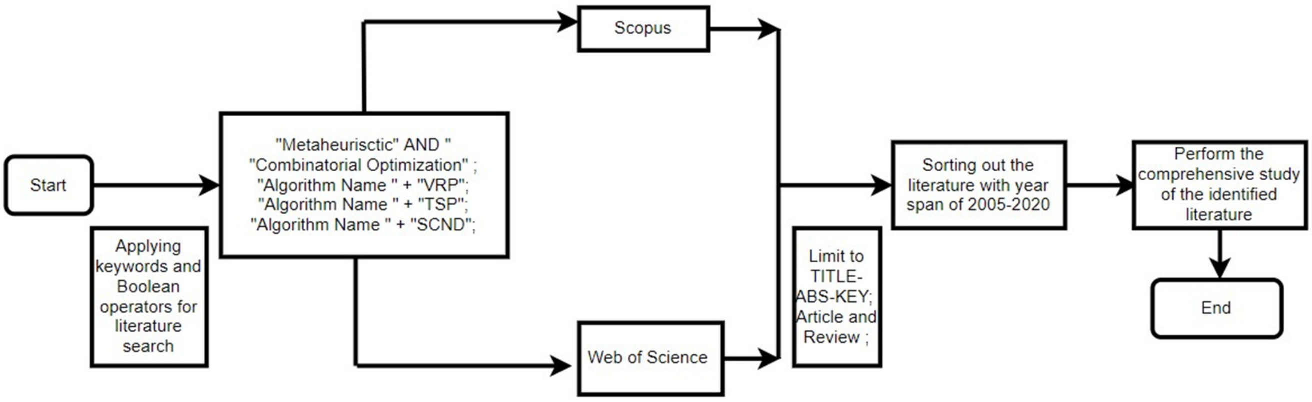
Start
Applying keywords and Boolean operators for literature search
"Metaheurisctic" AND " "Combinatorial Optimization" ; "Algorithm Name " + "VRP"; "Algorithm Name " + "TSP"; "Algorithm Name " + "SCND";
Scopus
Web of Science
Limit to TITLE-ABS-KEY; Article and Review ;
Sorting out the literature with year span of 2005-2020
Perform the comprehensive study of the identified literature
End

Figure 3. Systematic literature review.

Overview of Collected Research Papers

In this stage, the collected literature was segregated into the number of publications in combinatorial optimization. The pie chart in Figure 4 represents the total publication related to CO problems, e.g., VRP, TSP, SCND, according to the SCOPUS database. From Figure 4. it has been clearly observed that Artificial Bee Colony algorithm (ABC) and Non-dominated Sorting Genetic Algorithm (NSGA-2) have been implemented in a large no of publications.

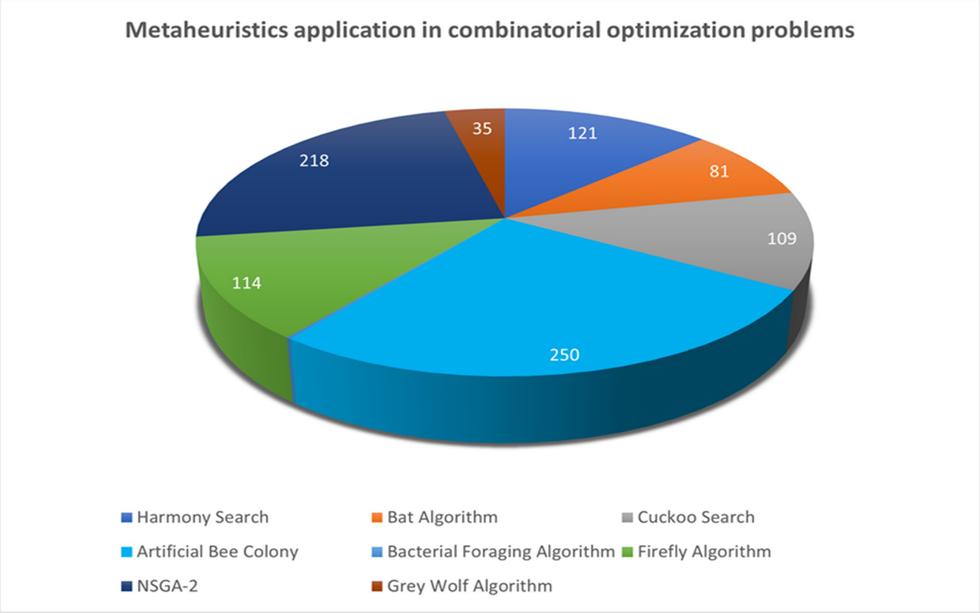

Figure 4. No of papers with relative metaheuristic technique used in combinatorial optimization (SCOPUS Database).

In vehicle routing problems, NSGA-2 has been adopted in 148 publications. Moreover, in travelling salesman problem Firefly Algorithm has the most number of application. Supply chain design problems have been tackled mostly through NSGA-2 (145 publications) and ABC (41 publications).

The bar chart in Figure 5 has a statistical representation of the metaheuristics application in CO problems according to the Web of Science database.

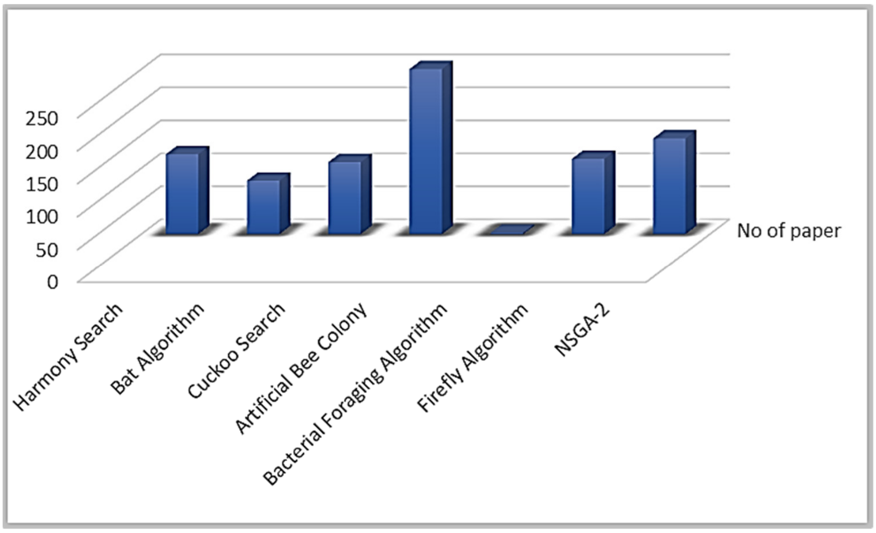

Figure 5. No of papers with relative metaheuristic technique used in combinatorial optimization (Web of Science Database).

3. Background

This section includes the background and structural definition of combinatorial optimization problems.

3.1. Combinatorial Optimization Problems

Combinatorial optimization problems (COP) are a special branch of optimization problems having a finite search space with discrete decision variables. Although the search space is finite, it's hard for an exhaustive search [18]. The mathematical expression of COP can be illustrated as:

$$minimize\ d(x) \tag{1}$$

Subject to:

$$h(x) \geq c$$

$$x \geq 0, x\epsilon f$$

where $h(x) \geq c$ and $x \geq 0$ denotes the constraints of a convex polytope around the objective function $d(x)$ is to be minimized under f finite set of feasible solutions. Numerous NP hard problems like vehicle routing problem (VRP), travelling salesman problem (TSP), supply chain design optimization (SCND) problems are often formulated as combinatorial optimization problems.

The following section includes the basic definitions and structures of CO problems, specifically VRP, TSP, SCND.

3.1.1. Vehicle Routing Problem

Vehicle routing is an indispensable part of logistics and every production line that controls how a fleet of vehicles can effectively serve warehouses, retailers and customers. As such, it is a state-of-the-art problem of combinatorial optimization. Moreover, vehicle routing problem has been tackled by metaheuristic algorithms in 41% of research papers, whereas exact methods and hybrid algorithms were adopted in 16% and 14% respectively [19].

Vehicle routing has some specifications while modelling. Each step can be demonstrated by following key points.

- In a known geographical location, every group of customers must be served by a fleet of vehicles from a common single depot.
- In a group of customers, everyone should have demand specification.
- Every route has a starting point and a terminal point, and it should be concise into the depot.
- The main objective is to find the shortest or most efficient path where transportation or total operative cost is minimum as well as the number of routes.

This general VRP can be illustrated by the following mathematical expression:
Minimize

$$f(X) = \sum\nolimits_{i=0\ i\neq j,} \sum\nolimits_{j=0} d_{ij} x_{ij}, \forall i, j\epsilon V \tag{2}$$

With respect to:

$$\sum\nolimits_{i=0\ i\neq j} x_{ij} = 1, \forall j\epsilon V \tag{3}$$

$$\sum\nolimits_{i=0\ i\neq j} x_{ij} = 1, \forall i\ \epsilon\ V \tag{4}$$

$$\sum\nolimits_{i} x_{ij} \geq |S| - v(s) \tag{5}$$

$$\sum\nolimits_{i\epsilon S} q_i y_i^r \leq C, \forall r\ \epsilon K \tag{6}$$

where:

$$y_i^r\ \epsilon\ \{0,1\}, \forall r\ \epsilon\ K \tag{7}$$

$$x_{ij}\ \epsilon\ \{0,1\}, \forall \{i,j\} \epsilon A$$

and d_{ij} = distance between i routes and j routes.

First, Equation (2) represents the objective function that describes the total distance covered by all vehicles. In Equation (7), it declares a binary variable that gives the value 1 or 0 for the customer satisfaction done or not. In the Equations (4) and (5) states that one vehicle is allowed to visit one client for a single time. Equation (5) stands for the dismissal of subtours between every vehicle to the client. Above all, the capacity constraint has been declared by the Equation (6), which assures the demand must not exceed the capacity of vehicle.

3.1.2. Travelling Salesman Problem

The core objective of Travelling Salesman Problems can be discussed by this way.

- Basic concept is to minimize the distant factor between cities visited by salesman.
- During the movement, every city should have single time visit.
- Starting point and the ending point should be the identical location.

TSP can be represented mathematically by following equations:

$$Minz = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{8}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i = 1, 2, \ldots\ldots, n \tag{9}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j = 1, 2, \ldots\ldots, n \tag{10}$$

$$x_{ij} = \{0, 1\}$$

Here d_{ij} = distance between i city and j city.

$$d_{ij} = \infty \text{ when } i = j \text{ and } d_{ij} = d_{ji}$$

Here, Equation (8) illustrates the objective function that denotes the total distant factor covered by a salesman. In Equations (9) and (10), represents the binary variable which states the value of 0 or 1 for the salesman visit has executed or not.

3.1.3. Supply Chain Network Design

The supply chain is considered to be an integral part of operational research. The generic supply chain represents the flow of products, services, and business entities from the supplier warehouse to the customer level in Figure 6. In this era of industrialization, every company or production process has to tackle the efficient flow of its product line to deliver it at the final stage. Eventually, the supply chain network problem has evolved to find the optimum distribution of entities with respect to unavoidable constraints. It can be stated that a huge portion or 80% of SCND problems can be categorized as facility location problems, as the efficient flow of products largely depends on the location of the facilities. This definition of facility location concludes that companies function at multiple levels within the supply chain, from factories, warehouses, delivery facilities, retail shops, clients, and the like. In a broad context, what is ideal for design, information on supply and customer locations, the location of potential locations for facilities, customer/market demand forecast, facility, labor and materials costs per facility, inventory costs per facility, transportation costs, amongst each pair of facilities, product selling prices, taxes, tariffs, customer reaction times, should be considered while modeling [20].

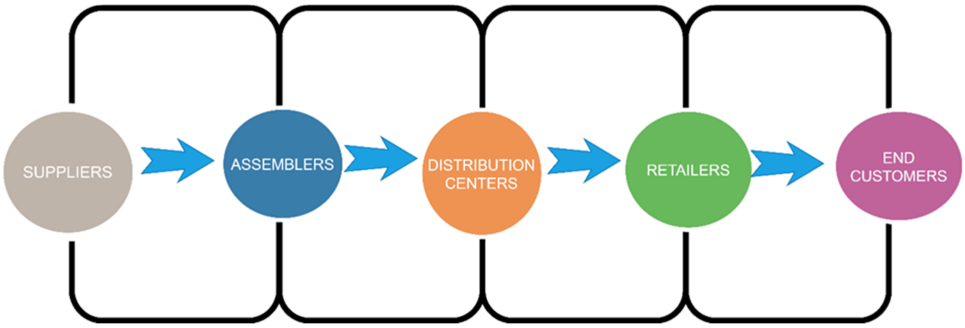

Figure 6. Framework of classical supply chain network design.

4. New Generation Metaheuristic Algorithms

The following section describes the core framework of selected metaheuristic algorithms and their implementation in CO problems.

4.1. Firefly Algorithm (FA)

A new population-based metaheuristic in resolving questions related to continuous optimization is the firefly algorithm (FA). FA was inspired by the simulation of the social light flashing behavior of the fireflies. Fireflies emit light and their light intensity variation differentiates among each firefly's identity.

The firefly algorithm follows these rules:

- Every firefly will attract each other beyond considerations of sex.
- A firefly having high brightness can attract a less bright firefly, and a brighter firefly has the scope to move to the next position quicker than the less bright firefly. Eventually, the objective function can be considered as the light intensity of a firefly.

$$I_x \ \propto TC_x \tag{11}$$

Here,

$$C_x = \cos\text{t function and } I_x = \text{light intensity}$$

- For two fireflies where the first one has less light intensity than the other, the first firefly moves in the direction of a brighter firefly and updates the solution. The following formula can describe this event:

$$x_i' = x_i + \beta_0 e^{-\delta r^2}\left(x_j - x_i\right) + \alpha\epsilon \tag{12}$$

where x_i' = new position of firefly, α = mutation coefficient (takes the value 0–1); β = attraction co-efficient (normally the value is 1; δ = light absorption coefficient (value range is 0.01–100)). These values are recommended by the inventor of FA [21].

FA has the ability to categorize fireflies into several small groups due to its inherent subdivision capability. This is the main reason why FA can handle multimodal and nonlinear problems so well. FA's position updates the direction; on the other hand, it is mostly single-dimensional as well as diagonal. In fact, FA cannot search in a regional direction, resulting in increased computational complexity and poor diversification [22]. So, several researchers have adopted modified FA with incorporation of encoding mechanism in FA's classical framework. For example, Trachanatzi et al. [23] designed a modified firefly algorithm on a VRP to find maximum prize collection with a minimum carbon emission factor. This modified FA has outperformed Differential Evolution (DE), Bat Algorithm (BA) and Gurobi optimizer claimed by the authors.

The basic framework of FA is illustrated in Figure 7.

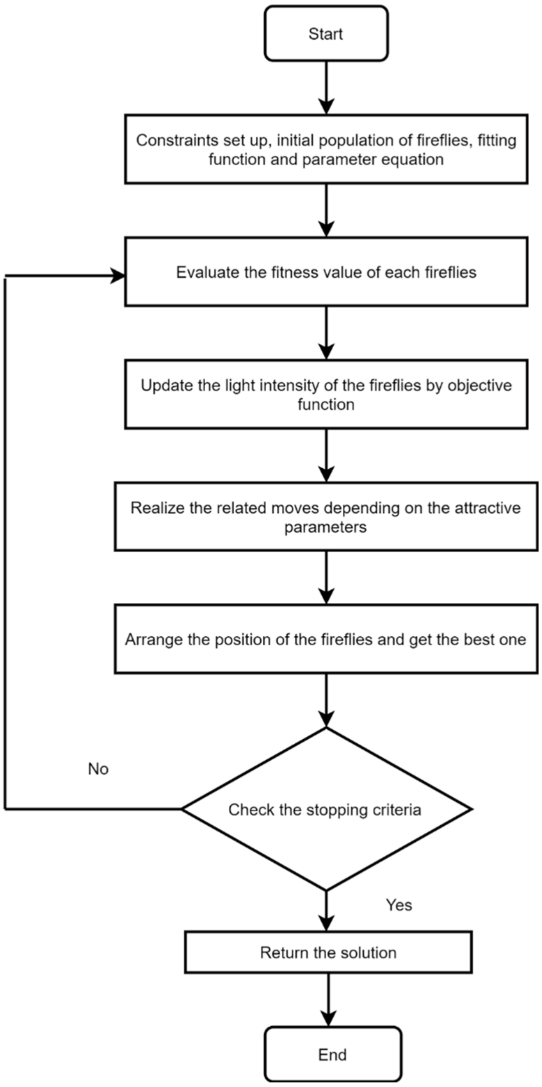
Start
Constraints set up, initial population of fireflies, fitting function and parameter equation
Evaluate the fitness value of each fireflies
Update the light intensity of the fireflies by objective function
Realize the related moves depending on the attractive parameters
Arrange the position of the fireflies and get the best one
No
Check the stopping criteria
Yes
Return the solution
End

Figure 7. Flowchart of the firefly algorithm [24].

Application of FA in CO Problems

In order to optimize the capacitated VRP, Goel and Maini [25] used a hybrid algorithm HAFA where FA worked as an exploration tool and pheromone shaking approach in ant colony algorithm helped to avoid local optima. It was claimed that the proposed HAFA outperforms EDFA, ACS and GA in finding optimal solutions for the same no of iterations. For a time-dependent VRP, Alinaghian and Naderipour [26] presented an improved firefly algorithm (IFA) based on the Gaussian Firefly Algorithm, considering the constraint of fuel consumption. Authors have used two heuristics for random initialization, which accelerates to a good variety of solutions. Later on, a comparative analysis on three metaheuristics, e.g., IFA, GD-FF, FA, demonstrated that IFA had shown less mean percentage error than the aforementioned algorithms. An enhanced FA was designed by Altabeeb et al. [27] for tackling a CVRP problem that was intercorporate with a two-stage local search strategy. Moreover, the authors have implemented a crossover and mutation operator to find better exploration and exploitation. A performance evaluation has been executed with respect to several benchmark functions and it exhibits that the proposed algorithm (CVR-FA) was enabled to produce the best-known solution (BKS) than the capacity using the genetic and firefly algorithm (CFAGA).

Atabaki et al. [28] designed a multi-stage closed-loop supply chain model considering forward and reverse flow of product with price-sensitive demand metric. That profit-maximization model was structured as a Mixed Integer Linear Programming (MILP) model initially. Consequently, authors have adopted an encoding–decoding strategy to modify the firefly algorithm and compared the results of proposed algorithm's performance with business solver GAMS/CPLEX. In the numerical experiment in the CPU metrics, PBFA attained the optimum solution much quickly than GAMS/CPLEX solver against large instance dataset. In addition, Table 2 represents some of the selected literature of FA with their parameter-setting and how parameter tuning has made an impact on the performance of the proposed algorithm.

Table 2. Parameter-setting of some selected literature for FA.

Authors	Proposed Algorithm	Parameters				Algorithm/Function Comparison	Optimality
		Max Gen.	Gamma	Beta	Alpha		
[29]	ACO-FA	50	1	1	0.2	SZGA and DRASET	47.5% better computational efficiency
[30]	FA	100	0–10	0.02	0.05	Benchmark function evaluation	84%
[31]	Traditional FA	30	0.01–0.15	0.02	0.05	Travelling Salesman Problem	88%
[32]	FASPA	500	1	0.2	0.5	Traditional FA	FASPA exhibits better convergence than FA
[33]	Hybrid FA-GA	100	1	1	0.2	Traditional FA and GA	Hybrid approach shows 7 times better computational efficiency than traditional method
[34]	Discrete FA	100	0.5–1	0–1	0–1	SA, ACO, GA, CS, PSO	DFA shows less mean deviation

Khalifehzadeh and Fakhrzad [35] discussed a multi-stage Product Distribution Network (PDN) model with a demand uncertainty factor. The authors have implemented a heuristic-based firefly algorithm where the customer demand was treated with fuzzy logic. It was claimed that heuristic-based selective FA (SFA) had outperformed classical FA in improvement percentage index for creating optimum solution. Tokhmehchi et al. [36] addressed a closed-loop supply chain network problem concerning reversed logistics with a cost-minimization approach. Authors have implemented an interesting strategy to hybrid firefly algorithm and genetic algorithm, which provides a better solution for the optimization. Later on, the authors stated that an Analysis of Variance (ANOVA) test justified their hybridization technique to be successful.

Saghaeeian and Ramezanian [37] illustrated a comparative study between two rival companies supply chain design which includes multi-stage and multi-product competition in a duopolistic market. Authors investigated how demand of products could be affected by the competitiveness between two companies while distributing multi-products. They adopted a unique strategy called Matheuristics, where the initial problem had been solved by hybrid GA–FA algorithm and later on a business solver named GAMS continue the optimization. In that Matheuristic algorithm, authors have implemented a priority-based GA encoding technique as a local search and introduced a four-stage crossover method to execute the optimization. The numerical results stated that the proposed hybrid genetic algorithm (HGA) outperforms traditional GA in the optimality index.

4.2. Harmony Search Algorithm (HS)

The harmony search concept is derived by musicians looking for the best harmonious result in harmony improvising. The improvisation method of musical harmony usually comprises three steps: first, selecting a suitable pitch of music from the memory of the musicians; second, a pitch of music related to a reasonable pitch that is slightly tweakable and formulating a new or random pitch. The aforementioned musical harmony improvisation strategy was first discussed by Geem et al. [7]. Similar to other nature-inspired metaheuristics, HS also replicates the elitist rule. A group of better-quality harmonies is considered, and weak sets of harmony are withdrawn from the process. The process continues until any optimum harmony is attained. The HS algorithm mechanism can be demonstrated by the following steps: initialization of harmony memory, formulating new vectors for the new solution (applying a two-stage probability-based harmony memory). The HS algorithm's core tweakable factors are pitch adjustment probability (PAR) along with the retention probability of harmony memory [38]. Here, the PAR ensures the local search capabilities of the vectors around sub problem space. Eventually, the recommended value of PAR has been stated as 0.1–0.5 and for HMCR is 0.7–0.95 by Geem et al. [7] for a wide number of cases. Moreover, retention probability index HMCR has an important impact on optimality finding for HS algorithm. It has been observed that the higher number of HMCR leads to having a better probability for searching optimal solution [39]. The small value of HMCR tends to provide precise solutions, but it takes higher computational time to provide that solution. This is why choosing a balanced value of HMCR can really impact the performance of HS algorithm in different problem instances. Figure 8 shows a visual representation of the harmony search algorithm mechanism.

Application of HS in CO Problems

Z. Guo et al. [40] discussed the integrated output and transport scheduling issue in the MTO supply chain. To address this issue, a harmonic search-based memetic optimization model is developed, in which certain heuristic procedures are proposed to turn the issue under investigation into an order assignment problem. A novel improvisation method is also suggested to improve optimum efficiency. Numerical tests demonstrated the validity of the proposed model in comparison with genetic algorithms and other nature-inspired algorithms.

In order to describe manufacturing, Z. X. Guo et al. [41] addressed a multi-objective multi-site order planning issue, recognizing various real-world aspects, such as production challenges and learning impacts. A new harmonious multi-objective optimization search algorithm has been developed for this problem. The main component of this model consists of harmonic searches for Pareto optimization (HSPO) and a simulation framework for the Monte Carlo simulation. Based on the real market data, the efficiency of the proposed model is calculated by a set of tests. The numerical experimentation illustrates that HSPO process can generate performance superior to those generated by a multi-objective genetic algorithm.

Eskandari-Khanghahi et al. [42] addressed a sustainable supply chain network problem for a blood bank. The authors considered three types of sustainability impacts on that model, in addition to the economic factors. As their model has several contradictory objective functions, they implemented a ϵ-constraint method to transform the multi-objective mathematical model to a mono objective model. Any testing questions are studied in order to verify the proposed model. For broad issues, a meta-heuristic algorithm is established to solve the model, namely simulated annealing (SA). Any numerical example is defined and evaluated, and the SA algorithm's efficiency is contrasted with the harmonic search (HS).

Yassen et al. [43] addressed a vehicle routing problem with the time windows factor. For optimizing the mentioned model, the authors implemented a novel meta-HAS approach which provided a better balance between exploration and exploitation. In this work, the researchers have introduced three renowned neighborhood strategies, such as the two-opt star, interchange and shift. It was claimed through numerical experimentation that meta-HAS outperformed standard HS and the other variant of HS in the standard deviation metric.

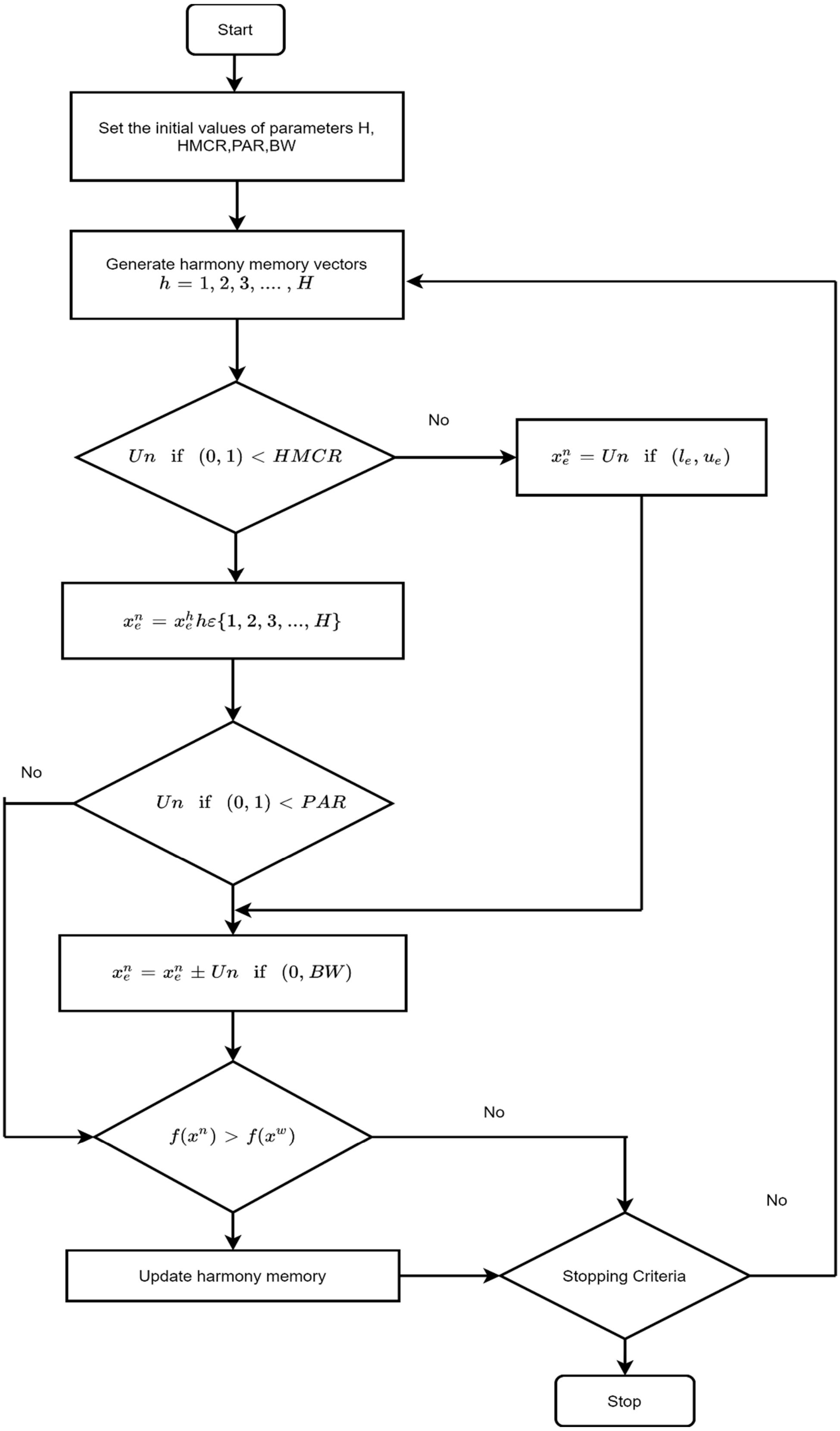
Start
Set the initial values of parameters H, HMCR,PAR,BW
Generate harmony memory vectors $h = 1, 2, 3, \ldots, H$
Un if $(0,1) < HMCR$
No
$x_e^n = Un$ if (l_e, u_e)
$x_e^n = x_e^h h \varepsilon \{1, 2, 3, \ldots, H\}$
No
Un if $(0,1) < PAR$
$x_e^n = x_e^n \pm Un$ if $(0, BW)$
$f(x^n) > f(x^w)$
No
No
Update harmony memory
Stopping Criteria
Stop

Figure 8. Framework of Harmony Search.

S. Chen et al. [44] constructed a modified HS for a classic combinatorial problem named the Dynamic Vehicle Routing problem with Time Windows case where a customer request can be monitored in real-time. Incorporating with the Variable Neighborhood Descent (VND) method, the local search strategy of HS has been modified. The proposed algorithm efficiency was examined upon Lackner benchmark instances along with two popular methods Improved Large Neighborhood Search (ILNS) and General Variable Neighborhood Search (GVNS). Harmony Search Variable Neighborhood Descent (HSVND) exhibits a good result in maintaining a balanced exploration and exploitation for the DVRPTWs instance.

Ruano-Daza et al. [45] demonstrated a multi-objective algorithm, the Multi-Objective Global Best Harmony Search (MOGBHS), for a transit network design and frequency setting problem in Columbia. This multi-objective model from that aforementioned research was sorted by two methods like the Pareto front and crowding distance method. That problem had two conflictive objectives, e.g., minimal lead time and minimum operational cost. Later on, experimental data showed that MOGBHS had provided a better true Pareto front better than the state-of-the-art multi-objective algorithms like NSGA-2 (Non-dominated Sorting Genetic Algorithm-2) and MOEA/D(Multi-objective Evolutionary Algorithm) in the scale of inverted generational distance metric and the Friedman and Wilcoxon statistical test.

Gao et al. [46] solved a flexible job scheduling problem considering the fuzzy processing time through a discrete harmony search algorithm. That problem consists of one objective, and it was to minimize the fuzzy processing time. Researchers initialized the search space population with a new heuristic method named MiniEnd, which leads to the Discrete Harmony Search into a comprehensive performance rather than the six metaheuristic methods while producing minimal execution time.

Kianfar [47] discussed a profit-based supply chain model where impacts of advertising and demand were taken into consideration. The authors have adopted two metaheuristic algorithms, e.g., SA and HS, to solve that proposed model. It was claimed that in the mean ideal distance metric (MID), SA achieved a better Pareto front than HS.

Hosseini et al. [48] addressed a transportation problem for the milk distribution industry with a cost-minimization approach. Initially, authors have constructed an integer programming model and applied an improved harmony search algorithm. Results obtained from different experiments described the superiority of adopted HS over business solver CPLEX in computational time and cost function metric.

Boryczka and Szwarc [49] modified the contemporary HS by adding the improving harmony memory with a view to surpassing the weakness in exploitation stage of HS. They implemented this modified strategy to optimize a traveling salesman problem. Consequently, the adopted HSIHM (Harmony Search with improving harmony memory) could enable a 59% better solution than the traditional HS.

4.3. Bat Algorithm

The Bat Algorithm is a nature memetic algorithm introduced by Yang (2010). This metaheuristic approach follows an echolocation-based technique including randomization, secondly updating the position along with comparison of the best outcomes.

Many of the metaheuristic techniques have a common criterion as they follow a randomization process to update the solution process in a certain dimension. Though in real life, bats move randomly to their food or prey; in a defined dimension, their randomization value becomes skeptical [50].

In every generation, the bats provide the information of frequency and velocity and prepare a new solution. The update of solution can be illustrated by the following equations

$$f_i = f_{min} + \beta(f_{max} - f_{min}) \tag{13}$$

$$v_i^t = v_i^{t-1} + \left(x_i^t - x_*\right) f_i \tag{14}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{15}$$

where f_{min} and f_{max} denotes the minimum and maximum frequency of the bats while v_i^t represents the velocity of a bat at time t. Eventually, the Bat Algorithm follows the echolocation technique, which means whenever a bat is nearer to a prey, the loudness factor A_i and pulse emission rate r is updated. The relation is vice versa whenever a solution update and loudness value (A) decreases, and the pulse emission rate increases at a time. The following equations can describe this phenomenon:

$$A_i^{t+1} = \alpha A_i^t \tag{16}$$

$$r_i^{t+1} = r_i^0 \left(1 - e^{\gamma t}\right) \tag{17}$$

In the above-mentioned equations, α and γ are constants; A_i , r_i represents initial loudness and pulse rate consecutively. BA's working mechanism is represented in Figure 9.

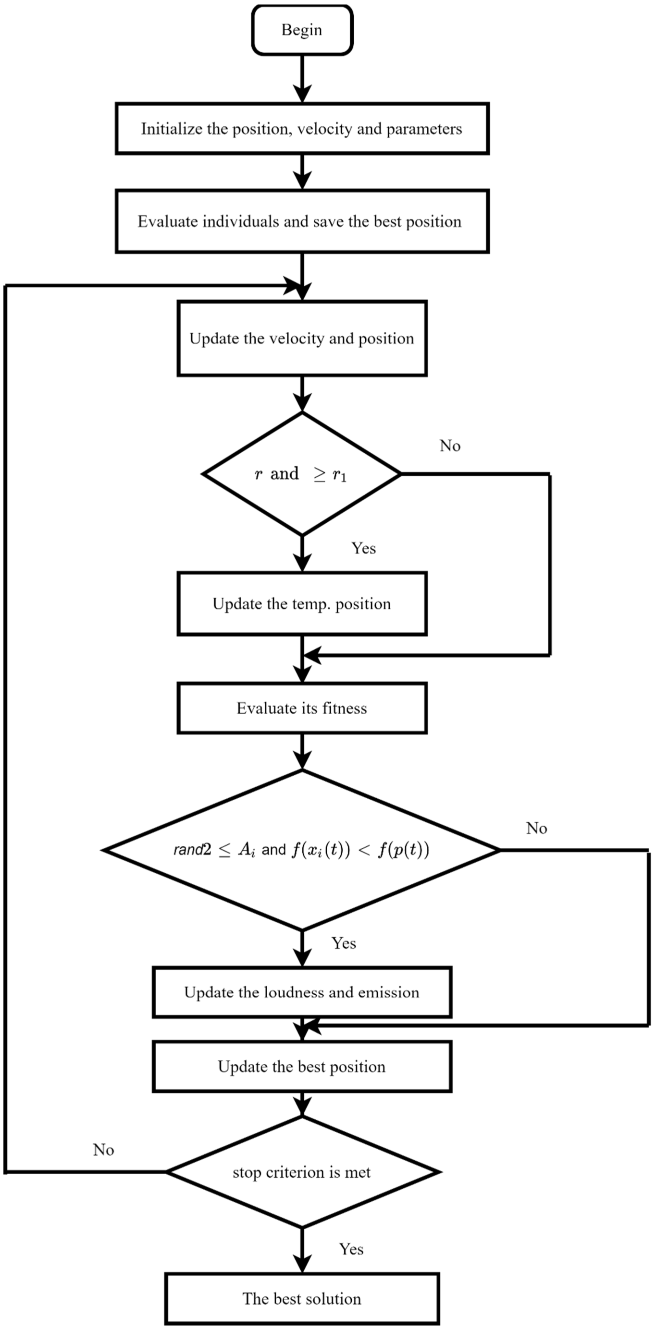
Begin
Initialize the position, velocity and parameters
Evaluate individuals and save the best position
Update the velocity and position
r and $\geq r_1$
No
Yes
Update the temp. position
Evaluate its fitness
$rand2 \leq A_i$ and $f(x_i(t)) < f(p(t))$
No
Yes
Update the loudness and emission
Update the best position
No
stop criterion is met
Yes
The best solution

Figure 9. Framework of the Bat Algorithm.

It has been observed that the Bat Algorithm has been evolved with several tuning in pulse frequency range factor with an application perspective [51]. Moreover, BA has a normal tendency to be trapped into local optima. To resolve this issue, researchers have implemented several manipulations in standard BA, the like levy flight technique, instead of a conventional step walk or swarm techniques in the local search stage and inertia weight for velocity update. Table 3 shows the pulse frequency setting preferred by different researchers.

Table 3. The different pulse frequency ranges implemented by the researchers for the Standard BA modification.

Authors	Pulse Frequency Range
[52]	0–1
[53]	0–2
[54]	0–5
[55]	0–100

Application of BA in Combinatorial Optimization

Table 4 presents a detailed analysis of the application for the BAT algorithm for CO issues.

Table 4. Bat Algorithm implementation in several CO problems.

Authors	Proposed Algorithm	Application	Objective Function	Algorithm Modification Approach	Remarks
[56]	Improved Bat Algorithm	Capacitated Vehicle routing problem	Minimizing the sum of vehicle travel routes	Dynamic inertia weight and time factor	Modified BA has shown better performance-optimizing benchmark VRP functions compared to standard BA
[57]	Bi-population-based Discrete Bat Algorithm (BDBA)	Low-Carbon Job shop scheduling problem	Minimizing the sum of energy consumption cost along with execution time cost	Parallel searching strategy leads to produce two sub population as well as a better solution	In average relative percent deviation metric BDBA gives best value among GA, Improved Whale optimization Algorithm (IWOA) and Single-population discrete bat algorithm
[58]	Hybrid Self-Adaptive Bat Algorithm (HSABA)	Emergency Location—Routing Problem	Minimization of resettlement cost of the warehouse and lead time for shipment	Implementing self-adaptive parameter tuning as well as the exploitation of standard DE algorithm approach "rand/a/bin" which enables to escape from local optima trapping issue.	Against the Solomon classic test dataset, proposed HSABA produced smooth convergence plot rather than standard BA
[59]	Self-adaptive Bat Algorithm	Truck and trailer routing problem	Minimizing the cost for route path	Self-adaptive technique helps the parameters to control searching process in problem space and concurrent objective function evaluation	SA-BA outperforms other algorithms in the metric of actual CPU average time

Table 4. *Cont.*

Authors	Proposed Algorithm	Application	Objective Function	Algorithm Modification Approach	Remarks
[60]	Discrete Bat Algorithm and large Neighborhood Search	Vehicle routing problem with time windows	Minimizing no of vehicles along with finding the shortest distance	Integrating the removal heuristic approach of large neighborhood search with discrete BA had proven to produce better exploration around the problem space	Proposed Discrete BA-LNS algorithm has been claimed to produce comprehensive solution in faster runtime (only 300 iterations) which has outperformed other state-of-the-art heuristic approaches in VRP
[61]	Discrete Bat Algorithm	Travelling Salesman Problem	Minimizing the distance among every node of a distribution network	Implementation of 2-opt crossover encoding in position update stage of Bat Algorithm	DBA has performed well than DPS and GSA with respect to solve benchmark functions of TSPLIB instances
[62]	Improved Discrete Bat Algorithm	Symmetric TSP	Minimizing the total traveling cost of route	Two different moving structures for bats in solution space leads to create better exploration	IBA is proven better than Discrete BA with respect to Student's *t*-test.
[63]	Multi-Population Discrete Bat Algorithm (MPDBA)	Travelling Salesman Problem	Minimizing the total traveling cost of route	Parallel population technique and crossover technique implementation had provided better exploitation around problem space	Proposed MPDBA outperforms other variants just like Improved BA, Evolutionary Simulated Annealing in convergence index against TSPLIB benchmark functions and percentage deviation index
[64]	Tuned Hybrid Bat Algorithm	Inventory model of a 3 echelon Supply chain	Optimizing machine count, minimum possible path along with minimizing total chain cost	Using PSO as a local search strategy moreover implementation of Taguchi method for parameter tuning of HBA for the first time	Comparison metric of system cost, best system reliability, HBA outperforms standard BA. Later on, another experiment proves HBA is better than standard GA is optimizing all factors

4.4. Cuckoo Search (CS)

The Cuckoo Search is an evolutionary algorithm that follows the trade-off between randomization and local search like other metaheuristics. Having a minimal parameter to work with it, CS has been widely used by the researchers as it needs less effort for parameter initialization. Compared to other population-based metaheuristics, for example, GA, PSO, and CS have shown improved performance for their simplicity in an algorithm-built nature [65].

The Cuckoo Search was evolved by X. S. Yang and Deb [66] through adopting an enhanced type of random walk, named the Lévy Flight, rather than isotropic random walk. Several nature creations have Lévy flight behavior and inventors were inspired by the performance of Lévy flights because it has infinite mean variance and has better exploration capabilities, rather than standard Gaussian processes [67].

Having a predatory reproduction strategy of some Cuckoo species, the Cuckoo Search is the representation of brood parasitism. In nature, some of the species do not build their own nests and rather than they choose nest from another species to camouflage their eggs to continue their reproduction. In contrast to other birds' laying eggs strategies, the Cuckoo Search contains its individual trick to identify their eggs.

The standard Cuckoo Search has three distinguished rules, which activates the following:

- Each cuckoo lays one egg at a time and puts it a nest selected randomly.
- The best nests would follow the elitist strategy and best quality nest will carry forward to the following generations.
- There would be a fixed number of available host nests and the egg discovery probability by the host bird can be expressed as $p_a \epsilon (0,1)$.

The Cuckoo Search follows an integration of a local random walk and a global explorative random walk where a switching parameter p_a guides the walk. The following equation can represent the local random walk:

$$x_i^{t+1} = x_i^t + \alpha \mathrm{s} \otimes \mathrm{H}(p_a - \epsilon) \otimes \left(x_j^t - x_k^t\right) \tag{18}$$

where x_j^t and x_k^t are selected different solutions are sorted by random permutation, ϵ is a random number from uniform distribution along with s represents step size. As per the second strategy of CS, the global random walk will follow the equation below:

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \tag{19}$$

It has been observed that when the step size scaling factor $\alpha = O\left(\frac{L}{100}\right)$, CS works more effectively and ignores excessive exploration [67]. A complete framework of CS algorithm will be represented by following Figure 10.

Applications of Cuckoo Search (CS) in CO

Tarhini et al. [68] modified a contemporary model of vehicle routing along with the application of the Cuckoo Search-based hyperheuristics. They attempted to distinguish a customer priority VRP model with the demands of every zone of a routing. Their adopted algorithm CS-based hyperheuristics have proven better compared to the modified Clark weight heuristics algorithm on real industrial data from Lebanon.

Rezaei and Kheirkhah [69] developed a multi-echelon supply chain network model considering three objectives, e.g., economic, environmental, and social requirements. Initially, a multi-objective MILP model was structured; later on, a multi-objective CS strategy was implemented on the aforementioned model. Experimentally, MOCS (Multi-objective Cuckoo Search) has shown the most efficient Pareto solution on the basis of computational time, mean ideal distance, and spacing thread with the comparison of Multi-Objective Imperialist Competitive Algorithm (MOICA) and even MOPSO (Multi-Objective Particle Swarm Optimization). Jamali et al. [70] used an inventory priority model for a supply chain network model with demand uncertainty and a cost-minimization approach. An improved hybrid cuckoo search algorithm along with genetic algorithm had implemented to solve that model. The proposed ICSGA (Improved Cuckoo Search Genetic Algorithm) has optimized that cost minimization problem to a better extent than individual GA, ICS and GA performance.

In considering a group of cuckoos, Ouaarab et al. [71] suggested a discrete version of the CS algorithm carrying out Lévy flights and modifying this algorithm with promising results to solve TSP problem. During this modification, authors have restructured the population as well as introduced a separate class of cuckoos. Therefore, the discrete cuckoo search algorithm could perform well with less no of iterations. Later on, the authors have demonstrated a numerical experimentation that proved the supremacy of proposed discrete CS over discrete particle swarm optimization.

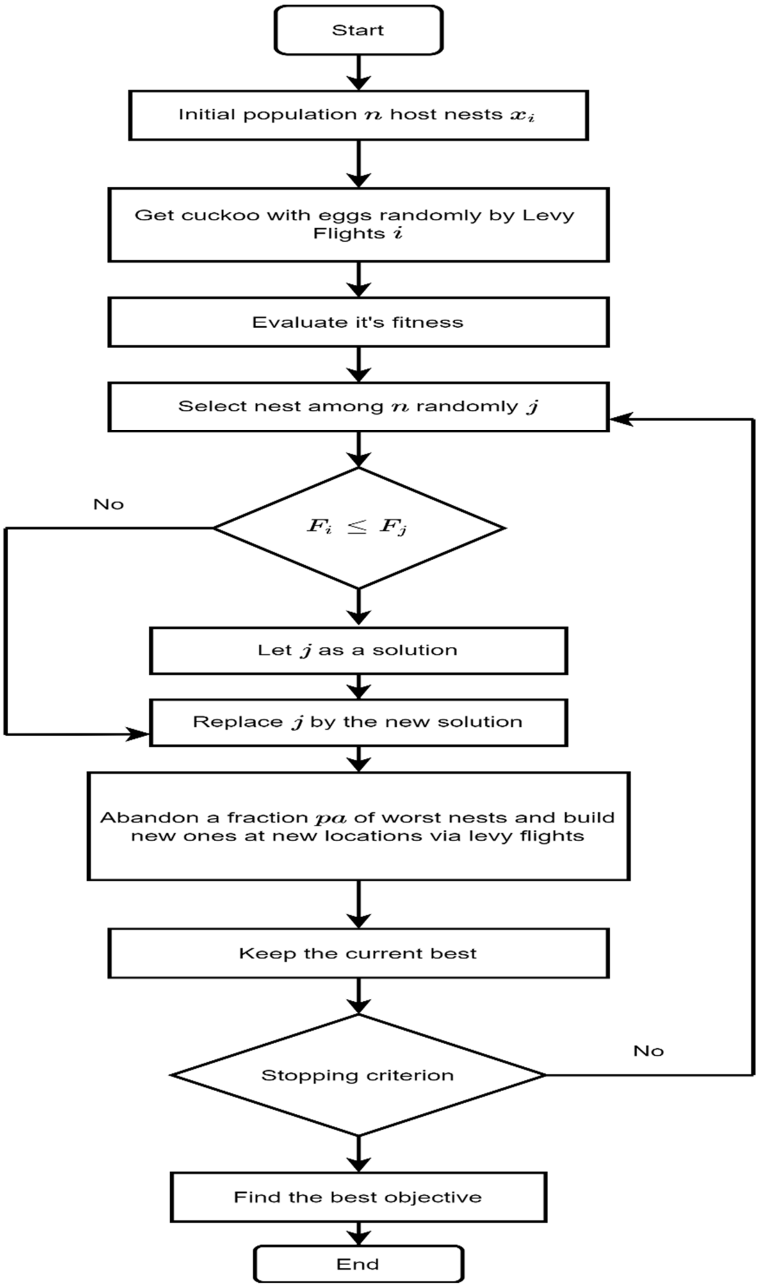
Start
Initial population n host nests x_i
Get cuckoo with eggs randomly by Levy Flights i
Evaluate it's fitness
Select nest among n randomly j
$F_i \leq F_j$
No
Let j as a solution
Replace j by the new solution
Abandon a fraction pa of worst nests and build new ones at new locations via levy flights
Keep the current best
Stopping criterion
No
Find the best objective
End

Figure 10. Framework of the standard Cuckoo Search (CS) Algorithm.

X. Chen and Wang [72] implemented a novel hybrid variant of cuckoo search algorithm in VRP, which was the integration of three algorithms, as examples, the Optical Optimization (OO), Particle Swarm optimization (PSO) and Cuckoo search. In respect of solution type, optimality and computational efficiency, HCS has demonstrated significant better performance than contemporary algorithms. Zheng et al. [73] suggested a modified variant of Cuckoo Search algorithm named GRASP-CS along with Greedy Random Adaptive Search Procedure to solve a VRP model. Later on, their experimental results demonstrated a significant performance on the benchmarking functions of VRP. Xiao et al. [74] implemented an improved CS consists of a split method to initialize parameters which gave a better initial solution for a patient transportation problem. Not only two heuristic methods for the initial population. In addition, they applied new cuckoo category to enhance searchability of the CS algorithm. Later on, selected strategy performance has been compared with three previously adopted popular methods of tackling VRP problems, for example, the Unsupervised Fuzzy Clustering approach (UFC), Large Neighborhood

Search (LNS) and hybrid approach of ACO named LNS (Large Neighborhood Search-ACO). ICS has shown better results than the aforementioned algorithms.

4.5. Artificial Bee Colony (ABC) Algorithm

The Artificial Bee Colony (ABC) algorithm simulates honeybee drinking behavior and was applied to many realistic problems. This optimization technique is used. ABC was proposed by Karaboga [10] and it is a part of the swarm intelligence group of metaheuristic algorithms.

The ABC algorithm works in a certain way by having three different categories of bees assigned for the execution. Consecutively, predefined bees tend to search around a food source or feasible solution. The onlooker bees measure the objective function values and sort out the food source with respect to nectar amount. This metaheuristic technique also has the elitist mechanism, as after a certain number of trials, if the function value remains stuck in the previous value, a new food source has been added by the scout bees to formulate a better solution for a certain objective function [75]. In brief, the ABC algorithm performs three phases in every evaluation as follows: (1) employed bee phase or local search phase, (2) onlooker phase, (3) conducting global search phase recognized as scout bee phase. Figure 11 describes the mechanism of the ABC algorithm.

Application of ABC in Combinatorial Optimization

Artificial Bee Colony (ABC) optimization has a widespread application in several combinatorial problems such as job shop scheduling, vehicle routing problem, traveling salesman problems and closed-loop supply chain optimization.

Researchers have implemented several strategies to enhance the performance of ABC. Yao et al. [77] developed an Improved Artificial Bee Colony (IABC) for the upgradation of initial search results of ABC for Job Shop Scheduling Problem (JSSP). Eventually, their adopted strategy can be formulated by mutation, which was used to widen the search space along with local optima avoidance. X. Li et al. [78] designed a hybrid ABC algorithm in accordance with the Tabu Search (TS) strategy for optimizing a flexible job shop scheduling problem. They deployed crossover operator in order to enhance the exploitation of the search space and experimentally, their implemented HABC performed better than the Tabu Search and Particle Swarm Optimization.

The ABC algorithm has two local search stages and an optional global search stage for every iteration it runs. Most of the researchers have tried different strategies in the local search stage. Meng et al. [79] intercorporate an updated Migratory Bird Optimization algorithm with ABC in order to balance global exploration and local exploitation.

Khan and Maiti [80] improvised the ABC through a perturbation technique, K-Opt, in order to solve the travelling salesman problem. Their adopted strategy has outperformed several state-of-the-art metaheuristic algorithms with respect to the traditional TSP (Travelling Salesman Problem) benchmark function. Pandiri and Singh [81] attempted an ABC-based approach to optimize a colored traveling salesman problem and experimentally illustrated the performance of their algorithm against traditional algorithms with respect to computational time and solution quality. Zhong et al. [82] introduced a hybrid discrete artificial bee colony algorithm for a traveling salesman problem along with a threshold acceptance criterion for scout bees in ABC algorithm phases. Later on, numerical experimentation has shown an impressive result on benchmarking function rather than traditional greedy acceptance criterion.

NP-hard problems, like the Vehicle Routing Problem (VRP), have been solved through ABC by several researchers. For example, Baradaran et al. [83] implemented a Taguchi-based binary ABC on a stochastic VRP model and produced better performance on standard test functions.

A closed-loop supply chain network model has been proposed by Cui et al. [84], considering uncertainty in demand within that network. They structured a Genetic Artificial Bee Colony algorithm approach in accordance with the Taguchi method. GABC's performance has outperformed the individual result of GA and ABC in optimizing the cost-based CLSC network problem. Lu and Jiang [57] implemented a gradient descent and Simulated Annealing-based ABC on a three-echelon complex supply chain network. These supplier and resource-based models are efficiently optimized by the Simulated Annealing-based ABC as the experimentation represents the better exploration on problem space for that complex SCN problem.

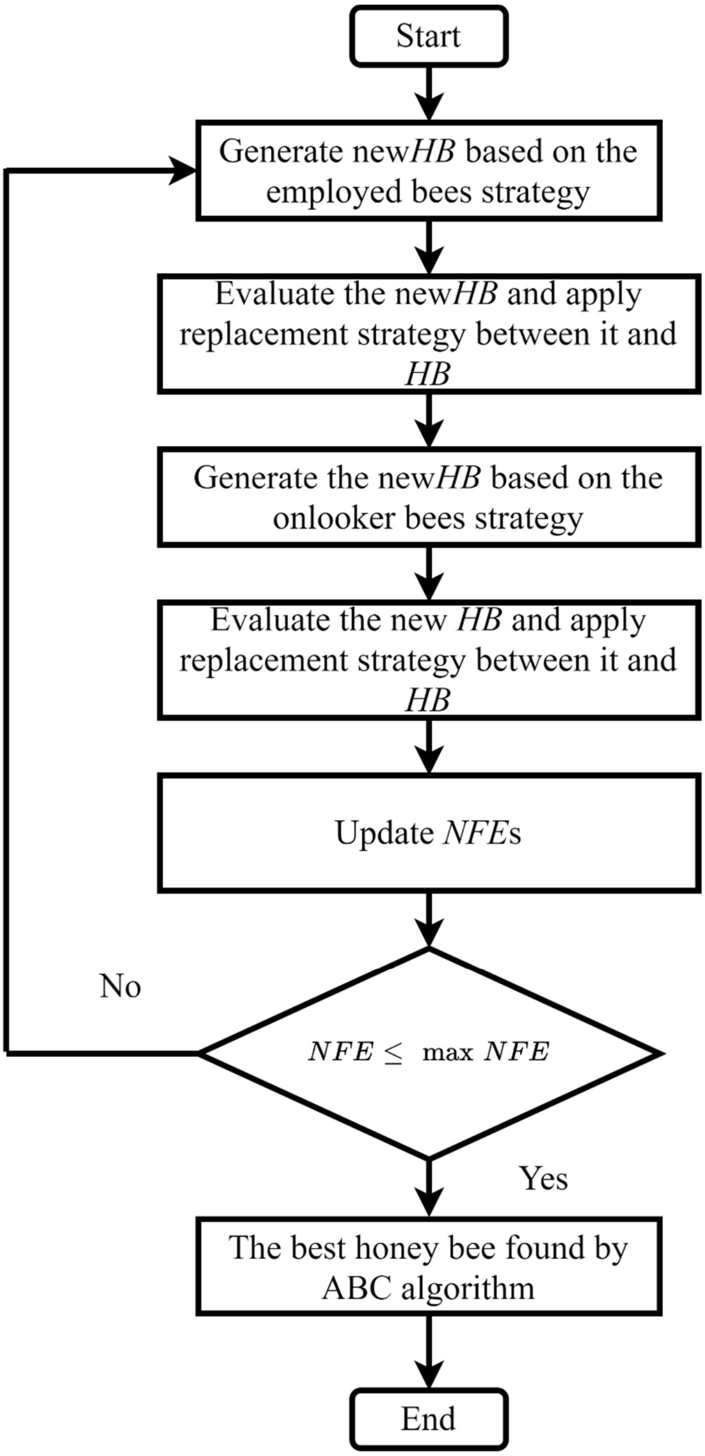
Start
Generate new*HB* based on the employed bees strategy
Evaluate the new*HB* and apply replacement strategy between it and *HB*
Generate the new*HB* based on the onlooker bees strategy
Evaluate the new *HB* and apply replacement strategy between it and *HB*
Update *NFE*s
$NFE \leq \max NFE$
No
Yes
The best honey bee found by ABC algorithm
End

Figure 11. Framework of Artificial Bee Colony Algorithm [76].

Szeto et al. [85] suggested ABC as a swarm-based heuristic for solving CVRP problem. They also proposed an enhanced version of their ABC techniques so as to improve the solution quality of original ABC.

4.6. Bacterial Foraging Algorithm (BFOA)

BFOA is structured on a group foraging strategy from *E. coli* bacteria swarm instances. Bacteria prefer to look for nutrients to thrive in a harsh environment in such a way as to enhance the energy obtained per unit time. A bacterial foraging algorithm was proposed by Passino [11] from the natural phenomena of bacteria swarms. During the foraging process, a group of tensile flagella initiates locomotion where *E. coli* bacteria can swim or tumble around the cell. While the direction of flagella is clockwise, that particular flagella hold strictly a cell around it with having lesser tumble movement. On the other hand, the counterclockwise direction enables quick swimming motion for the flagella's better exploration.

4.6.1. Classification

Normally, the standard BFAO has three stages to accomplish its mechanism can be concluded as chemotaxis, reproduction, and elimination and dispersal.

Chemotaxis

Chemotaxis is the first stage which translates the tumbling and swimming movement of the *E. coli* cell through flagella. The following equation represents this stage:

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + c(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{20}$$

where $c(i)$ is the run-length unit; $\Delta(i)$ represents a random direction vector. In every update of bacterial position, the fitness value $J(i,j,k,l)$ should be evaluated, here i,j,k,l denotes the position, chemotactic, reproductive and elimination, dispersal step consecutively.

Reproduction

In this stage, all bacteria follow the elitist rule of nature. Bacteria are categorized in chronological order with respect to their health status. Half of healthy bacteria can replace the remaining weaker ones being a double number of healthier bacteria. Strategically, BFO's successful operation largely depends on how the unhealthy bacteria are selected [5].

Elimination and Dispersal

The final step of BFAO illustrates the change of environment surrounding bacteria. In this method, a given probability is used to determine whether or not the bacteria disperse. If a single bacterium fulfils the criteria, it is transferred in the atmosphere to another location. The whole BFO algorithms standard framework is illustrated in Figure 12.

Similar to other population-based metaheuristics, BFOA was widely implemented in several complex instance problems, such as [87] for a load dispatch problem, for portfolio optimization considering liquidity risk [88], and eventually in multi-objective optimization [89], where researchers found BFAO has shown impressive performance over NSGA-2 in the aspect of computational time and producing the true Pareto front.

4.6.2. Application BFOA in Combinatorial Optimization

Liu et al. [90] constructed a comparative study among Improved Bacteria Foraging Algorithm having five-phase-based heuristic (FPBH) and four combined metaheuristics (MBFA, CBFA, HBGA, HBSA) method for optimizing a supply chain network model. The production planning along with facility transfer condition these two vital stages of a SCN were made into consideration. Experimentally, IBFA incorporated with FPBH has outperformed mentioned four algorithms in creating a better initial solution. Consecutively, implemented modified strategy on the stages on IBFA has widened the chance to select healthier bacteria in each generation.

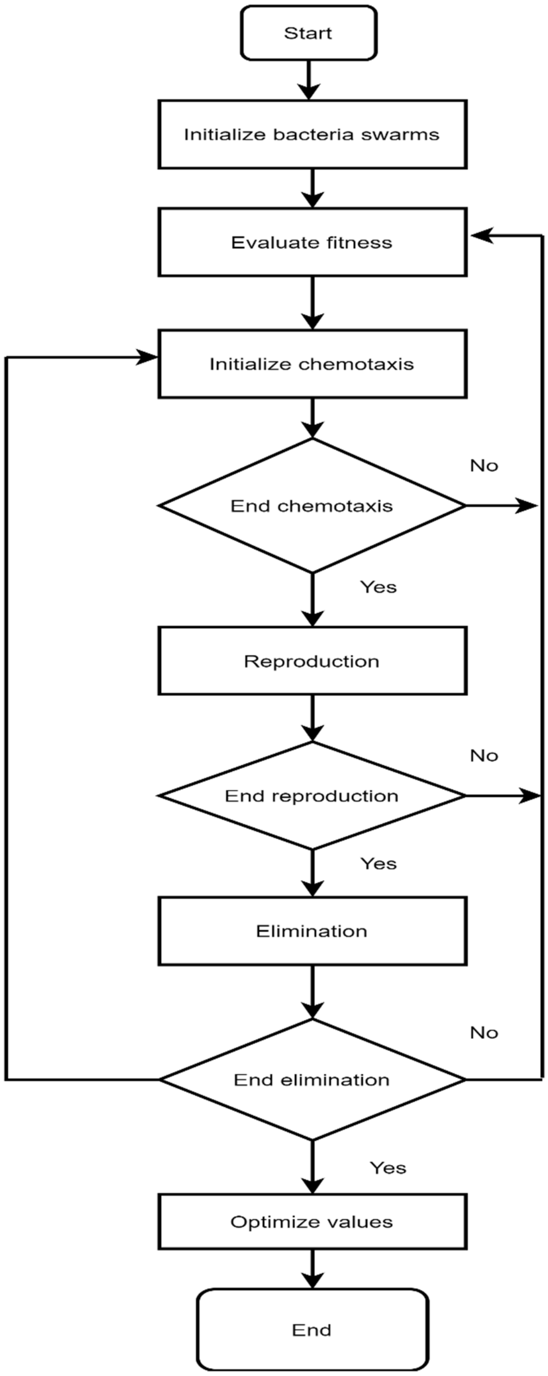

Figure 12. The framework of BFOA [86].

An established NP-hard problem, named the Supply Chain Network Model for Perishable Products (SCNMPP), was tackled by Sinha and Anand [91] with the improved bacteria foraging algorithm approach. Dealing with the cost minimization objective of that model, IBFA has minimized the transportation cost along with total cost in a better approach rather than standard BFA.

4.7. NSGA-2 (Non-Dominated Sorting Genetic Algorithm-2)

NSGA-2 has been the most profound algorithm with respect to its application in multi-objective optimization. Basically, NSGA-2 was the updated version of NSGA, which was first introduced by Deb et al. [12]. Moreover, NSGA-2 was designed to overcome the difficulties faced by the NSGA algorithm, such as more computational complexity, parameter tuning difficulty, and the absence of perfect elitism. Due to the robustness of NSGA-2 while optimizing complex problems, it has become the most popular approach in MOO.

NSGA-2 works on a random population and the solution gets enriched by the chronological iterations. The progress over iteration can be expressed by the change of fitness value in each iteration. Later, the population tends to create the Pareto front with non-domination sorting technique (where the first Pareto can be considered as smallest rank, and it is updated as the cardinal way). Every single solution of the population can be expressed as chromosomes that can hold binary values or real values. Then, each member's Pareto front distance is calculated by the linear distance metric. NSGA-2 follows the elitist mechanism, which is why the crowding distance and both ranks should be measured in each population by the selection operator. While selecting two members, the least ranked member was counted. However, if two members hold the same rank, the member with the better crowding distance gets selected.

Normally, NSGA-2 adopts the same stages as Genetic Algorithms, such as selection, crossover, and mutation. Initially, an offspring population, *Q-t*, is formulated from the parent population, *P-t*. To build an intermediate population of 2N, the offspring population gets integrated with the parent population. In every iteration, fitness value is calculated according to the objective function value. Later on, multiple members are selected through several selection criteria. This process continues until the termination condition is occupied. At the end of the whole process, a set of nondominant Pareto solutions are formed, which are the best at every aspect of a multi-objective optimization. The NSGA-2 algorithm is illustrated in Figure 13.

Application of NSGA-2 Algorithm in Combinatorial Optimization

The Non-dominated Sorting Genetic Algorithm (NSGA-2) has been adopted in numerous cases in multi-objective combinatorial optimization. In Table 5, the most recent implementation of the Non-dominated Sorting Genetic Algorithm (NSGA-2) has been discussed with various modification strategies.

Table 5. Selected literature on NSGA-2 implementation in CO.

Authors	Proposed Algorithm	Application	Objective Function	Algorithm Modification Approach	Remarks
[92]	NSGA-2 and NRGA	Multi echelon supply chain design problem	Cost-based modelling considering uncertainty	Both NSGA-2 and NRGA have modified with three-section encoding strategy	For performance evaluation, both algorithms were engaged with Simple Additive Weighting Test. NSGA-2 performed better than NRGA in every instance
[93]	Multi-Objective Biogeography-based optimization	Bi-objective supply chain network model	Minimizing Supply chain Cost and average tardiness	A Taguchi method was used to tune the MOBBO parameters	A statistical method ANOVA (analyze of variance) was implemented for performance analysis. MOBBO created better Pareto front than NSGA-2 and MOSA.
[94]	MOICA (Multi-Objective Imperialist Competitive Algorithm)	Bi-objective location- routing-inventory model with uncertain demand	Minimizing total cost along with maximum mean time for product delivery	Response Surface Methodology was implemented for parameter tuning	MOICA exhibits better performance than NSGA-2, MOPSO against 30 test problems

Table 5. *Cont.*

Authors	Proposed Algorithm	Application	Objective Function	Algorithm Modification Approach	Remarks
[95]	Hybrid Self Learning Particle Swarm Optimization	Multi-period vehicle routing with time window	Minimizing total cost and total carbon emissions	Special value of parameters of SLPO leads to better Pareto front.	Comparison to NSGA-2, MOSLPSO gave a better trade-off result.
[96]	Modified NSGA-2	Tri-objective supply chain problem	Minimizing total cost, order variance and total inventory	Mutation operator has been implemented for entire population	Modified NSGA-2 shows better minimizing objective function value than original NSGA-2
[97]	Adopted NSGA-2	Multi-objective sustainable hub location-scheduling problem for perishable food industry	Minimizing transportation cost, maximizing the quality of product at delivery moment and minimize carbon emission of vehicles	Parameters of NSGA-2 were selected against the CAB dataset experiment, which affects the mutation and crossover probability of proposed algorithm	Adaptive NSGA-2 created more Pareto points than epsilon constraint method which gave a significant result.
[98]	Improved Multi-Objective Keshtel Algorithm	Bi-objective optimization model for citrus CLSC industry	Minimizing total cost and maximizing responsiveness of customers	Application of non-dominated sorting operator in initialization of every iteration	MOKA performed better in Number of Pareto solution and Mean ideal distance matrices compared to NSGA-2, NRGA
[99]	Decomposition-based MOEA	Crude Oil Supply chain design	Maximizing profit and minimizing impact of emission	Segregation of main problem into single objective several sub problems lead to less CPU time	With respect to number of Pareto fronts, CPU time and spacing points, MOEA-D seems better than MOPSO and NSGA-2
[95]	NSGA-2	Multi-period supply chain network design	Maximization of total profit, minimization of supply disruption and carbon emission	Taguchi method was implemented for parameter tuning, which gave minimal solution variance for that algorithm	Numerical experiment shows the robustness of NSGA-2 for multi-Objective problem instance
[100]	Greedy-based NSGA-2 with adaptive strategy	Capacitated Green Vehicle Routing	Minimization of total fuel consumption, maximizing customer satisfaction	Adaptive and greedy strategy have been adopted to modify NSGA-2	Number of Pareto solutions, diversification, quality and MID in these metric Adaptive NSGA-2 performed better than standard NSGA
[101]	Evolutionary algorithm having a set of experience knowledge structures (SOEKS) and DNA	Multiple Travelling Salesman Problem	Self-adaptive use and increase experience	Unique adaptive knowledge structure for decision modelling	Proposed algorithm has shown significant impact on solving travelling salesman problem
[102]	Modified NSGA-2	Two echelon location-routing problem considering demand uncertainty	Minimize total cost and maximizing service reliability	Application of novel validity measurement function and specially chosen initial population by customer clustering	With respect to cost value, number of routes and CPU time, M-NSGA-2 performed better than MOGA and MOPSO.

Table 5. *Cont.*

Authors	Proposed Algorithm	Application	Objective Function	Algorithm Modification Approach	Remarks
[103]	NSGA-2	Bi-objective vehicle routing problem with time windows	Minimizing transportation cost and time cost	Two objectives are optimized separately to avoid complexity	Against the Solomon benchmark metric proposed algorithm performed competitively
[104]	Integration of NSGA-2 and Monte Carlo Simulation	Industrial hazardous waste location-routing problem	Minimizing total cost, total site risk function along with transportation risk while delivering the waste	First approach to integrate NSGA-2 and Monte Carlo Simulation to tackle CO	Proposed simheuristic approach tends to produce a high-quality solution within a short CPU time

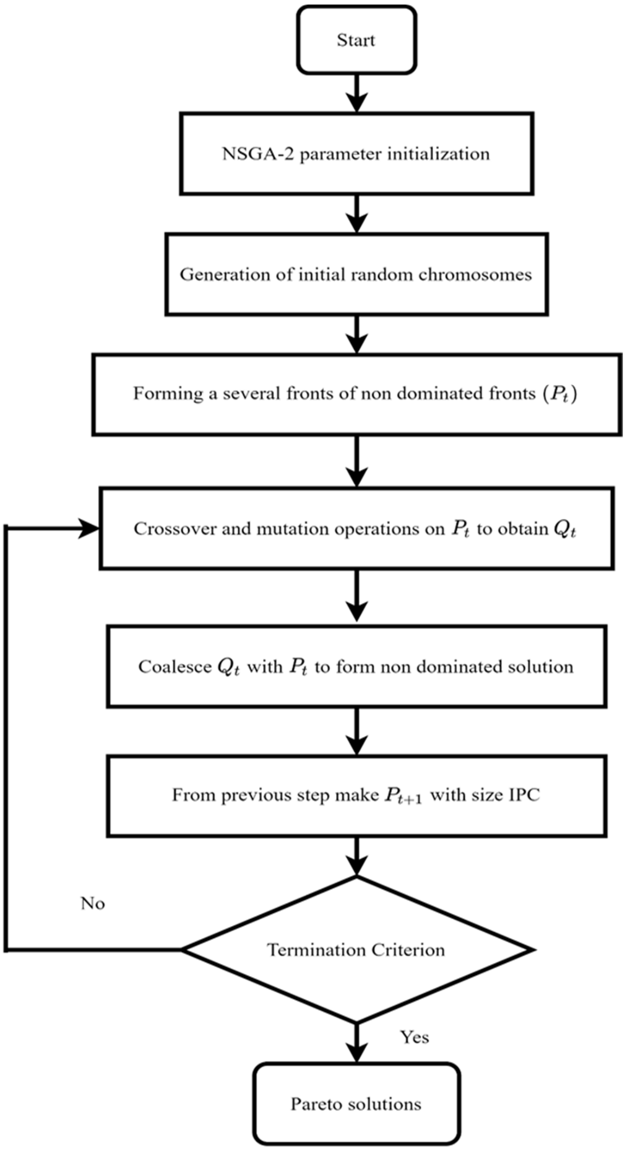
Start
NSGA-2 parameter initialization
Generation of initial random chromosomes
Forming a several fronts of non dominated fronts (P_t)
Crossover and mutation operations on P_t to obtain Q_t
Coalesce Q_t with P_t to form non dominated solution
From previous step make P_{t+1} with size IPC
No
Termination Criterion
Yes
Pareto solutions

Figure 13. Framework of the NSGA-2 algorithm.

5. Observation and Discussion

Through the above-mentioned discussion about the metaheuristic's implementation in CO problems, authors can conclude these following points to distinguish the adopted algorithms modification and constraint handling strategy

- Firefly algorithm has a strong tendency of sticking in local optima, so most of the researchers have attempted to improve the local search strategy through levy flight, randomization and gaussian distribution. Moreover, FA contains a single-direction randomization technique for determining the nearest solution. This phenomenon causes fewer exploration capabilities to reach global optima. Some of the researchers have recommended a Selective Firefly Algorithm (SFA) to improve the solution nature of standard FA. One of the advantages of FA is its self-adaptive nature for parameter tuning, which is why it has been easily manipulated to tackle several types of CO problems.
- The penalty function strategy has been the most frequently used strategy by researchers when faced with difficult constraints in CO problems. Due to the fact that CO problems typically have a more constrained search space, several researchers have implemented a penalty function strategy to deal with constrained search space criteria. Static penalty functions, co-evolutionary penalty functions, dynamic penalty functions, and adaptive penalty functions are all examples of penalty functions. Due to the easy implementation, they are the most widely used in CO problems. However, static penalty functions have a significant disadvantage in that they limit the exploration capabilities of metaheuristics algorithms by strictly eliminating initially infeasible solutions. As a result, some researchers have proposed using adaptive penalty functions to deal with this problem.
- Parameter tuning is critical for determining the precision and optimal computing time of any algorithm. The Taguchi and response surface methods appear to be the most frequently used techniques for parameter optimization in CO problems. In comparison to the response surface method, the Taguchi method has been shown to be more convenient because it requires fewer design experiments. Additionally, the response surface method is a local analysis, and as such, it is not valid for regions other than the one from which it originated.
- The preference for adopting metaheuristics mostly relies on the diversification and intensification balance factor of any certain algorithm. For example, CS uses Lévy flight for exploration, random walk for exploitation and lastly, follows the elitist mechanism like GA to form a better solution. It's been observed that for large instance problems, randomization has to be more impactful than levy flight for cuckoo search algorithm.
- Several researchers have recommended gamma distribution, Gaussian distribution and Cauchy distribution for the improvement of step size. The main drawback of CS is that there is no theoretical analysis exists which can illustrate the structural framework of CS while optimizing a difficult problem. In contrast, other reputed metaheuristics such as particle swarm optimization (PSO) and the genetic algorithm (GA) have a distinctive mathematical framework, which can describe the whole optimization process.
- Harmony search algorithm's main strengths include efficiency and ability to be modified and hybridized with other optimization algorithms; however, it has inherent weaknesses, including a tendency to hide in local optima since it emphasizes exploitation (local search and intensification) instead of exploration (global search). Harmony search's performance is strongly dependent on adjusting the parameters.
- The Bat Algorithm (BA) is naturally structured to optimize continuous problems; however, it needs some tweaking inside the algorithm framework to work with numerical or NP-hard problems like combinatorial optimization. The Bat Algorithm has a strong exploitation capability, which is why it tends to be stuck in local optima for large-scale problem instances. Researchers have designed several strategies to perform a balance between exploration and exploitation stage while searching occurs. BA has

been modified by inserting PSO's similar strategy for local search [64]. Moreover, implementing inertia weight by [62] for SCP and TSP consecutively. In VRP problems, self-adaptive strategy has been proven more effective than standard from BA and other variants. Another significant case was observed when self-adaptive BA was integrated with DE, which led to better convergence against benchmark function rather than standard BA.

- The artificial bee colony algorithm's initialization stage has been manipulated by mutation operator, levy flight technique, and chaotic operators. Moreover, these mentioned techniques have contributed to a better diversification solution processed by ABC as the problem classification.
- In the BFOA algorithm, the chemotaxis and reproduction stages are responsible for the local search process. When the problem space is large, it is recommended to adopt an elimination and dispersal process to avoid getting stuck in local optima. The exploration and exploration capability of the BFOA algorithm is controlled by the probability of elimination and dispersal parameters. At the beginning of the search process, it is recommended to assign a larger probability value, whereas, at the end of the search process, the probability of elimination and dispersal parameters should be reduced for better performance.
- NSGA-2 has been mostly implemented for optimizing the bi-objective CO problems. As per the nature of CO problems, the standard NSGA-2 has been modified in general by changing crossover and mutation operators. Several researchers recommended the Roulette wheel selection mechanism as a parent selection mechanism to get a significantly better convergence rate for NSGA-2.

6. Conclusions and Future Research Directions

This survey discusses the current trends of metaheuristics implementation in combinatorial optimization problems, such as travelling salesman problem, vehicle routing problem and supply chain design problems. Moreover, in this study, authors have conducted an extensive analysis of modification strategy, parameter tuning methods, along with the diversification processes of the seven most-cited metaheuristic algorithms, which are efficient to optimize complex combinatorial optimization problems. From the authors' viewpoint, it can be stated that most of the metaheuristics' success to handle complex combinatorial optimization problems mostly rely on encoding strategy, balanced diversification and exploration process as well as adaptive parameter tuning. The standard framework of these aforementioned metaheuristic algorithms was not compatible with combinatorial optimization problems. To deal with the complexity of combinatorial optimization problems, these metaheuristics have been modified through the hybridization of heuristic and metaheuristic approaches. During the hybridization process, most of the researchers have faced a challenge to address issues with local searches, including parameter tuning and robustness of the solution. After this, thorough analysis authors want to recommend some future research prospects by these following points:

- Performance evaluations of metaheuristics have been a critical issue, as always, for the researchers. Previous researchers have attempted to justify their proposed algorithm against some benchmarking functions, such as the Solomon benchmark function and TSPLIB (Travelling Salesman Problem Library). It has been observed that researchers had to find a statistical analysis test for the deficiency of proper benchmark functions while evaluating proposed algorithm performance. Thus, improvising the benchmark function can result in an interesting prospect for future endeavors.
- Combining newer metaheuristics with heuristic methods such as simheuristics and hyper-heuristics may provide a substantial result and a new scope for algorithm implementation.
- Decomposition-based techniques such as Lagrangian decomposition, Dantzig–Wolfie decomposition and Bender's decomposition have a successful transformation of solving MILP problems alike combinatorial optimization problems. So, this research's

future coverage can be extended to explore the recent advancements of decomposition-based metaheuristics for optimizing combinatorial problems.

- As new metaheuristics evolve for a global optimization perspective, it is suggested that numerous recently proposed algorithms such as the Spotted Hyena Optimizer [105], Gravitational Local Search, the Random Forest Algorithm [106], and the Opposition-Based Learning Method [107] could be adopted for successful optimization of combinatorial problems.

Metaheuristic algorithms are problem-specific, and the no-free-lunch theorem states that no algorithm is superior to another when it comes to solving all types of problems [108]. The researchers may consider repository maintenance, search process handling, problem identification system, evaluation component, and experiment runner of the solver algorithm when selecting metaheuristic algorithms for combinatorial optimization problems.

Author Contributions: Conceptualization, M.A.R.; methodology, M.A.R.; software, M.A.R.; validation, M.A.R. and K.B.; formal analysis, M.A.R. and K.B.; investigation, M.A.R. and R.S.; resources, M.A.R. and R.S.; data curation, M.A.R. and M.O.; writing—original draft preparation, M.A.R. and K.B.; writing—review and editing, R.S., L.A. and E.A.K.; visualization, M.A.R.; supervision, R.S.; project administration, M.A.R. and K.B.; funding acquisition, M.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Yayasan UTP (YUTP), grant number 015LCO-296.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the Center of Graduate Studies of Universiti Teknologi PETRONAS and the YUTP research fund (015LCO-296) for the funding of this research. Authors also recognize the contribution of the Department of Fundamental and Applied Sciences of the Universiti Teknologi PETRONAS for their invaluable support.

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

References

1. Schrijver, A. *Combinatorial Optimization: Polyhedra and Efficiency*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2003; Volume 24, ISBN 3540443894.
2. Jourdan, L.; Basseur, M.; Talbi, E.-G. Hybridizing exact methods and metaheuristics: A taxonomy. *Eur. J. Oper. Res.* **2009**, *199*, 620–629. [CrossRef]
3. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* **2003**, *35*, 268–308. [CrossRef]
4. Neumann, F.; Witt, C. Bioinspired computation in combinatorial optimization: Algorithms and their computational complexity. In Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, Amsterdam, The Netherlands, 6–10 July 2013; pp. 567–590.
5. Dokeroglu, T.; Sevinc, E.; Kucukyilmaz, T.; Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **2019**, *137*, 106040. [CrossRef]
6. Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78. [CrossRef]
7. Geem, Z.W.; Kim, J.H.; Loganathan, G. A New Heuristic Optimization Algorithm: Harmony Search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
8. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
9. Yang, X.-S.; Deb, S. Engineering optimisation by cuckoo search. *Int. J. Math. Model. Numer. Optim.* **2010**, *1*, 330. [CrossRef]
10. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report-TR06; Computer Engineering Department, Engineering Faculty, Erciyes University: Kayseri, Turkey, 2005; Volume 200, pp. 1–10.
11. Passino, K.M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control. Syst. Mag.* **2002**, *22*, 52–67. [CrossRef]
12. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]

13. Ramos-Figueroa, O.; Quiroz-Castellanos, M.; Mezura-Montes, E.; Schütze, O. Metaheuristics to solve grouping problems: A review and a case study. *Swarm Evol. Comput.* **2020**, *53*, 100643. [CrossRef]
14. Rachih, H.; Mhada, F.Z.; Chiheb, R. Meta-heuristics for reverse logistics: A literature review and perspectives. *Comput. Ind. Eng.* **2019**, *127*, 45–62. [CrossRef]
15. Euchi, J. The vehicle routing problem with private fleet and multiple common carriers: Solution with hybrid metaheuristic algorithm. *Veh. Commun.* **2017**, *9*, 97–108. [CrossRef]
16. Eskandarpour, M.; Dejax, P.; Miemczyk, J.; Péton, O. Sustainable supply chain network design: An optimization-oriented review. *Omega* **2015**, *54*, 11–32. [CrossRef]
17. Prodhon, C.; Prins, C. A survey of recent research on location-routing problems. *Eur. J. Oper. Res.* **2014**, *238*, 1–17. [CrossRef]
18. Korte, B.; Vygen, J. *Combinatorial Optimization: Theory and Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009; ISBN 4431100210.
19. Moghdani, R.; Salimifard, K.; Demir, E.; Benyettou, A. The green vehicle routing problem: A systematic literature review. *J. Clean. Prod.* **2021**, *279*, 123691. [CrossRef]
20. Chopra, S.; Meindl, P.; Kalra, D.V. *Supply Chain Management: Strategy, Planning, And Operation*; Pearson: Boston, MA, USA, 2013; Volume 232.
21. Yang, X.-S. Firefly algorithm, Levy flights and global optimization. In *Research and Development in Intelligent Systems XXVI*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 209–218.
22. Osaba, E.; Yang, X.-S.; Diaz, F.; Onieva, E.; Masegosa, A.D.; Perallos, A. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Soft Comput.* **2016**, *21*, 5295–5308. [CrossRef]
23. Trachanatzi, D.; Rigakis, M.; Marinaki, M.; Marinakis, Y. A firefly algorithm for the environmental prize-collecting vehicle routing problem. *Swarm Evol. Comput.* **2020**, *57*, 100712. [CrossRef]
24. Arora, J.S. Introduction to optimization. In *Optimization of Structural and Mechanical Systems*; Elsevier: Amsterdam, The Netherlands, 2007; pp. 1–34. ISBN 9789812779670.
25. Goel, R.; Maini, R. A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems. *J. Comput. Sci.* **2018**, *25*, 28–37. [CrossRef]
26. Alinaghian, M.; Naderipour, M. A novel comprehensive macroscopic model for time-dependent vehicle routing problem with multi-alternative graph to reduce fuel consumption: A case study. *Comput. Ind. Eng.* **2016**, *99*, 210–222. [CrossRef]
27. Altabeeb, A.M.; Mohsen, A.M.; Ghallab, A. An improved hybrid firefly algorithm for capacitated vehicle routing problem. *Appl. Soft Comput.* **2019**, *84*, 105728. [CrossRef]
28. Atabaki, M.S.; Khamseh, A.A.; Mohammadi, M. A priority-based firefly algorithm for network design of a closed-loop supply chain with price-sensitive demand. *Comput. Ind. Eng.* **2019**, *135*, 814–837. [CrossRef]
29. Rizk-Allah, R.; Zaki, E.M.; El-Sawy, A.A. Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems. *Appl. Math. Comput.* **2013**, *224*, 473–483. [CrossRef]
30. Udaiyakumar, K.; Chandrasekaran, M. Application of Firefly Algorithm in Job Shop Scheduling Problem for Minimization of Makespan. *Procedia Eng.* **2014**, *97*, 1798–1807. [CrossRef]
31. Jati, G.K. Evolutionary discrete firefly algorithm for travelling salesman problem. In Proceedings of the International Conference on Adaptive and Intelligent Systems, Klagenfurt, Austria, 6–8 September 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 393–403.
32. Eslami, M.; Shareef, H.; Khajehzadeh, M. Firefly algorithm and pattern search hybridized for global optimization. In Proceedings of the International Conference on Intelligent Computing, Nanning, China, 28–31 July 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 172–178.
33. Roeva, O. Genetic algorithm and firefly algorithm hybrid schemes for cultivation processes modelling. In *Transactions on Computational Collective Intelligence XVII*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 196–211.
34. Marichelvam, M.K.; Prabaharan, T.; Geetha, M. Firefly algorithm for flow shop optimization. In *Recent Advances in Swarm Intelligence and Evolutionary Computation*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 225–243.
35. Khalifehzadeh, S.; Fakhrzad, M.B. A Modified Firefly Algorithm for optimizing a multi stage supply chain network with stochastic demand and fuzzy production capacity. *Comput. Ind. Eng.* **2019**, *133*, 42–56. [CrossRef]
36. Tokhmehchi, N.; Makui, A.; Sadi-Nezhad, S. A Hybrid Approach to Solve a Model of Closed-Loop Supply Chain. *Math. Probl. Eng.* **2015**, *2015*, 1–18. [CrossRef]
37. Saghaeeian, A.; Ramezanian, R. An efficient hybrid genetic algorithm for multi-product competitive supply chain network design with price-dependent demand. *Appl. Soft Comput.* **2018**, *71*, 872–893. [CrossRef]
38. Shams, M.; El-Banbi, A.; Sayyouh, H. Harmony search optimization applied to reservoir engineering assisted history matching. *Pet. Explor. Dev.* **2020**, *47*, 154–160. [CrossRef]
39. Biswas, K.; Vasant, P.M.; Vintaned, J.A.G.; Watada, J. A Review of Metaheuristic Algorithms for Optimizing 3D Well-Path Designs. *Arch. Comput. Methods Eng.* **2021**, *28*, 1775–1793. [CrossRef]
40. Guo, Z.; Shi, L.; Chen, L.; Liang, Y. A harmony search-based memetic optimization model for integrated production and transportation scheduling in MTO manufacturing. *Omega* **2017**, *66*, 327–343. [CrossRef]
41. Guo, Z.X.; Yang, C.; Wang, W.; Yang, J. Harmony search-based multi-objective optimization model for multi-site order planning with multiple uncertainties and learning effects. *Comput. Ind. Eng.* **2015**, *83*, 74–90. [CrossRef]

42. Eskandari-Khanghahi, M.; Tavakkoli-Moghaddam, R.; Taleizadeh, A.A.; Amin, S.H. Designing and optimizing a sustainable supply chain network for a blood platelet bank under uncertainty. *Eng. Appl. Artif. Intell.* **2018**, *71*, 236–250. [CrossRef]
43. Yassen, E.T.; Ayob, M.; Nazri, M.Z.A.; Sabar, N.R. Meta-harmony search algorithm for the vehicle routing problem with time windows. *Inf. Sci.* **2015**, *325*, 140–158. [CrossRef]
44. Chen, S.; Chen, R.; Gao, J. A Modified Harmony Search Algorithm for Solving the Dynamic Vehicle Routing Problem with Time Windows. *Sci. Program.* **2017**, *2017*, 1–13. [CrossRef]
45. Ruano-Daza, E.; Cobos, C.; Torres-Jimenez, J.; Mendoza, M.; Paz, A. A multiobjective bilevel approach based on global-best harmony search for defining optimal routes and frequencies for bus rapid transit systems. *Appl. Soft Comput.* **2018**, *67*, 567–583. [CrossRef]
46. Gao, K.Z.; Suganthan, P.; Pan, Q.-K.; Tasgetiren, M.F. An effective discrete harmony search algorithm for flexible job shop scheduling problem with fuzzy processing time. *Int. J. Prod. Res.* **2015**, *53*, 5896–5911. [CrossRef]
47. Kianfar, K. Maximizing profit in a supply chain by considering advertising and price elasticity of demand. *Comput. Ind. Eng.* **2019**, *135*, 265–274. [CrossRef]
48. Hosseini, S.D.; Shirazi, M.A.; Ghomi, S.M.T.F. Harmony search optimization algorithm for a novel transportation problem in a consolidation network. *Eng. Optim.* **2013**, *46*, 1538–1552. [CrossRef]
49. Boryczka, U.; Szwarc, K. The Harmony Search algorithm with additional improvement of harmony memory for Asymmetric Traveling Salesman Problem. *Expert Syst. Appl.* **2019**, *122*, 43–53. [CrossRef]
50. Ramli, M.R.; Abas, Z.A.; Desa, M.I.; Abidin, Z.Z.; Alazzam, M.B. Enhanced convergence of Bat Algorithm based on dimensional and inertia weight factor. *J. King Saud Univ. Inf. Sci.* **2019**, *31*, 452–458. [CrossRef]
51. Wang, Y.; Wang, P.; Zhang, J.; Cui, Z.; Cai, X.; Zhang, W.; Chen, J. A Novel Bat Algorithm with Multiple Strategies Coupling for Numerical Optimization. *Mathematics* **2019**, *7*, 135. [CrossRef]
52. Hasançebi, O.; Teke, T.; Pekcan, O. A bat-inspired algorithm for structural optimization. *Comput. Struct.* **2013**, *128*, 77–90. [CrossRef]
53. Gandomi, A.H.; Yang, X.-S. Chaotic bat algorithm. *J. Comput. Sci.* **2014**, *5*, 224–232. [CrossRef]
54. Fister, I.; Fong, S.; Brest, J. A Novel Hybrid Self-Adaptive Bat Algorithm. *Sci. World J.* **2014**, *2014*, 1–12. [CrossRef]
55. Ali, E. Optimization of Power System Stabilizers using BAT search algorithm. *Int. J. Electr. Power Energy Syst.* **2014**, *61*, 683–690. [CrossRef]
56. Li, Y.; Guo, Q.; Liu, J. Improved bat algorithm for vehicle routing problem. *Int. J. Perform. Eng.* **2019**, *15*, 317–325. [CrossRef]
57. Lu, Y.; Jiang, T. Bi-Population Based Discrete Bat Algorithm for the Low-Carbon Job Shop Scheduling Problem. *IEEE Access* **2019**, *7*, 14513–14522. [CrossRef]
58. Luan, S.; Yang, Q.; Zhou, H.; Jiang, Z.; Wang, W.; Wang, Z.; Chu, R. The HSABA for Emergency Location-Routing Problem. *Math. Probl. Eng.* **2019**, *2019*, 1–12. [CrossRef]
59. Wang, C.; Zhou, S.; Gao, Y.; Liu, C. A self-adaptive bat algorithm for the truck and trailer routing problem. *Eng. Comput.* **2018**, *35*, 108–135. [CrossRef]
60. Taha, A.; Hachimi, M.; Moudden, A. A discrete Bat Algorithm for the vehicle routing problem with time windows. In Proceedings of the 2017 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA), Rabat, Morocco, 27–28 April 2017; IEEE: Picataway, NJ, USA, 2017; pp. 65–70.
61. Saji, Y.; Riffi, M.E. A novel discrete bat algorithm for solving the travelling salesman problem. *Neural Comput. Appl.* **2016**, *27*, 1853–1866. [CrossRef]
62. Osaba, E.; Yang, X.-S.; Diaz, F.; Lopez-Garcia, P.; Carballedo, R. An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems. *Eng. Appl. Artif. Intell.* **2016**, *48*, 59–71. [CrossRef]
63. Al-Sorori, W.; Mohsen, A.M. Multi-population Discrete Bat Algorithm with Crossover to Solve TSP. In Proceedings of the International Conference on Hybrid Intelligent Systems, Marrakech, Morocco, 21–23 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 466–478.
64. Sadeghi, J.; Mousavi, S.M.; Niaki, S.T.A.; Sadeghi, S. Optimizing a bi-objective inventory model of a three-echelon supply chain using a tuned hybrid bat algorithm. *Transp. Res. Part E Logist. Transp. Rev.* **2014**, *70*, 274–292. [CrossRef]
65. Civicioglu, P.; Besdok, E. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artif. Intell. Rev.* **2011**, *39*, 315–346. [CrossRef]
66. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009, Coimbatore, India, 9–11 December 2009; IEEE: Picataway, NJ, USA, 2009; pp. 210–214.
67. Yang, X.-S.; Deb, S. Cuckoo search: Recent advances and applications. *Neural Comput. Appl.* **2014**, *24*, 169–174. [CrossRef]
68. Tarhini, A.; Danach, K.; Harfouche, A. Swarm intelligence-based hyper-heuristic for the vehicle routing problem with prioritized customers. *Ann. Oper. Res.* **2020**, 1–22. [CrossRef]
69. Rezaei, S.; Kheirkhah, A. A comprehensive approach in designing a sustainable closed-loop supply chain network using cross-docking operations. *Comput. Math. Organ. Theory* **2018**, *24*, 51–98. [CrossRef]
70. Jamali, G.; Sana, S.S.; Moghdani, R. Hybrid improved cuckoo search algorithm and genetic algorithm for solving Markov-modulated demand. *RAIRO Oper. Res.* **2018**, *52*, 473–497. [CrossRef]
71. Ouaarab, A.; Ahiod, B.; Yang, X.S. Improved and discrete cuckoo search for solving the travelling salesman problem. *Stud. Comput. Intell.* **2014**, *516*, 63–84. [CrossRef]

72. Chen, X.; Wang, J. A novel hybrid cuckoo search algorithm for optimizing vehicle routing problem in logistics distribution system. *J. Comput. Theor. Nanosci.* **2016**, *13*, 114–119. [CrossRef]
73. Zheng, H.Q.; Zhou, Y.Q.; Luo, Q.F. A hybrid cuckoo search algorithm-GRASP for vehicle routing problem. *J. Converg. Inf. Technol.* **2013**, *8*. [CrossRef]
74. Xiao, L.; Dridi, M.; El Hassani, A.H.; Fei, H.; Lin, W. An Improved Cuckoo Search for a Patient Transportation Problem with Consideration of Reducing Transport Emissions. *Sustainability* **2018**, *10*, 793. [CrossRef]
75. Samanta, S.; Philip, D.; Chakraborty, S. Bi-objective dependent location quadratic assignment problem: Formulation and solution using a modified artificial bee colony algorithm. *Comput. Ind. Eng.* **2018**, *121*, 8–26. [CrossRef]
76. Kaveh, A.; Bakhshpoori, T. *Metaheuristics: Outlines, MATLAB Codes and Examples*; Springer: Cham, Switzerland, 2019; ISBN 3030040674.
77. Yao, B.Z.; Yang, C.Y.; Hu, J.J.; Yin, G.D.; Yu, B. An improved artificial bee colony algorithm for job shop problem. In *Applied Mechanics and Materials*; Trans Tech Publications: Bäch, Switzerland, 2010; Volume 26, pp. 657–660.
78. Li, X.; Peng, Z.; Du, B.; Guo, J.; Xu, W.; Zhuang, K. Hybrid artificial bee colony algorithm with a rescheduling strategy for solving flexible job shop scheduling problems. *Comput. Ind. Eng.* **2017**, *113*, 10–26. [CrossRef]
79. Meng, T.; Pan, Q.-K.; Sang, H.-Y. A hybrid artificial bee colony algorithm for a flexible job shop scheduling problem with overlapping in operations. *Int. J. Prod. Res.* **2018**, *56*, 5278–5292. [CrossRef]
80. Khan, I.; Maiti, M.K. A swap sequence based Artificial Bee Colony algorithm for Traveling Salesman Problem. *Swarm Evol. Comput.* **2019**, *44*, 428–438. [CrossRef]
81. Pandiri, V.; Singh, A. A swarm intelligence approach for the colored traveling salesman problem. *Appl. Intell.* **2018**, *48*, 4412–4428. [CrossRef]
82. Zhong, Y.; Lin, J.; Wang, L.; Zhang, H. Hybrid discrete artificial bee colony algorithm with threshold acceptance criterion for traveling salesman problem. *Inf. Sci.* **2017**, *421*, 70–84. [CrossRef]
83. Baradaran, V.; Shafaei, A.; Hosseinian, A.H. Stochastic vehicle routing problem with heterogeneous vehicles and multiple prioritized time windows: Mathematical modeling and solution approach. *Comput. Ind. Eng.* **2019**, *131*, 187–199. [CrossRef]
84. Cui, Y.Y.; Guan, Z.; Saif, U.; Zhang, L.; Zhang, F.; Mirza, J. Close loop supply chain network problem with uncertainty in demand and returned products: Genetic artificial bee colony algorithm approach. *J. Clean. Prod.* **2017**, *162*, 717–742. [CrossRef]
85. Szeto, W.; Wu, Y.; Ho, S.C. An artificial bee colony algorithm for the capacitated vehicle routing problem. *Eur. J. Oper. Res.* **2012**, *215*, 126–135. [CrossRef]
86. Datta, A.; Nandakumar, S. A survey on bio inspired meta heuristic based clustering protocols for wireless sensor networks. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Busan, Korea, 25–27 August 2017; Volume 263, p. 52026.
87. Hota, P.; Barisal, A.; Chakrabarti, R. Economic emission load dispatch through fuzzy based bacterial foraging algorithm. *Int. J. Electr. Power Energy Syst.* **2010**, *32*, 794–803. [CrossRef]
88. Niu, B.; Fan, Y.; Xiao, H.; Xue, B. Bacterial foraging based approaches to portfolio optimization with liquidity risk. *Neurocomputing* **2012**, *98*, 90–100. [CrossRef]
89. Niu, B.; Wang, H.; Tan, L.; Xu, J. Multi-objective optimization using BFO algorithm. In Proceedings of the International Conference on Intelligent Computing, Chongqing, China, 8–9 January 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 582–587.
90. Liu, C.; Wang, J.; Leung, J.Y.-T. Integrated bacteria foraging algorithm for cellular manufacturing in supply chain considering facility transfer and production planning. *Appl. Soft Comput.* **2018**, *62*, 602–618. [CrossRef]
91. Sinha, A.K.; Anand, A. Optimizing supply chain network for perishable products using improved bacteria foraging algorithm. *Appl. Soft Comput.* **2020**, *86*, 105921. [CrossRef]
92. Pasandideh, S.H.R.; Niaki, S.T.A.; Asadi, K. Bi-objective optimization of a multi-product multi-period three-echelon supply chain problem under uncertain environments: NSGA-II and NRGA. *Inf. Sci.* **2015**, *292*, 57–74. [CrossRef]
93. Sarrafha, K.; Rahmati, S.H.A.; Niaki, S.T.A.; Zaretalab, A. A bi-objective integrated procurement, production, and distribution problem of a multi-echelon supply chain network design: A new tuned MOEA. *Comput. Oper. Res.* **2015**, *54*, 35–51. [CrossRef]
94. Nekooghadirli, N.; Tavakkoli-Moghaddam, R.; Ghezavati, V.R.; Javanmard, S. Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics. *Comput. Ind. Eng.* **2014**, *76*, 204–221. [CrossRef]
95. Kumar, R.S.; Choudhary, A.; Babu, S.A.K.I.; Kumar, S.K.; Goswami, A.; Tiwari, M.K. Designing multi-period supply chain network considering risk and emission: A multi-objective approach. *Ann. Oper. Res.* **2017**, *250*, 427–461. [CrossRef]
96. Bandyopadhyay, S.; Bhattacharya, R. Solving a tri-objective supply chain problem with modified NSGA-II algorithm. *J. Manuf. Syst.* **2014**, *33*, 41–50. [CrossRef]
97. Musavi, M.; Bozorgi-Amiri, A. A multi-objective sustainable hub location-scheduling problem for perishable food supply chain. *Comput. Ind. Eng.* **2017**, *113*, 766–778. [CrossRef]
98. Cheraghalipour, A.; Paydar, M.M.; Hajiaghaei-Keshteli, M. A bi-objective optimization for citrus closed-loop supply chain using Pareto-based algorithms. *Appl. Soft Comput.* **2018**, *69*, 33–59. [CrossRef]
99. Azadeh, A.; Shafiee, F.; Yazdanparast, R.; Heydari, J.; Fathabad, A.M. Evolutionary multi-objective optimization of environmental indicators of integrated crude oil supply chain under uncertainty. *J. Clean. Prod.* **2017**, *152*, 295–311. [CrossRef]
100. Xu, Z.; Elomri, A.; Pokharel, S.; Mutlu, F. A model for capacitated green vehicle routing problem with the time-varying vehicle speed and soft time windows. *Comput. Ind. Eng.* **2019**, *137*, 106011. [CrossRef]

101. Wang, P.; Sanin, C.; Szczerbicki, E. Evolutionary algorithm and decisional DNA for multiple travelling salesman problem. *Neurocomputing* **2015**, *150*, 50–57. [CrossRef]
102. Wang, Y.; Assogba, K.; Liu, Y.; Ma, X.; Xu, M.; Wang, Y. Two-echelon location-routing optimization with time windows based on customer clustering. *Expert Syst. Appl.* **2018**, *104*, 244–260. [CrossRef]
103. Zhao, P.; Luo, W.; Han, X. Time-dependent and bi-objective vehicle routing problem with time windows. *Adv. Prod. Eng. Manag.* **2019**, *14*, 201–212. [CrossRef]
104. Rabbani, M.; Heidari, R.; Yazdanparast, R. A stochastic multi-period industrial hazardous waste location-routing problem: Integrating NSGA-II and Monte Carlo simulation. *Eur. J. Oper. Res.* **2019**, *272*, 945–961. [CrossRef]
105. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [CrossRef]
106. Wang, H.; Jin, Y. A Random Forest-Assisted Evolutionary Algorithm for Data-Driven Constrained Multiobjective Combinatorial Optimization of Trauma Systems. *IEEE Trans. Cybern.* **2020**, *50*, 536–549. [CrossRef] [PubMed]
107. Ghafori, S.; Gharehchopogh, F.S. Advances in Spotted Hyena Optimizer: A Comprehensive Survey. *Arch. Comput. Methods Eng.* **2021**. [CrossRef]
108. Ahmed, R.; Mahadzir, S.; Rozali, N.E.M.; Biswas, K.; Matovu, F.; Ahmed, K. Artificial intelligence techniques in refrigeration system modelling and optimization: A multi-disciplinary review. *Sustain. Energy Technol. Assess.* **2021**, *47*, 101488. [CrossRef]