

## Review Article

# A Comprehensive Review on Scatter Search: Techniques, Applications, and Challenges

Minakshi Kalra,<sup>1</sup> Shobhit Tyagi,<sup>2</sup> Vijay Kumar ,<sup>2</sup> Manjit Kaur ,<sup>3</sup>  
Wali Khan Mashwani ,<sup>4</sup> Habib Shah,<sup>5</sup> and Kamal Shah <sup>6</sup>

<sup>1</sup>Computer Science Department, Government College, Bahadurgarh, Jhajjar, Haryana, India

<sup>2</sup>Computer Science and Engineering Department, National Institute of Technology Hamirpur, Hamirpur, Himachal Pradesh, India

<sup>3</sup>Computer Science Engineering, Bennett University, Greater Noida, India

<sup>4</sup>Institute of Numerical Sciences, Kohat University of Science & Technology (KUST), Kohat, Pakistan

<sup>5</sup>College of Computer Science, Department of Computer Science, King Khalid University, Abha, Saudi Arabia

<sup>6</sup>Department of Mathematics, University of Malakand, Chakdara Dir(L), KPK, Pakistan

Correspondence should be addressed to Wali Khan Mashwani; mashwanigr8@gmail.com

Received 28 February 2021; Accepted 22 May 2021; Published 31 May 2021

Academic Editor: Erivelton Geraldo Nepomuceno

Copyright © 2021 Minakshi Kalra et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recent years have witnessed the use of metaheuristic algorithms to solve the optimization problems that usually require extensive computations and time. Among others, scatter search is the widely used evolutionary metaheuristic algorithm. It uses the information of global optima, which is stored in a different and unique set of solutions. In this paper, an updated review of scatter search (SS) is given. SS has been successfully applied in a variety of research problems, namely, data mining, bioinformatics, and engineering design. This paper presents the modified and hybrid versions of SS with their applications. The control strategies are discussed to show their impact on the performance of SS. various issues and future directions related to SS are also discussed. It inspires innovative researchers to use this algorithm for their research domains.

## 1. Introduction

The real-life problems require a large number of variables and constraints, which makes the computation process complex. Some problems also include nonlinear constraints, large solution space, high computational cost, multiple local optimum solutions, and complexity [1]. Recently, it has been observed from the literature that metaheuristic algorithms are being used to solve the real-life problems such as medical imaging, remote sensing, engineering design problems, and traffic control management [2–4]. Unlike traditional algorithms whose main focus includes precision, accuracy, and completeness, metaheuristic algorithms (MHAs) focus on speed and efficiency. For a given problem, there is no guarantee that MHA produces the global optimal solution. However, they are able to produce an approximate solution using their intuitive and plausible arguments.

MHAs are broadly classified into four main categories [5]. These are evolutionary algorithms (or EAs), trajectory-based algorithms (or TBAs), swarm-based algorithms (or SBAs), and physics-based metaheuristic algorithms (or PBAs) (see Figure 1). EAs are the algorithms, which are based on the evolution of species. These are based on the theory of Charles Darwin, “Survival of Fittest.” The fitness of a particular candidate is measured by its adaptation capability towards the environment [6]. Basically, EAs focus only on a subset of mechanisms defined over the biological evolutionary process. Some of the popular EAs are genetic algorithm (GA) [7], differential evolution (DE) [8], etc. Both GA and DE use the evolutionary process to find the best candidate solution with highest fitness value. Other EAs include biogeography-based optimizer (BBO) [9], genetic programming (GP) [10], and evolution strategy (ES) [11].

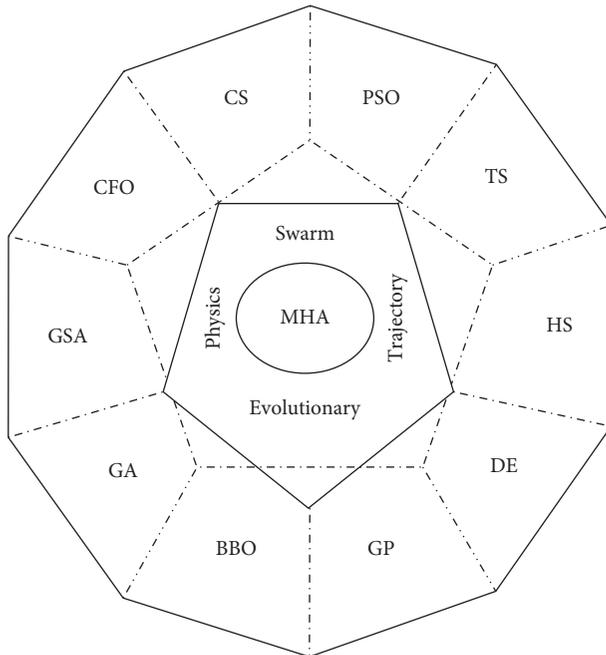


FIGURE 1: Classification of metaheuristic algorithms.

TBAs are inspired by biology to evolve a single trajectory of search points rather than the population. These are also known as human-based metaheuristic algorithms. The well-known single trajectory algorithms are tabu search (TS) [12], harmony search (HSA) [13], simulated annealing (SA) [14], and teaching-learning-based optimization (TLBO) [15]. PBAs are derived from the existing physical laws. Some of these algorithms are Central Force Optimization (CFO) [16], Big-Bang Big-Crunch (BBBC) [17], and Gravitational Search Algorithm (GSA) [18]. The search agents used by these algorithms can communicate with each other using the laws of physics such as laws of thermodynamics and gravitational force. SBAs are also known as swarm intelligence algorithms. These algorithms are derived by the “collective intelligence.” SBAs include flocks of birds, schools of fish, and colonies of ants. The well-known algorithms are particle swarm optimization (PSO) and ant colony optimization (ACO). PSO investigates probabilistic algorithms inspired by the flocking, schooling, and herding [19]. ACO are based on the behaviour of several species of ants [20]. Apart from PSO and ACO, other SBAs include artificial bee colony (ABC) algorithm [21], cuckoo search (CS) [22], bat-inspired algorithm (BA) [23], spiral spotted hyena optimizer (SSHO) [24], etc.

Scatter search (SS) is a metaheuristic algorithm whose working is based on its sibling tabu search [12]. SS is useful for continuous as well as discrete optimization problems. SS shows the promising results for solving many nonlinear and combinatorial problems. SS can be used in many nonlinear optimization problems on bounded variables such as linear ordering problem, max clique problem, graph coloring problem, job shop scheduling, bipartite drawing problem, and capacitated multicommodity network design. Figure 2 shows the

number of research articles published on SS according to Google Scholar (2013–2021).

The main attention of this paper is to present a comprehensive review of SS and encourage the young researchers to utilize SS in their research area. This paper discusses the control parameters of SS. The modification and hybridization of SS with other optimization techniques are discussed with pros and cons. The recent applications of SS are discussed with their methodologies and features. The possible future research directions for young researchers are also investigated.

This paper summarizes the scatter search as follows:

- (1) Workflow of SS that comprises the various components of SS
- (2) Improvements in the performance of SS that comprise modification and hybridization with other optimization techniques
- (3) Improvements in control parameters of SS that involve path relinking, subset reduction, set covering, and memory
- (4) Applicability of SS that comprises scheduling, data mining, civil engineering, bioinformatics, and manufacturing
- (5) Possible research direction of SS, which consists of modification and applicability of SS in different research domains

The rest of this paper is structured as follows: Section 2 presents the research methodology used in this review. Section 3 presents the classical scatter search along with its working procedure. In Section 4, the techniques used to improve scatter search are discussed. Section 5 presents the variants of scatter search. The applications of scatter search are mentioned in Section 6. Section 7 discusses future research directions followed by concluding remarks in Section 8.

## 2. Research Methodology

The guidelines of PRISMA [25] are used to conduct the systematic review of scatter search. The search has been conducted on four different databases such as Google Scholar, Scopus, PubMed, and arXiv during 20–25 April, 2021. Besides this, manual search has also been conducted to include the research articles. During search process, the query string comprises of “Scatter search” or “SS” or “((review) AND (scatter search))” or “tabu search” or “Applications of scatter search” or “((application) AND (SS))” or “((scatter search) AND (software engineering))” or “modified scatter search” or “improved SS”. The research articles related to advancement and hybridization in scatter search were also considered.

After the identification, all the duplicate and irrelevant research articles were excluded. 246 research articles were considered for next phase of screening. After examining the titles and abstract of screened articles, 182 articles were eliminated. 64 articles were selected for eligibility phase. After reading the full articles, 34 articles were removed

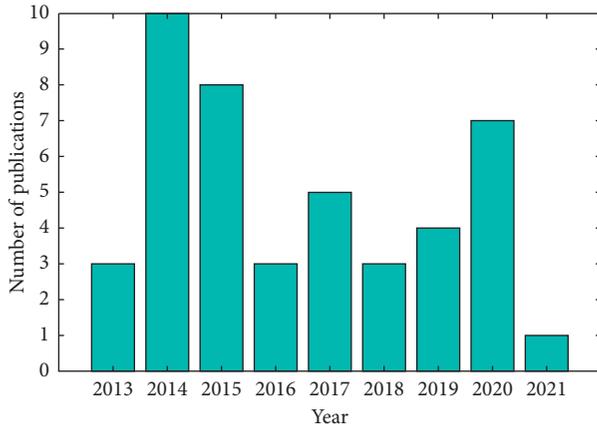


FIGURE 2: Number of publications on SS.

according to excluding criteria mentioned in Table 1. Finally, 30 articles were considered for this review. Figure 3 shows the search process with including and excluding criteria.

### 3. Classical Scatter Search

In this section, the inspiration behind the scatter search (SS) is discussed. Thereafter, the working of SS is presented.

**3.1. Inspiration.** The concept of SS is derived from tabu search (TS) [26]. Similar to TS, SS employed local searches for optimization. In local search, TS finds the similar solutions in their neighborhood that have minor differences with the hope of finding an improved solution. The problem with local searches is that they may stick up in the suboptimal or local regions. In TS, different moves are used to enhance their performance. The first move (also known as Worsening Move) is accepted if no improved move is found (i.e., in case of local search). The second move (known as prohibition) is used to back track the search for the earlier visited solutions. TS is implemented using memory structures that consist of user provided and visits solutions. If any potential solution violates a rule or visited again in short span of time, then this solution is marked as “tabu” (forbidden) and the search does not consider this solution again [26].

**3.2. SS Algorithm.** Scatter search (SS) algorithm was proposed by Glover in 1977 [27]. SS uses an iterative method in which the initial population of different and high-quality candidate solutions is partitioned into a set of subsets [28]. A set of subset solutions is known as reference set (RS), and they are used to create new ones. These subsets are linearly recombined using a predefined heuristic in the context of reference set (RS) whether the process is repeated or not. SS joins the subset solutions using Euclidean space to construct generalized paths. Unlike GA, SS uses relatively small population in the reference set (RS).

Glover [29] presented a template for implementation of SS. According to this template, SS consists of five procedures

to reach the final solution set. These are diversification generation, solution improvement, reference set update, subset generation, and solution combination methods. The relationship between input and output is illustrated in Table 2. The brief description of these methods is given in the succeeding subsections. The framework of SS is depicted in Figure 4.

**3.2.1. Diversification Generation Method.** Diversification Generation Method (DGM) is used to create distinct and different trial solutions [30]. These trial solutions become the basis for initializing the search process. These solutions are also known as arbitrary (or seed) solutions and are used as an input to the algorithm. SS generates high-grade results when DGM produces solutions that include a good mix of both quality and diversification. Examples of DGM include randomization, semigreedy, and greedy randomized adaptive search procedure (GRASP) [31]. Table 3 illustrates the different types of diversification generators that can be used to produce initial population.

**3.2.2. Improvement Method.** The improvement method (IM) is used to generate one or more unique trial solutions. IM converts the seed solution into an “enhanced” solution using some mechanism [31]. If the initial trial solution is the same as the improved solution, then the enhanced solution is to be considered as the input seed solution. The aim of IM is to improve the quality and feasibility of the input solution. The output of IM may be better than the original solution in terms of quality and feasibility. IM is able to handle both feasible and nonfeasible solutions. Some of IM are local search and variable neighborhood descent (VND). IM has deterministic orientation and termination criterion. In general, IM is designed for specific problems to provide specific results [32].

**3.2.3. Reference Set Update Method.** DGM produces different solution set of large size (say,  $S$ ). Typically, the size of  $S$  produced by DGM is much larger than reference set ( $Ref_{set}$ ). The initial  $Ref_{set}$  is built according to the reference set update method (RSUM). RSUM is used to update and build the population, which comes out from DGM and IM methods. RSUM consists of two steps [33].  $Ref_{set}$  is a good mixture of highly different solutions, which are drawn from the initial  $S$  population. This step is carried out by different methods in different implementations of SS. From population  $S$ , the best set of solutions ( $S/2$ ) is selected using the values obtained from the objective function. This set is added to  $Ref_{set}$  and deleted from  $S$ . The remaining half is selected using a sophisticated measure called distance  $D(r, p)$ . Here,  $r$  is the reference solution and  $p$  is the solution in  $S$ . The distance is measured by a method of designer’s choice but must follow certain usual metric conditions such as [34]

$$\begin{aligned} D(r, p) &= 0 \Leftrightarrow r = p, \\ D(r, q) + D(q, p) &\geq D(r, p). \end{aligned} \quad (1)$$

Euclidean distance is a commonly used method for continuous variables and is represented as [35]

TABLE 1: Selection criteria used for shortlisting research articles.

S. No.	Parameter	Include criteria	Exclude criteria
1	Analysis	Research articles involve the various control parameters of SS	Research articles involve the different control parameters of other metaheuristics
2	Comparison	Research articles focus on the variants and control operators of SS	Research articles focus on the variants of other metaheuristics
3	Application	Research articles on software engineering, civil, mechanical, manufacturing, bioinformatics, and networking	Research articles on social-economics, environment, and society
4	Study	Research involves mathematical foundation and experimental results	Research involves patent, articles in different language other than English

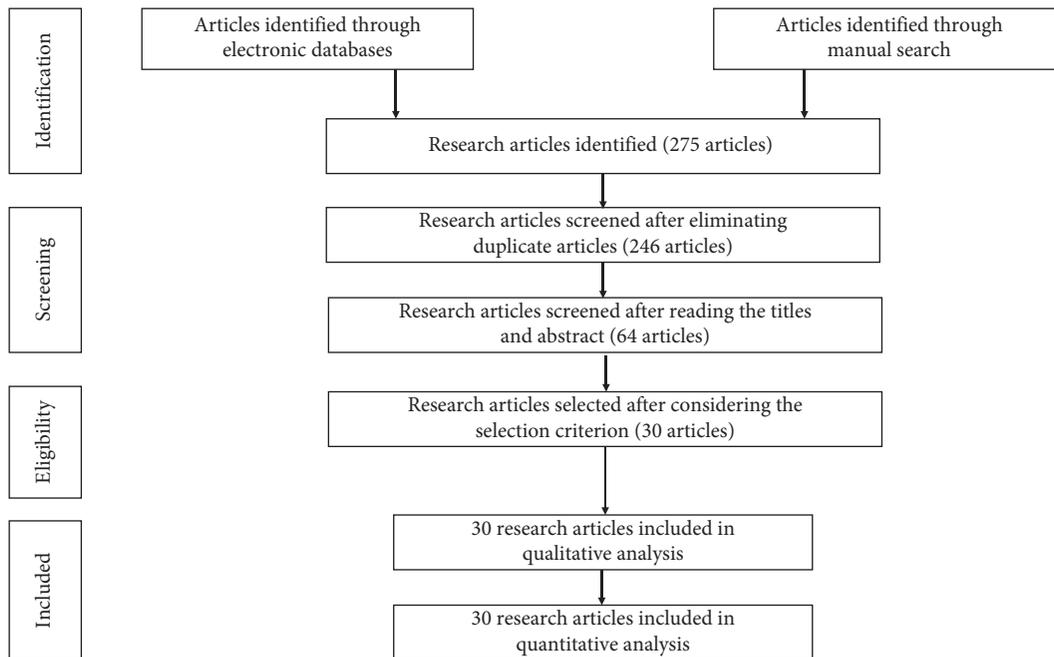


FIGURE 3: Research methodology used for shortlisting articles.

TABLE 2: Relationship between input and output of different procedures of SS.

Procedure	Input	Output	Link to next procedure
Diversification Generation	Random seed solutions	Diverse trail solutions	Improvement
Improvement	Diverse trail solutions	Enhanced trail solutions	Reference set update
Reference set update	Enhanced trail solutions	Set of best solutions	Subset generation
Subset generation	Set of best solutions	Subset of solution for Reference set	Solution combination
Solution combination	Subset of solution for Reference set	Combined solutions	Improvement

1. Diversification Generation and Improvement Methods
2. while (termination criteria is not satisfied)
3. {
4. Reference Set Update Method
5. while (new reference solutions)
6. {
7. Subset Generation Method
8. Subset Combination Method
9. Improvement Method
10. Reference Set Update Method
11. }
12. Rebuild Reference Set
13. }

FIGURE 4: Framework of SS.

TABLE 3: Different types of diversification generators.

Generator type	Description	Pros	Cons
Random	Randomly selects a subset of variables to set the value of 1 for every solution produced	Ease to implement, few parameters are required	Diversity in population is not mentioned
Sequential	The solutions are produced that utilize seed solution, number of problem variables, and an approximation parameter	Fast	Generator does not ensure the quality of produced solutions
Dichotomous	The variable set is partitioned into two subsets and complemented with seed solution	Fast, better diversity than sequential generator	A small number of solutions are produced with no control over the quality
Relaxation	Produces an initial population based on the objective value	Both solution quality, diverse solutions	Higher temporal complexity

$$D(r, p) = \sqrt{\sum_{i=1}^n (r_i - p_i)^2}. \quad (2)$$

Hamming distance (or HD) is equal to the different number of bits in the corresponding string symbols [36]. Similarly, HD is the number of changes needed to convert one string to another bit by bit. The following mathematical formulation is used to compute HD:

$$D(r, p) = \sum_{i=1}^n v_i, \quad (3)$$

where

$$v_i = \begin{cases} 0, & \text{if } r_i = p_i, \\ 1, & \text{if } r_i \neq p_i. \end{cases} \quad (4)$$

The distance is used to measure the minimum distance between population solution ( $p$ ) and the reference solutions ( $r$ ) [32]:

$$\begin{aligned} D_{\min}(p) &= \min_{r \in \text{Ref}_{\text{set}}} \{D(r, p)\}, \\ p^* &= p: \max_{p \in S} \{D_{\min}(p)\}. \end{aligned} \quad (5)$$

This overall phase is repeated to produce the initial  $\text{Ref}_{\text{set}}$ .  $D_{\min}$  is updated as per the requirement. Now,  $p^*$  is the new population solution that is going to be added in  $\text{Ref}_{\text{set}}$ . The new minimum distance can be rewritten as follows:

$$D_{\text{new\_min}}(p) = \min(D_{\min}(p), D(p, p^*)). \quad (6)$$

In general, RSUM is executed for 500 iterations and the most different  $\text{Ref}_{\text{set}}$  is chosen to initiate SS. At this point of time, the method is applied on trial solutions, that are produced by the subset generation, combination, and IM (see the steps 7–9) in framework of SS. The Euclidean distance computation can also be done by using maximum diversity problem (MDP). The diversity is calculated by the sum of distance between all pairs of elements present in the given set. MDP is nonlinear NP-hard problem, and SS procedure can be used to produce an optimal solution for such problems [32, 36].

**3.2.4. Subset Generation Method.** The subset generation (SG) method provides input to CM. Initially, all the reference solutions are new. This set of solutions is the initial  $\text{Ref}_{\text{set}}$  on which SG method is used. SG method is a list of all pairs of reference solutions. Since the SG method works on the universe of all pairs of reference solutions, therefore, there must be a moderate size of  $\text{Ref}_{\text{set}}$  (typically 20). Each time the SG method is executed, the number of reference solutions depends upon the strategy implemented in the RSUM. In general, the number of 2 subsets produced by the SG method is calculated as follows [32, 37]:

$$\frac{nm + (n^2 - m^2)}{2}, \quad (7)$$

where  $n$  and  $m$  are the number of new and old solutions in the reference set, respectively. The generation of subsets can be carried out by many methods of different sizes by creating subset types. Table 4 shows the subset types in the SG method. Martí [38] studied the results of the SG method with different types of subsets. The studies also indicate that the subset Type-1 is an efficient and effective one. Most of SS implementations avoid the use of higher-dimension subsets.

**3.2.5. Combination Method.** The combination method (CM) produces new trial solutions by combining the existing elements of different subsets. This method is directly related to the representation of solutions. In CM, each subset can create one or more new solutions that depend upon the specific form of solution. All solutions produced by CM are subjected to IM. All solutions produced by CM are subjected to IM and RSUM as outlined in Figure 5 dedicated for the procedure of SS.

## 4. Control Strategies Used in SS

In the recent past, researchers have developed many methods/strategies for modification in components of SS to enhance the performance and efficiency. Modern designs of SS generally increase the complexity along with new additional search parameters. There is no algorithm that can be used systematically to improve the performance of SS implementation [32–34]. Some of advanced designs did not manifest the importance or ranking of these

TABLE 4: Subset types in the SG method.

Subset type	Description
Type-I	SG contains two-element subsets
Type-II	Three-element subsets constituted from two-element subsets
Type-III	Four-element subsets constituted from three-element subsets
Type-IV	Subsets constituted from $k$ -best elements (i.e., $k$ lies in the range of $[5, Ref_{set}]$ )

### ***Scatter Search Algorithm***

#### ***Begin***

Set  $P = \phi$

**while** ( $|P| \leq Pop\_Size$ )

*Apply diversification generation on random seed solution to construct a solution  $y$*

*Apply improvement method on  $y$  to produce  $y^*$*

**if**  $y^* \notin P$  **then**

*add  $y^*$  to  $P$*

**else**

*discard  $y^*$*

**endif**

*Arrange solutions in  $P$  according to fitness function*

**while** ( $t < MaxGeneration$ )

Generate reference set  $Ref_{Set} = Ref_{Set1} \cup Ref_{Set2}$  /\*  $|Ref_{Set1}| = B_1$  and  $|Ref_{Set2}| = B_2$  \*/

Set  $Flag = TRUE$

**while** ( $Flag$ )

Compute  $Max_{Subset}$  that have at least one element

**for each**  $Max_{Subset}$

*Generate new subset ( $s$ ) using Subset Generation Method*

*Apply Solution Combination Method on newly generated subsets ( $y_s$ )*

*Apply Improvement Method on combined subset ( $y_s$ ) to obtain improve solution  $y_s^*$*

**if**  $y_s^* \notin Ref_{Set}$  and  $fit(y_s^*) > fit_{worst}(Ref_{Set1})$  **then**

*Add  $y_s^*$  to  $Ref_{Set1}$  and Set  $Flag = TRUE$*

**else**

**if**  $y_s^* \notin Ref_{Set2}$  and  $D_{min}(y_s^*) > D_{min}(y)$  **then**

*Add  $y_s^*$  to  $Ref_{Set1}$  and Set  $Flag = TRUE$*

**end if**

**end if**

**end for**

**end while**

**end while**

*A new set  $P$  using diversification generation method*

**end**

FIGURE 5: Procedure of SS.

strategies [38]. Figure 6 depicts the strategies used to improve the design of SS.

**4.1. Dynamic Update Strategy.**  $\text{Ref}_{\text{set}}$  is the most important part in SS. During the search process, the SS method is incapable of improving the best solution found by CM for the same reference solutions. The output of CM is based on the diversity of  $\text{Ref}_{\text{set}}$ . Therefore, an advance CM gives an advantage even if the  $\text{Ref}_{\text{set}}$  is not built efficiently [39]. In the SS method, the static update strategy is used in which  $\text{Ref}_{\text{set}}$  is not updated until all pairs are not subjected to CM. The new  $\text{Ref}_{\text{set}}$  is built by selecting the best solutions from the solution pool, and the solutions are currently present in  $\text{Ref}_{\text{set}}$ .

Dynamic updating strategy is a better approach to update the solutions in  $\text{Ref}_{\text{set}}$ . In this method, instead of waiting for all the solutions to be combined by the CM, the set is admitted immediately before the next combination is performed. If the new solution is admitted to  $\text{Ref}_{\text{set}}$ , then CM has to be applied on the newly inserted solution. Therefore, the solutions are either discarded or become the part of  $\text{Ref}_{\text{set}}$  as soon as they are produced. The main advantage of this approach is that the solutions of inferior quality will be discarded quickly and replaced with the newly produced solutions, which can be further used to create improved solutions. The promising solutions may be removed in special circumstances. Also, this method is more complex than the static updating of  $\text{Ref}_{\text{set}}$ . The dynamic updating approach can be used to fine-tune the procedure of SS.

**4.2. Candidate Generation Strategy.** Two well-known methods, namely, clustering and anticlustering are used to improve the candidate solution set. Glover [39] proposed a way to improve SS. This approach divides  $\text{Ref}_{\text{set}}$  into different clusters for selecting the parent solutions. This method operates with different subpopulations to produce new population [40]. These points may be present on different clusters that lead to diversification. The parents can be chosen alternatively. In “anticlustering,” the parents are selected to be as far as possible from different clusters within the same family of parents. Clustering gives the strategic advantage to SS.

**4.3. Diversity Control Strategy.** SS does not allow duplicates in  $\text{Ref}_{\text{set}}$ . The hash function is used to reduce the computation and duplicate solutions in  $\text{Ref}_{\text{set}}$ . The following hash function can be used to avoid duplicates and comparing solutions. Here, the solutions are represented with permutation  $x$  of size  $n$ :

$$H(x) = \sum_{i=1}^n i \times x(i)^2. \quad (8)$$

The implementations of SS are designed to check the duplicates in  $\text{Ref}_{\text{set}}$  [40]. They generally do not check the diversity of the solution when creating the initial  $\text{Ref}_{\text{set}}$ . In  $\text{Ref}_{\text{set}}$  rebuilding, the diversity of set  $B2$  is strictly checked by some distance measure. Initially, the diversity of set  $B1$  is not

monitored. Therefore, a minimum diversity test can be used to check whether  $B1$  has diversified solutions or not. The solutions should be chosen as members of the initial  $\text{Ref}_{\text{set}}$ .

**4.4. Rebuilding Strategy.** This approach is used when no new trial solutions are admitted to  $\text{Ref}_{\text{set}}$ . In this method, more diversified sets are created using the existing reference solutions.  $\text{Ref}_{\text{set}}$  is divided into two sets  $B1$  and  $B2$  of different or equal sizes [39]. The solution set  $B2$  is removed from  $\text{Ref}_{\text{set}}$ . DGM is reinitialized with the goal of generating solutions that are diverse with respect to  $B1$ . Thereafter, DGM is used to construct new  $P$  solutions.  $B2$  solutions in  $\text{Ref}_{\text{set}}$  are sequentially selected from  $P$  to maximize the diversity. To solve this problem, some distance measure is used.  $\text{Ref}_{\text{set}}$  is rebuilt to increase the diversity.

**4.5. Solution Combination Strategy.** Constructive linking and vocabulary building methods are used for solution combination [39, 40]. The constructive linking utilized the valuable neighborhoods for making offspring from the collection of parent solutions. The guiding solutions remain the same, but the initial solutions start as null or incomplete solutions.

## 5. Variants of SS

In the last few years, researchers have made various modifications in SS to deal with different research problems. The variants of SS are broadly categorized into four classes such as improved, hybrid, chaotic, and multiobjective (see Figure 7). Figure 8 shows the percentage distribution of SS variants according to research articles. The summary of SS's variants is illustrated in Table 5.

**5.1. Improved SS.** Remli et al. [41] developed an enhanced SS (ESS) for solving the parameter estimation problem in biochemical systems. Three modifications were done in  $\text{RefSet}$  formation,  $\text{RefSet}$  exploitation, and  $\text{RefSet}$  combination. ESS was further combined with opposition-based learning (ESSOL) to improve the performance. ESSOL was tested on 116 kinetic parameters in Chinese hamster ovary cells. This method provided the best solution with minimum computational time. Laguna and Martí [42] studied various updating mechanisms for reference set of SS. These mechanisms were used to maintain the balance between diversity and intensification during the search process. The performance of SS was evaluated on 40 different test problems. SS was able to generate the optimal solution for 30 out of 40 problems. Herrera et al. [43] introduced a continuous SS (CSS) based on the integration of BLX- $\alpha$  and improvement mechanisms such as Solis and Wets algorithm and Nelder–Mead simplex algorithm. The exploration capability of the Solis and Wets algorithm was used to enhance the best solutions for complex problems, whereas the exploration capability of the Nelder–Mead simplex algorithm to refine the best solutions for simple problems. The performance of CSS was assessed by testing it on 45 test functions and three

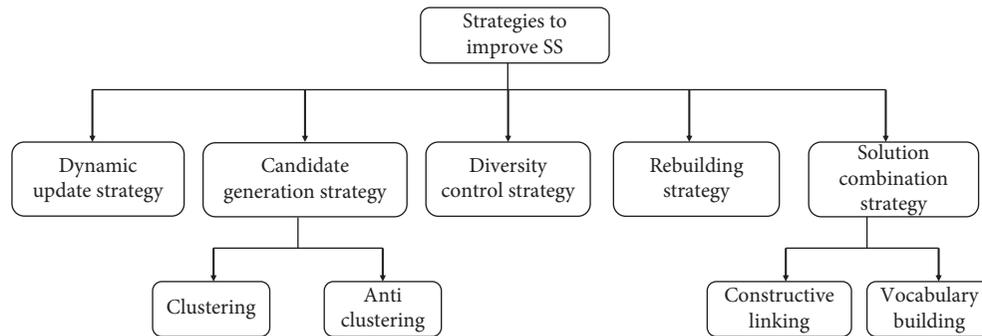


FIGURE 6: Strategies used to improve the design structure of SS.

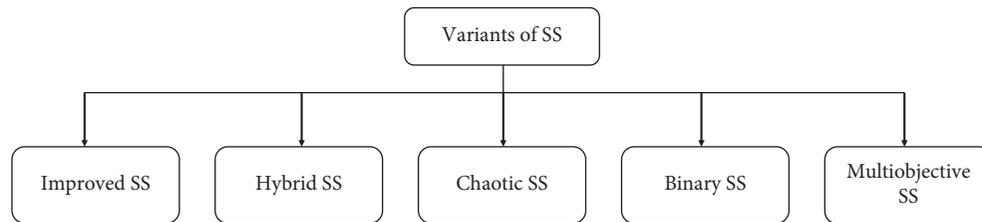


FIGURE 7: Variants of SS.

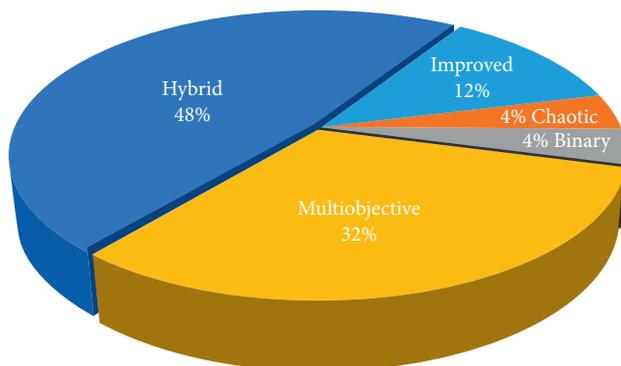


FIGURE 8: Percentage of SS variants according to research articles.

real-life problems. The comparison results revealed the supremacy of SS over the existing techniques. Another version of continuous SS is presented in [44, 45].

**5.2. Hybrid SS.** Ugray et al. [46] presented an OQNLP to find the global optimal solution for mixed-integer problems. OptQuest known as commercial implementation of SS was used in OQNLP. Local NLP solver was incorporated in OptQuest to improve the search capability for optimization problems. OQNLP evaluated on 100 variables and 100 constraints. OQNLP was able to generate the optimal solutions for all problems. Sagheer et al. [47] implemented a hybrid algorithm based on the bee algorithm and SS to handle traveling salesman problem. The bee algorithm was incorporated in the improvement method to enhance the search capability of SS. This approach (Bee-SS) provided better solutions than the SS. However, the computational time of hybrid approach is larger than the SS. In [48], SS and GA were integrated to solve the knapsack problems. The

crossover and mutation operators were used to replace the combination and improvement components of SS. The hybrid approach (GA-SS) attained the best solution for knapsack problem. However, the computational time of hybrid algorithm is little bit higher than the SS.

Al-Obaidi [49] combined cuckoo search (CS) and SS (CS-SS) to solve the traveling salesman problem. CS was incorporated in SS to enhance the improvement method. The hybrid algorithm provided 23.2% improvement over SS in terms of fitness value. A new hybrid algorithm based on SS and variable neighborhood search (VNS) was proposed in [50]. VNS was used as an improvement procedure to improve the solutions. VNS-based SS (VNS-SS) performed better than the other algorithms. Alsaidi et al. [51] proposed a hybrid algorithm that combined SS and meerkat clan algorithm (MCA-SS). MCA was used to enhance the performance of the reference set and improvement method. The proposed algorithm was applied on traveling salesman and job shop scheduling problems. The experimental results revealed that the proposed hybrid algorithm was able to provide competitive results over the existing algorithms. El-Fallahi and Martí [52] integrated TS and SS (TS-SS) to solve the training problem of feedforward neural networks. TS incorporated elements for better balance between exploration and exploitation. The hybrid method utilized context information for move selection. The results indicated the effectiveness of TS-SS to solve the training problem of neural network over the existing algorithms. In [53], SS combined with direction rounding (DR-SS) to solve 0–1 mixed-integer programming problems.

Davendra et al. [54] implemented a hybrid algorithm based on population algorithms (eDE-SS) to solve quadratic assignment and flow shop scheduling problems. The performance of the proposed hybrid algorithm was validated by using experiment methodology. The performance of the

TABLE 5: Summary of scatter search's variants.

Type	Algorithm	Description	Application	Performance	Ref.
Improved	ESSOL	Modification is done in reference set by using opposition-based learning scheme	Kinetic models of biochemical systems	Faster convergence rate, avoid local optima	[41]
	SS	Improvement in reference set	Optimization problems	Improvement in performance	[42]
	CSS	BLX- $\alpha$ operator, Solis and Wets algorithm, and Nelder–Mead simplex are employed to improve the performance of SS	Optimization problems	Better exploitation, better efficiency	[43]
Hybrid	OQNLP	Local NLP solver is employed to enhance the global search ability of SS	Optimization problems	Improvement in performance	[46]
	Bee-SS	Bee algorithm is integrated with SS to enhance the improvement method	Traveling salesman problem	More computational time than SS, better performance	[47]
	GA-SS	Crossover and mutation are used to replace combination and improvement	Knapsack problem	Better exploration capability	[48]
	CS-SS	CS is employed to enhance the improvement method	Traveling salesman problem	Better performance	[49]
	VNS-SS	VNS is used to replace both intensification and improvement components of SS to improve the quality solutions	Integer programming problem	Improvement in performance	[50]
	MCA-SS	MCA is used to update the reference set and improvement method of SS	Traveling salesman and job shop scheduling problems	Better exploration capability	[51]
	TS-SS	TS is used to modify the search element for better exploration and exploitation	Feedforward neural network problem	Better exploitation and exploration capability	[52]
	DR-SS	DR is employed to improve the search capability of SS	Mixed-integer programming	Improved performance	[53]
	eDE-SS	Enhanced DE is used to retain the quality solutions in long-term memory	Quadratic assignment and flow shop scheduling problems	Improve the effectiveness of SS	[54]
	DE-SS	Population reinitialization method is used to avoid the local optima	Optimization problems	Robust, better performance	[55]
Chaotic	ACO-SS	ACO is incorporated in SS to improve the search ability	Multicast routing problems	Minimize the cost	[125]
	ChSS	Four chaos maps are employed to generate CPRNG for SS	Flow shop problem	Better exploration capability	[56]
Binary	BSS	Modification in diversification generation, improvement, and combination methods	Knapsack problem, maximum diversity problem, max-cut problem	Improvement in performance	[57]
Multiobjective	AbYSS	Pareto dominance, density estimation, and external archive are incorporated in SS to store nondominated solutions	Optimization problems	Improve the performance	[58]
	SSMO	Ranking, crowding, and Pareto dominance are incorporated in SS	Optimization problems	Improve the performance	[59]
	MOSS	TS is used in the diversification generation method to generate Pareto-optimal solutions	Optimization problems	Better convergence, reasonable computational time	[60]
	SSPMO	TS is employed in SS to improve the Pareto-optimal solutions	Optimization problems	Better performance	[61]
	MPSOSS	A new leader selection is used to improve the convergence. PSO is incorporated in SS to improve spread	Optimization problems	Better convergence rate, improve spread	[62]
	RP-MOSS	A new preference-based TS and SS are introduced	Optimization problems	Better Pareto-optimal solutions	[63]
	MOSS-II	TS and convex combination methods are used as the diversification generation method. A constraint handling mechanism is employed to handle multiple constrains	Optimization problems	Improved performance	[64]
	MOSSSA	SA is employed as an improvement method. Fast archiving strategy is used to store the set of nondominated solutions	Water distribution network	Improved quality of network	[65]

proposed hybrid algorithm clearly indicate the superiority of eDE-SS than some existing evolutionary algorithms in terms of proximity and diversity. Li and Tian [55] combined differential evolution (DE) and SS to solve the global optimization problems. The multiple mutation operators of DE were used to select the most appropriate mutation operator during the evolution process. The results revealed the superiority of the proposed hybrid algorithm over the other algorithms.

**5.3. Chaotic SS.** Davendra et al. [56] modified SS by incorporating the four different chaotic maps such as Burgers, Lozi, Tinkerbell, and Delayed Logistics. The chaos pseudorandom number generators (CPRNGs) were produced from chaotic maps. CPRNGs were utilized in SS to enhance the search capability of SS. The chaotic SS (ChSS) was used to solve the flow shop problem. The results of ChSS were compared with Mersenne Twister. The results revealed that the ChSS outperforms Mersenne Twister.

**5.4. Binary SS.** Gortázar et al. [57] proposed the adaptation of SS to solve the binary optimization problems. The binary-coded SS (BSS) was used to improve the exploitation process. Three diversification generation methods were designed to generate the binary vectors. An improvement method based on two neighborhoods was proposed to produce feasible solutions. Seven different combination methods were proposed to generate high-quality solutions. The effectiveness of BSS was evaluated on 376 problem instances. In terms of validation, BSS performed better than other techniques.

**5.5. Multiobjective SS.** Nebro et al. [58] developed an archive-based hybrid scatter search (AbYSS) to solve the multiobjective problems (MOPs). AbYSS incorporated the concepts of Pareto dominance, density estimation, and external archive. External archive was used to store the nondominated solutions. AbYSS was evaluated on benchmark test functions. The experimental results indicated the effectiveness of AbYSS to solve test functions over the existing algorithms. Nebro et al. [59] introduced SS to solve the MOPs. A nondominated sorting procedure was used to construct the reference set from the initial set. A mutation operator-based on local search was used to enhance the solutions obtained from the reference set. The multiobjective SS (SSMO) was evaluated on benchmark test functions. The experimental results indicated the superiority of SSMO over the existing multiobjective algorithms. Beausoleil [60] introduced a hybrid optimization algorithm based on integration of tabu and scatter search for solving MOPs. Tabu tenures were incorporated in SS to improve the diverse approximation to the Pareto-optimal solutions. The reference set was formed after each iteration of SS. A weighted sum approach was used to compute the solution quality. The results showed the effectiveness of MOSS in solving optimization problems. MOSS provided the better convergence than the other multiobjective algorithms.

Molina et al. [61] proposed a multiobjective SS (SSPMO) that utilizes the concept of TS. In SSPMO, the reference set was obtained by choosing the optimal solutions from the initial set for each objective function. The remaining part of reference set was obtained by selecting the remaining optimal solutions in the set that maximized the distance to the solutions present in the reference set. The updating process was done in the initial set for obtaining the better quality solutions. SSPMO was able to provide the good Pareto-optimal solutions. Santana-Quintero et al. [62] presented a multiobjective evolutionary approach (MPSOSS) that utilizes the concept of PSO and SS. A new leader selection mechanism of PSO was used to enhance the convergence rate. SS was used to improve the nondominated solutions. The performance of MPSOSS was evaluated on benchmark test functions. MPSOSS outperforms NSGA-II in terms of convergence and quality solutions. Beausoleil [63] utilized the preference based on the reference points to design multiobjective tabu/scatter architecture (RP-MOSS). A novel mechanism was developed to generate the subsets of solution for designing the reference set. RP-MOSS was able to generate the better Pareto-optimal solutions than the other algorithms.

In [64], a new version of multiobjective SS (MOSS-II) was developed to solve constrained problems. A constrained handling mechanism was presented to handle the multiple constraints. MOSS-II utilized TS and convex methods for the diversification generation component of SS. MOSS-II was able to generate the feasible solution for constrained problems. Baños et al. [65] proposed a multiobjective metaheuristic (MOSSSA) that combined SS and SA to solve water distribution network problem.

## 6. Scatter Search Applications

In the last few decades, SS has been used effectively in different research domains. The applications of SS are classified into following categories: data mining, image processing, bioinformatics, civil engineering, mechanical engineering, electrical engineering, industrial engineering, networking, healthcare, and software engineering. Figure 9 depicts the applications of SS in different problem domains. The overview of different applications of SS is presented in Table 6.

**6.1. Scheduling.** Naderi and Ruiz [66] designed a modification in SS for distributed permutation flow shop scheduling problem (DPFSP). They used hybrid RefSet for solutions and job assignment vectors. These solutions and job assignment vectors were combined in the solution-combined procedure. This approach was tested on 720 large instances and obtained 589 new best solutions. Rahimi-Vahed et al. [67] utilized a MOSS to solve FSP that minimizes the completion time and tardiness simultaneously. An adaptive archive updating mechanism was proposed to alleviate nondominated solutions. The performance of MOSS was better than the strength Pareto evolutionary algorithm (SPEA-II). However, the computational time of MOSS is 3.5 times than the SPEA-II. Nowicki and Smutnicki [68] proposed a modified SS (MSS) for FSS with makespan criteria. They used big valley

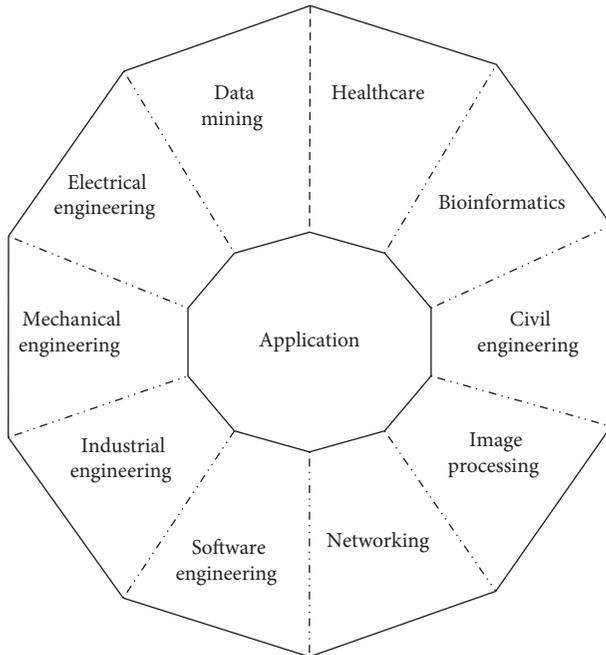


FIGURE 9: Applications of SS.

phenomenon. MSS was evaluated on 120 hard instances of twelve different sizes. MSS outperformed the existing techniques in terms of accuracy. Geng et al. [69] incorporated PSO in SS to minimize the earliness and tardiness constraints of FSP. DE was also used to produce the quality solutions in SS. Ten different jobs were used to assess the performance of this hybrid SS approach. This approach has better search capability and convergence than PSO and GA. Hariharan and Nimal [70] combined GA and SS to minimize the completion time of PFSP. Twenty benchmark problems were used to evaluate the performance of combined approach. The combined approach provides better results than the other techniques. However, this approach did not consider the two important criteria such as due date and setup time. González et al. [71] hybridized tabu search with SS to solve the FSP. The neighborhood structures and distance measures were used to predict the completion time. This approach was able to enhance the 58 solution instances from 178 FSP instances [72].

Sari et al. [73] evaluated both GA and SS on project scheduling problem (PSP). As many as 100 benchmark test problems were used to assess the performance of these algorithms. The increased number of iterations is significant in SS, but not in GA. Alvarez-Valdes et al. [74] designed a modification in SS for PSP with renewable resource constraint. The preprocessing techniques were used to reduce the number of variables and constraints. The proposed approach was able to obtain the optimal solutions for 3826 Schirmer test instances. Hiermann et al. [75] evaluated the performance of variable neighborhood search (VNS), memetic search (MS), SS, and SA for solving home healthcare scheduling problem. A random procedure was used to generate the initial solutions. Thereafter, the initial solutions were improved through the VNS, MS, SS, and SA. MS provides better solutions than SS, SA, and VNS.

Jaradat and Ayob [76] used an SS to solve the timetabling problem. The reference set and combined solution procedures were used to maintain diverse solutions. A probabilistic selection method was used to produce the elite solutions. The optimal solutions were generated through iterated local search process. SS was tested on Socha's instances and attained the good results as compared to the existing techniques. Mansour and Sleiman-Haidar [77] proposed a parallel SS algorithm for generating the optimal exam timetables within reasonable time duration. The candidate solutions were distributed over a number of processors. The procedures of SS were computed in parallel fashion. SS produced quality solutions in reasonable time duration. Mohammed et al. [78] utilized SS for producing the optimal exam timetables for Anbar University. SS was able to generate the satisfactory solutions for timetabling problems [79]. Mansour [80] proposed a heuristic technique based on SS for finding the optimal solutions for exam timetabling. This approach was tested on Lebanese American University and produced better examination timetables as compared to manual and other techniques such as SA and GA. Martí et al. [81] used an SS to assign the proctor to final examination of Universitat Pompeu Fabra. The solutions obtained from SS were compared with the CPLEX model. The solutions obtained from SS were inferior from the CPLEX model. However, the reference set of SS provides a large number of quality solutions.

**6.2. Data Mining.** López et al. [82] proposed a parallel SS to select the optimal features for classification. Instance-based learning, Naïve-Bayes, and C4.5 were incorporated in SS. The proposed approach was assessed on twelve UCI repository datasets. The parallel SS attained better accuracy than the sequential SS. The proposed approach achieved 76.2% average reduction in feature subsets. Wang et al. [83] combined both SS and rough set for feature selection. The radial basis function, logistic regression, and J48 were integrated in the hybrid approach. The conditional entropy was used to assess the solution quality. The Australian credit dataset and Japanese consumer credit dataset were used to evaluate the performance of proposed approach. The proposed approach achieved the classification accuracies of 88.9% and 90.5% for Australian credit and Japanese consumer credit datasets, respectively. Adi [84] introduced the parallel SS for classical problem. For the Gisette dataset, parallel SS enhanced the classification performance from 0.97 to 0.99. The execution time was reduced from 120 hours to 40 hours. Duman and Ozcelik [85] hybridized GA and SS to minimize the wrongly classified transactions for credit card fraud detection.

**6.3. Healthcare.** Boumedine and Bouroubi [86] proposed an SS to envisage the best conformation from the given sequence of amino acid. SS provided better results than the other algorithms in terms of low energy. Mansour et al. [87] presented an SS for prediction of 3D structures of proteins. The fitness function was used to minimize the energy function associated with protein structure. Protein Data Bank was used to assess the performance of SS and produce

TABLE 6: Summary of scatter search's applications.

Application	Problem	Algorithm	Reference
Scheduling	Flow shop scheduling	SS, MOSS, MSS hybrid SS	[66–72]
	Project scheduling	Hybrid SS	[73, 74]
	Home-care scheduling	SS	[75]
	Course timetabling	SS	[76]
	Exam timetabling	SS, hybrid SS, parallel SS	[77–81]
Data Mining	Feature selection	Parallel SS, hybrid SS	[82–84]
	Credit card fraud detection	Hybrid SS	[85]
Healthcare	Protein structure prediction	SS	[86, 87]
	Nurse rostering problem	SS	[88, 89]
	System biology	SS	[90]
Image processing	Image registration	SS	[91–93]
Industrial engineering	Vehicle routing problem	SS, hybrid SS	[94–102]
	Facility layout problem	SS	[103–105]
	Territory design problem	SS, MOSS	[106, 107]
	Assembling/disassembling sequence problem	SS, MOSS	[108–112]
	Knapsack problem	SS, hybrid SS	[113–116]
Electrical engineering	Economic load dispatch	SS, MOSS	[117, 118]
	Power distribution	SS	[119]
Bioinformatics	DNA sequencing	Hybrid SS	[120]
	Gene expression	SS	[121]
	Chemical process optimization	SS	[122]
Networking	Routing problem	SS, hybrid SS	[123–125]
	Network planning problem	SS	[126–129]
Software engineering	Software testing	SS	[130–133]
	Software configuration	SS	[134, 135]
Mechanical engineering	Manufacturing system	SS, hybrid SS	[136–138]
	Design problem	SS	[139–143]
	Fault detection	MOSS	[144]
Civil engineering	High toll identification	SS	[145]
	Space allocation problem	ISS, hybrid SS	[146, 147]
	Runway operation	Hybrid SS	[148]
Engineering optimization	Linear ordering problem	SS	[149, 150]
	Graph coloring problem	SS	[151]
	Quadratic assignment problem	SS	[152]
	p-median problem	Parallel SS	[153]
	Permutation problem	Hybrid SS	[154]
	Regret location problem	Parallel SS	[155]
	SAT problem	SS, hybrid SS	[156, 157]
	Clique problem	SS	[158]
	Nonlinear optimization process	SS	[159–161]
Miscellaneous	BDDs minimization	SS	[162]
	Craniofacial superimposition	SS	[163]
	Joint replenishment	SS	[164]
	Kinetic models of biochemical	SS	[165]
	Cockpit crew rostering	Hybrid SS	[166, 167]
	E-mail spam detection	Hybrid SS	[168]
	Subset reduction	Hybrid SS	[169]

3D structure with reasonable energy value. Burke et al. [88] presented SS for assigning the shifts to nurse, which ensure that sufficient nurses are available to perform her duties. SS provided quality solutions for seven out of ten test instances within reasonable time duration. Maenhout and Vanhoucke [89] proposed an SS for nurse scheduling. The fitness function was designed by considering the time preference, coverage constraints, and case-specific constraints. The proposed approach was able to generate the optimal schedule for nurses. Egea et al. [90] developed a MEIGO

toolbox for solving problems associated with bioinformatics and system biology. They used enhanced SS, VNS, linear programming, and integer programming. The enhanced SS and VNS can be executed in sequential or parallel fashion.

*6.4. Image Processing.* Cordon et al. [91] used SS to solve the 3D image registration problem. This approach was tested on the BrainWeb database. SS provides better results than the improved iterative closet point matching (I-ICP). Cordon

et al. [92] developed a mechanism for SS to combine and enhance the solutions for 3D image registration problem. A procedure was developed to maintain the balance between exploration and exploitation in reference set. The preprocessing of images was used to enhance the performance of SS. Both MRI and CT scan images were used to judge the efficiency of proposed SS. The proposed approach outperforms the existing techniques. Santamaría et al. [93] applied SS to solve the pairwise 3D image registration problem. SS was able to provide good results for 3D model reconstruction. However, it suffers from some shortcomings during 3D reconstruction.

**6.5. Industrial Engineering.** Euchi [94] hybridized GA and SS to determine the optimal routes that satisfy the demands of customers and suppliers with minimum transportation cost. SS was used to perform the local search. The hybrid approach was tested on 100–500 customer instances. This approach produced better solutions with in reasonable time. Belfiore and Yoshizaki [95] used SS to solve vehicle routing problem (VRP) with time windows and split deliveries. This approach was applied on 519 stores in eleven Brazilian states. SS offered better solutions than the existing techniques in terms of trucks and distribution cost. Tang et al. [96] applied SS on VRP. They used transportation cost and vehicle load to design the fitness function. The vehicle load considered as a variable factor for customers in each trip. The nearest neighbor method was used to improve the solutions. SS was able to produce good solutions for VRP. SS was successfully used to solve the variants of VRP [97–102].

Hakli and Ortacay [103] implemented an improved version of SS (ISS) to solve incapacitated facility location (UFL). The crossover and mutation operators were incorporated in SS to enhance the search capability. ISS was applied on fifteen UFL problems from the OR-Lib dataset. This method shows superiority over the existing techniques. SS has been presented in [104] to solve facility layout problem. The performance of SS was tested on 58 large size instances. The results revealed that SS solved UFL problems and was able to reduce execution time. Xue and Li [105] used SS to solve container-loading problem. SS integrated with multiple neighborhood structures to improve the search ability of SS. SS attained optimal solution. Salazar-Aguilar et al. [106] proposed a MOSS to design the biobjective territory. The greedy random search procedure was used to improve the diversification generation component. The relinked local search was incorporated in an improvement component. The proposed approach was tested on large instances and outperformed the existing techniques in terms of feasible solutions. In [107], SS was used to select strip coils in shipping transportation.

In [108], SS applied on disassembly sequence problem. SS was evaluated on 48 different products with up to 100 components. SS has shown 2% average improvement over REVERSE. Rahimi-Vahed et al. [109] implemented an MOSS to optimize utility work, product rate variation, and setup cost for assembly line sequencing problem. MOSS generated diverse nondominated solution that is closer to

true Pareto front. However, the computational time of MOSS was 5.5 times than the MOGA. An improvement in SS is introduced in [110] for solving disassembly sequence problem. The solution improvement operator was used to enhance the solution quality. The improved SS (ISS) has less computational time than NSGA-II. ISS produced better optimal solutions than NSGA-II. However, this method did not consider the actual disassembly data. SS has successfully used to solve the mixed-model assembly line sequencing and knapsack problems [109, 111–116].

**6.6. Electrical Engineering.** Mizutani et al. [117] proposed an improvement in SS to forecast the correlation between input and output of power load. The weather information was incorporated in the proposed approach. The average correlation obtained from the proposed approach was 0.88. Costa e Silva et al. [118] developed an improved version of SS (ISS) to solve multiobjective environment/economic dispatch problem. The concepts of Pareto dominance and crowding were incorporated in SS. A new combination method strategy was proposed. The proposed approach was evaluated on the IEEE 30-bus six-generator system. ISS provided better solutions than the SS in terms of Pareto solutions. Posada et al. [119] implemented an SS to find the optimal location and price of distributed generation systems. The performance of SS was evaluated on 34 buses. The proposed approach provided 5.8% extra profit than the memetic algorithm.

**6.7. Bioinformatics.** Blazewicz et al. [120] integrated TS with SS to solve DNA sequencing problem. This approach works more efficiently and accurately to detect accurate sequence in comparison with TS, branch-and-bound, and hybrid GA algorithms. Nepomuceno et al. [121] introduced SS to determine biclusters from gene expression data. The correlation between genes was used to design a fitness function. The designed function was able to detect the different patterns from genes. This approach was evaluated on Yeast Cell Cycle, Human B-Cell Lymphoma, and Yeast Stress datasets. The performance of the proposed approach was better than the existing techniques in terms of fitness function. Egea et al. [122] implemented SS to solve the optimization process in chemical processes. Rebuilding was a key operation in reference set generation. The modifications have been done in the five components of SS. The experimental results revealed the supremacy of proposed approach over the other algorithms.

**6.8. Networking.** Greistorfer [123] utilized the SS in tabu search to solve the Chinese postman problem. The improvement procedure was modified. The solution combination method was used to combine the set of elite solutions. The hybrid approach was able to generate better solutions than the other techniques in terms of quality and computational time. Xu and Qu [124] introduced a new approach based on SS and path relinking (SSPR) to handle multicast routing problem. TS and VNS were used as local search for

further improvement in SSPR. The benchmark instances taken from OR-library were used to evaluate the performance of SSPR. The results indicated that the SSPR with VNS performed better than the SSPR with TS in terms of cost. SSPR outperforms the existing techniques in terms of tree cost. Chunxin et al. [125] presented a hybrid approach (ACO-SS) for multicast routing problem. ACO was embedded in SS to avoid the premature convergence by using pheromone trails. The pheromone trail updating strategy was used to improve the solution quality. Twenty transmission nodes were used to assess the performance of ACO-SS. ACO-SS offers better solution than ACO in terms of average and minimum cost.

Alvarez et al. [126] combined SS and GRASP to determine the feasible solutions for capacitated network design problem. The proposed approach was tested on different network instances. The comparison results showed the superiority of hybrid approach over CPLEX and Dual descent approaches in terms of optimal solution. In [127], GA was embedded in SS to generate better transmission solution for network expansion planning problem. Perez et al. [128] introduced SS to deal with deployment of dense wavelength division multiplexing (DWDM) in the place of fiber network. Diversification generation component of SS was used to allocate the spare capacity of existing network to demand raised by customers. SS was able to handle restoration problems associated with the network. Alvarez et al. [129] utilized SS to solve the multicommodity network design problem. It was shown from simulation results that SS has ability to determine the good solutions for large-scale problem within sufficient time.

*6.9. Software Engineering.* Blanco et al. [130] used SS to design the automatic test generators. The two different strategies were proposed for automatic generation of test cases. In the first strategy, a diversity mechanism was used to explore all the branches of program. The second strategy combined both diversity and local search methods to explore the unreachable branches of program. SS was able to generate the suitable test cases for branch coverage. SS and estimation of distributed algorithm (EDA) have been integrated by Sagarna and Lozano [131] to generate test cases automatically. The best solutions obtained from EDA were applied on SS generator. This approach suffers from diversity problem in some cases. SS has an ability to recover from this problem during rebuilding phase of the diversification method. In [133], SS was used to produce the test cases for covering all the branches of natural language processing (NLP) programs. The convergence rate of SS was better than the existing algorithms. SS was able to explore the all-possible branches of NLP programs.

Bouridah and Belhadef [134] introduced SS to configure service function of network function virtualization. SS was able to produce the optimal number and placement of virtual functions. Yamashita et al. [135] used SS to find the optimal resource availability by considering the deadline and relations between project activities. 2400 instances were used to assess the performance of proposed approach. The average

gap obtained from SS was 0.12% and 0.18% for 30 and 20 activities of project, respectively. The optimal solutions obtained from SS were 92% and 86% of problems from 30 to 20 activity sets, respectively.

*6.10. Mechanical Engineering.* Chinnusamy et al. [136] proposed an improved SS to design the optimal schedule for flexible manufacturing systems (FMSs). The dispatching rules were combined with Perti Net. The proposed approach performed better than the other techniques in terms of machine utilization. Krishna and Rao [137] implemented SS to find out optimal grinding parameters such as wheel speed, speed of work piece, dressing depth, and dressing lead. The weighted combination of production cost and surface finish was used to design the fitness function. The performance of SS was compared with GA, ACO, and quadratic programming. It is observed from results that SS outperformed the other algorithms. Prabu et al. [138] implemented SS to determine the optimal fixture layout with least deformation. The modification in diversification generation method was made to consider all layouts for the improvement method. SS performed better than GA in terms of computation time and convergence.

Francois et al. [139] introduced SS to determine the optimal solutions for boiling water reactor problem. Tan et al. [140] studied the scheduling problem of steel making continuous casting. They integrated both SS and mixed-integer programming (MIP) to solve this problem. SS was used to find out the assignment and sequence variables for charges. Based on the results obtained from SS, MIP designed a constraint network. Thereafter, the constraint network was applied on optimization software. The hybrid approach was able to solve the scheduling problem. The various versions of SS were investigated to solve the permutation problems and MIP problems [141, 142]. Su et al. [143] integrated SS and neural networks to optimize the parameter optimization problem. The relationship between values of control factor and their measures was evaluated using neural networks. This relationship was applied on SS to obtain the optimal parameter setting. A desirability function was designed to convert several responses into single one. The applicability of hybrid approach was tested on blood strip manufacturing company situated in Taiwan. The results showed the superiority of hybrid approach over the other algorithms. Le et al. [144] used SS to determine fault in the design of Network-on-Chip (NOC). The simulation results revealed that the SS algorithm attained low-power consumption, high reliability, and less communication time as compared to GA and PSO.

*6.11. Civil Engineering.* González-Velarde et al. [145] introduced SS to increase the number of tolls and maintain the optimal traffic flow for transportation network. Zhang et al. [146] proposed an improvement in SS (ISS) to solve the corridor allocation problem. The improvement procedures were designed to handle the specific characteristics of allocation problem. The solution obtained from ISS was similar to GUROBI software. The experimental results

revealed that ISS performed better performance than SA and SS in terms of solution quality. Beasley [147] combined SS and bionomic algorithms to determine the optimal landing sequence and scheduling of each aircraft. The performance of hybrid approach was assessed on 500 aircraft and 5 runways. The hybrid approach was able to produce the feasible solutions for aircraft landing problem. Soykan [148] presented a hybrid TS and SS approach to determine the optimal solutions for runway operation scheduling problem.

**6.12. Engineering Optimization.** Campos et al. [149, 150] implemented SS to solve the linear ordering problem. The improvement mechanisms were designed to combine solutions and maintain the balance between exploration and exploitation in reference set. SS was tested on both small and large instances of LOLIB. The computational time obtained from SS was reduced by 7% as compared to duplication checking mechanism. SS provided results better than the other techniques in terms of partial deviation. Hamiez and Hao [151] presented SS to solve graph coloring problem. An improvement was made in a set of configurations. The new configurations were replaced with others in terms of quality and diversity. The experimental results proved that SS was able to provide the optimal solutions for graph coloring problem. Cung et al. [152] proposed SS to handle quadratic assignment problem. A new operator was proposed to improve the solution quality. SS was evaluated on test problems of QAPLIB. The results showed that SS was able to provide the optimal solution.

Garcia-López et al. [153] developed new strategies for parallelization of SS to enhance exploration and performance. The parallelization of SS was tested on large instance of p-median problem obtained from TSPLIB. The proposed approach provided best solution within reasonable time. Campos et al. [154] integrated both SS and TS to solve the permutation problems. This approach was compared with two commercial software and attained the optimal solution for permutation problems. Ibri et al. [155] proposed a parallel SS to solve the regret location problem. Results proved that the parallel SS performed better results than the existing algorithms in terms of computational time and solution quality. In [156, 157], SS was implemented to solve SAT and W-MAX-SAT problems. Cavique et al. [158] developed an SS framework to solve the maximum clique problem. The guidance search and local search were incorporated in SS. SS provided the better feasible solutions than the competitive algorithms.

Egea and Balsa-Canto [159] developed an enhanced SS (ESS) to handle the noise and discontinuities of nonlinear processes. A local search procedure was incorporated to speed up the convergence of quality solutions. The performance of ESS was assessed on four different test problems and attained the better efficiency and scalability over the existing state-of-art techniques. In the recent years, the variants of SS were applied on cardinality constrained and dynamic models [160, 161].

**6.13. Miscellaneous.** Hung et al. [162] investigated SS to minimize the reduced-ordered binary decision diagrams (BDDs). SS was able to provide the optimal BDD size than the SA and GA. Ibanez et al. [163] studied craniofacial superimposition technique in which images of a missing individual are compared with skull fund to determine whether the same individual or not. SS was used to develop a skull face overlay method. This method was tested on six different cases, which were taken from University of Granada. The method showed fast and accurate performance. Gutierrez et al. [164] investigated the performance of SS and RAND on joint replenishment problem. SS performed better than RAND for most of the cases. In some cases, SS produced the worse solution than RAND. However, the worse solution was not more than 0.119% worst solution of other algorithms. In [166, 167], SS was utilized to solve the cockpit crew rostering problem. Gharehchopogh and Amjad [168] hybridized SS and k-nearest neighbors (KNN) to handle the e-mail spam detection. The simulation results revealed that the hybrid approach performed better than C4.5 and provided 94.54% accuracy with 57 features. Wang et al. [169] proposed SS for attribute reduction in rough set. Thirteen different datasets were used to assess the performance of SS. The simulation results revealed that SS was able to provide quality solution for attribute reduction.

## 7. Future Research Directions

In last few years, researchers have modified the design structure of SS by using control strategies and integrating the other optimization algorithms. Besides this, there are still few open research areas. The future research directions of SS are given below.

**7.1. Many-Objective Optimization.** SS has not been applied on many-objective optimization problems. SS used Pareto-optimal dominance concept to solve multiobjective problems. The concepts of reference vector adaption strategy, knee points, and hypervolume have to be incorporated in SS to show its efficiency for many-objective problems. There is need to investigate the adaptive archive strategy in SS.

**7.2. Binary Optimization.** In binary SS, the representation of solution vectors is transformed in binary form. There is a need to use the different transfer functions in SS to solve binary problems. S-shaped and V-shaped functions have to be incorporated in SS to check the performance of SS. Recently, Z-shaped functions are developed. These functions have to be added in SS to evaluate the performance of SS in binary space.

**7.3. Theoretical Evaluation of SS Structure.** The theoretical analysis of modified and hybrid SS has not done by researchers. Therefore, theoretical evaluation of SS is required for detailed analysis. Besides this, the theoretical study of control strategies and structure of SS is required.

**7.4. New Data Modalities.** SS has limited applicability in image processing problems. Yet, SS has shown good results on image registration for 3D image scan. SS has to be applied on other problems such as image segmentation, filtering, and edge detection. Besides this, the applicability of SS has to be tested on different data modalities such as CT scan, MRI, and X-ray images for medical imaging.

## 8. Conclusion

A comprehensive review of the SS algorithm is presented in this paper. The main focus of this paper is to summarize the variants and applications of SS. The variants of SS, namely, improved, hybrid, binary, chaotic, and multiobjective assisted to improve the performance of SS. SS has advantages over the other algorithms in terms of simplicity, flexibility, and hybridization with other algorithms. However, SS is being stuck in local optima for some cases. The appropriate selection of control strategies helped to improve the exploration and exploitation properties of SS. SS has been successfully applied on engineering design problems, data mining, bioinformatics, and networking problems. The future research directions are provided in this paper, which can motivate the innovative researchers for further research in their research domains. The researchers can work on SS to find out its efficiency and effectiveness on their research problems.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

## Acknowledgments

The authors are thankful to Deanship of Scientific Research at King Khalid University for awarding project ID: RGP.2/190/42 and titled Advanced Computational Methods for Solving Complex Computer Science and Mathematical Engineering Problems.

## References

- [1] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.
- [2] M. Dubey, V. Kumar, M. Kaur, and T.-P. Dao, "A systematic review on harmony search algorithm: theory, literature, and applications," *Mathematical Problems in Engineering*, vol. 2021, Article ID 5594267, 22 pages, 2021.
- [3] V. Kumar and D. Kumar, "A systematic review on firefly algorithm: past, present, and future," *Archives of Computational Methods in Engineering*, vol. 28, pp. 3269–3291, 2021.
- [4] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [5] V. Kumar, J. K. Chhabra, and D. Kumar, "Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems," *Journal of Computational Science*, vol. 5, no. 2, pp. 144–155, 2014.
- [6] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: from Natural to Artificial Systems*, Oxford University Press, Inc, Oxford, UK, 1999.
- [7] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [8] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [9] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [10] J. R. Koza, "Genetic programming: on the programming of computers by means of natural selection," *Statistics and Computing*, vol. 4, no. 2, 1994.
- [11] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.
- [12] F. Glover, "Tabu search—part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.
- [13] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [15] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-Learning-Based Optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences*, vol. 183, no. 1, pp. 1–15, 2012.
- [16] R. A. Formato, "Central force optimization: a new metaheuristic with applications in applied electromagnetics," *Progress in Electromagnetics Research*, vol. 77, pp. 425–491, 2007.
- [17] O. K. Erol and I. Eksin, "A new optimization method: big bang-big crunch," *Advances in Engineering Software*, vol. 37, no. 2, pp. 106–111, 2006.
- [18] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, November 1995.
- [20] M. Dorigo and M. Birattari, *Ant Colony Optimization*, Springer US, Boston, MA, USA, 2010.
- [21] D. Karaboga and B. Basturk, "Artificial bee colony optimization algorithm for solving constrained optimization problems," in *Proceedings of the International Fuzzy Systems Association World Congress*, pp. 789–798, Springer, Cancun, Mexico, June 2007.
- [22] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214, Coimbatore, India, December 2009.

- [23] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65–74, Springer, Berlin, Germany, 2010.
- [24] V. Kumar, K. K. Kaleka, and A. Kaur, "Spiral-inspired spotted hyena optimizer and its application to constraint engineering problems," *Wireless Personal Communications*, vol. 116, no. 1, pp. 865–881, 2021.
- [25] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *PLoS Medicine*, vol. 6, no. 7, Article ID e1000097, 2009.
- [26] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1977.
- [27] F. Glover, "Heuristics for integer programming using surrogate constraints," *Decision Sciences*, vol. 8, no. 1, pp. 156–166, 1977.
- [28] F. Glover, "Scatter search and path re-linking," *New Methods in Optimization*, vol. 19, no. 1, pp. 297–316, 1998.
- [29] F. Glover, "A template for scatter search and path re-linking," *Artificial Evolution*, vol. 1363, pp. 3–51, 1998.
- [30] F. Glover, M. Laguna, and R. Martí, "Fundamentals of scatter search and path re-linking," *Control and Cybernetics*, vol. 29, pp. 652–684, 2000.
- [31] R. Martí, A. Corberan, and J. Peiro, "Scatter search," *Handbook of Heuristics*, pp. 1–24, Springer, Berlin, Germany, 2016.
- [32] F. Glover, M. Laguna, and R. Martí, "Scatter search," in *Advances in Evolutionary Computing: Theory and Applications*, G. Ashish and T. Shigeyoshi, Eds., pp. 519–537, Springer-Verlag, New York, NY, USA, 2003.
- [33] R. Martí, M. Laguna, and F. Glover, "Principles of scatter search," *European Journal of Operational Research*, vol. 169, no. 2, pp. 359–372, 2006.
- [34] M. Laguna and R. Martí, *Scatter Search Methodology and Implementations in C*, Kluwer Academic, Dordrecht, Netherlands, 2003.
- [35] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, 1980.
- [36] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Advances in Neural Information Processing Systems*, pp. 1061–1069, MIT Press, Cambridge, MA, USA, 2012.
- [37] M. Laguna and V. A. Armentano, *Lessons from Applying and Experimenting with Scatter Search Metaheuristic Optimization via Memory and Evolution*, pp. 229–246, Springer, Berlin, Germany, 2005.
- [38] R. Martí, "Scatter search-wellsprings and challenges," *European Journal of Operational Research*, vol. 169, no. 2, pp. 351–358, 2006.
- [39] F. Glover, "Scatter search and star-paths: beyond the genetic metaphor," *OR Spektrum*, vol. 17, no. 2-3, pp. 125–137, 1995.
- [40] R. Martí, M. Laguna, and V. Campos, "Scatter search vs. genetic algorithms," in *Metaheuristic Optimization via Memory and Evolution*, pp. 263–282, Springer, Berlin, Germany, 2005.
- [41] M. A. Remli, S. Deris, M. S. Mohamad, S. Omatu, and J. M. Corchado, "An enhanced scatter search with combined opposition-based learning for parameter estimation in large-scale kinetic models of biochemical systems," *Engineering Applications of Artificial Intelligence*, vol. 62, pp. 164–180, 2017.
- [42] M. Laguna and R. Martí, "Experimental testing of advanced scatter search designs for global optimization of multimodal functions," *Journal of Global Optimization*, vol. 33, no. 2, pp. 235–255, 2005.
- [43] F. Herrera, M. Lozano, and D. Molina, "Continuous scatter search: an analysis of the integration of some combination methods and improvement strategies," *European Journal of Operational Research*, vol. 169, no. 2, pp. 450–476, 2006.
- [44] T. Trafalis and S. Kasap, "An affine scaling scatter search approach for continuous global optimization problems," *Intelligent Engineering Systems through Artificial Neural Networks*, vol. 6, pp. 1027–1032, 1996.
- [45] T. B. Trafalis and I. Al-Harkan, "A continuous scatter search approach for global optimization, extended abstract," in *Proceedings of the Conference in Applied Mathematical Programming and Modeling (APMOD'95)*, London, UK, April 1995.
- [46] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, "Scatter search and local NLP solvers: a multistart framework for global optimization," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, 2007.
- [47] A. M. Sagheer, A. T. Sadiq, and M. S. Ibrahim, "Improvement of scatter search using Bees algorithm," *Proceedings of the International Conference on Signal Processing and Communication Systems*, Springer, Berlin, Germany, 2012.
- [48] S. M. El-Sayed, W. F. A. EL-Wahed, and N. A. Ismail, "A hybrid genetic scatter search algorithm for solving optimization problems," in *Proceedings of the 6th International Conference on Informatics and Systems*, pp. 12–17, Cairo, Egypt, January 2008.
- [49] A. T. S. Al-Obaidi, "Improved scatter search using cuckoo search," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 61–67, 2013.
- [50] A. Fahim and A. Hedar, "Hybrid scatter search for integer programming problems," in *Proceedings of the 9th International Conference on Informatics and Systems*, pp. 61–69, Cairo, Egypt, December 2014.
- [51] S. A. A. Al-Saidi, D. K. Muhsen, and S. M. Ali, "Improved scatter search algorithm based on meerkat clan algorithm to solve NP-hard problems," *Periodicals of Engineering and Natural Sciences*, vol. 8, no. 3, pp. 1555–1565, 2020.
- [52] A. El Fallahi and R. Martí, "Tabu and scatter search for artificial neural networks," in *Computational Modeling and Problem Solving in the Networked World*, pp. 79–96, Springer, Berlin, Germany, 2003.
- [53] R. Todosijević, *Theoretical and Practical Contributions on Scatter Search, Variable Neighborhood Search and Metaheuristics for 0-1 Mixed Integer Programs*, Université de Valenciennes et du Hainaut-Cambresis, Valenciennes, France, 2015.
- [54] D. Davendra, I. Zelinka, and G. Onwubolu, "Hybrid differential evolution-scatter search algorithm for permutative optimization," *Evolutionary Computation*, 2009.
- [55] K. Li and H. Tian, "A DE-based scatter search for global optimization problems," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 303125, 9 pages, 2015.
- [56] D. Davendra, R. Senkerik, I. Zelinka, and M. Pluhacek, "Scatter search algorithm with chaos based stochasticity," in *Proceedings of the IEEE Congress on Evolutionary Computation*, Beijing, China, July 2014.
- [57] F. Gortázar, A. Duarte, M. Laguna, and R. Martí, "Black box scatter search for general classes of binary optimization problems," *Computers & Operations Research*, vol. 37, no. 11, pp. 1977–1986, 2010.
- [58] A. J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J. J. Durillo, and A. Beham, "AbYSS: adapting scatter search to multiobjective

- optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 4, pp. 439–457, 2008.
- [59] A. J. Nebro, F. Luna, and E. Alba, “New ideas in applying scatter search to multiobjective optimization,” in *Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science*, vol. 3410, pp. 443–458, Springer, Berlin, Germany, 2005.
- [60] R. P. Beausoleil, “MOSS multiobjective scatter search applied to non-linear multiple criteria optimization,” *European Journal of Operational Research*, vol. 169, no. 2, pp. 426–449, 2005.
- [61] J. Molina, M. Laguna, R. Martí, and R. Caballero, “SSPMO: a scatter tabu search procedure for non-linear multiobjective optimization,” *Inform Journal on Computing*, vol. 19, no. 1, pp. 91–100, 2007.
- [62] L. V. Santana-Quintero, N. Ramírez, and C. C. Coello, “A multi-objective Particle Swarm optimizer hybridized with scatter search,” *Lecture Notes in Computer Science*, vol. 4293, pp. 294–304, 2006.
- [63] R. P. Beausoleil, “Interactive multiobjective tabu/scatter search based on reference point,” *Revista de Matemática Teoría y Aplicaciones*, vol. 21, no. 2, pp. 261–282, 2014.
- [64] R. P. Beausoleil, “MOSS-II tabu/scatter search for nonlinear multiobjective optimization,” in *Advances in Metaheuristics for Hard Optimization, Natural Computing Series*, P. Siarry and Z. Michalewicz, Eds., pp. 39–67, Springer, Berlin, Germany, 2008.
- [65] R. Baños, C. Gil, J. Reca, and J. Martínez, “Implementation of scatter search for multi-objective optimization: a comparative study,” *Computational Optimization and Applications*, vol. 42, no. 3, pp. 421–441, 2009.
- [66] B. Naderi and R. Ruiz, “A scatter search algorithm for the distributed permutation flowshop scheduling problem,” *European Journal of Operational Research*, vol. 239, no. 2, pp. 323–334, 2014.
- [67] A. R. Rahimi-Vahed, B. Javadi, M. Rabbani, and R. Tavakkoli-Moghaddam, “A multi-objective scatter search for a bi-criteria no-wait flow shop scheduling problem,” *Engineering Optimization*, vol. 40, no. 4, pp. 331–346, 2008.
- [68] E. Nowicki and C. Smutnicki, “Some aspects of scatter search in the flow-shop problem,” *European Journal of Operational Research*, vol. 169, no. 2, pp. 654–666, 2005.
- [69] J.-C. Geng, Z. Cui, and X.-S. Gu, “Scatter search based Particle Swarm optimization algorithm for earliness/tardiness flowshop scheduling with uncertainty,” *International Journal of Automation and Computing*, vol. 13, no. 3, pp. 285–295, 2016.
- [70] R. Hariharan and R. J. G. R. Nimal, “Solving flow shop scheduling problems using a hybrid genetic scatter search algorithm,” *Middle-East Journal of Scientific Research*, vol. 20, no. 3, pp. 328–333, 2014.
- [71] M. A. González, C. R. Vela, and R. Varela, “Scatter search with path relinking for the flexible job shop scheduling problem,” *European Journal of Operational Research*, vol. 245, no. 1, pp. 35–45, 2015.
- [72] M. A. González, C. R. Vela, R. Varela, and I. González-Rodríguez, “An advanced scatter search algorithm for solving job shops with sequence dependent and non-anticipatory setups,” *AI Communications*, vol. 28, no. 2, pp. 179–193, 2015.
- [73] T. Sari, V. Cakir, S. Kilic, and E. Ece, “Evaluation of scatter search and genetic algorithm at resource constrained project scheduling problems,” in *Proceedings of the International Conference on Intelligent Engineering Systems*, Poprad, Slovakia, June 2011.
- [74] R. Alvarez-Valdes, E. Crespo, J. M. Tamarit, and F. Villa, “A scatter search algorithm for project scheduling under partially renewable resources,” *Journal of Heuristics*, vol. 12, no. 1-2, pp. 95–113, 2006.
- [75] G. Hiermann, M. Prandtstetter, A. Rendl, J. Puchinger, and G. R. Raidl, “Metaheuristics for solving a multimodal home-healthcare scheduling problem,” *Central European Journal of Operations Research*, vol. 23, no. 1, pp. 89–113, 2015.
- [76] G. M. Jaradat and M. Ayob, “Scatter search for solving the course timetabling problem,” in *Proceedings of the 3rd Conference on Data Mining and Optimization*, pp. 213–218, Putrajaya, Malaysia, June 2011.
- [77] N. Mansour and G. Sleiman-Haidar, “Parallel scatter search algorithms for exam timetabling,” *Trends in Developing Metaheuristic Algorithms and Optimization Approaches*, IGI Global, Hershey, PA, USA, 2013.
- [78] M. A. Mohammed, M. K. Abd-Ghani, S. A. Mostafa, and D. A. Ibrahim, “Using scatter search algorithm in implementing examination timetabling problem,” *Journal of Engineering and Applied Sciences*, vol. 12, no. 18, pp. 4792–4800, 2017.
- [79] D. Hadjidj and H. Drias, “Scatter search and graph heuristics for the examination timetabling problem,” *The International Arab Journal of Information Technology*, vol. 5, pp. 78–85, 2008.
- [80] N. Mansour, V. Isahakian, and I. Ghalayini, “Scatter search technique for exam timetabling,” *Applied Intelligence*, vol. 34, no. 2, pp. 299–310, 2011.
- [81] R. Martí, H. Lourenço, and M. Laguna, “Assigning proctors to exams with scatter search,” in *Computing Tools for Modeling, Optimization and Simulation*, M. Laguna and J. L. González Velarde, Eds., Kluwer Academic Publishers, Berlin, Germany, pp. 215–227, 2000.
- [82] F. G. López, M. G. Torres, B. Melian, and J. A. Moreno Parez, “Solving feature subset selection problem by a parallel scatter search,” *European Journal of Operational Research*, vol. 169, no. 2, pp. 477–489, 2006.
- [83] J. Wang, A.-R. Hedar, S. Wang, and J. Ma, “Rough set and scatter search metaheuristic based feature selection for credit scoring,” *Expert Systems with Applications*, vol. 39, no. 6, pp. 6123–6128, 2012.
- [84] S. I. Adi, “Parallel scatter search algorithm for feature selection in high dimensional datasets,” Master of Informatics, Palestine Polytechnic University, Hebron, Palestinian, 2017.
- [85] E. Duman and M. H. Ozcelik, “Detecting credit card fraud by genetic algorithm and scatter search,” *Expert Systems with Applications*, vol. 38, no. 10, pp. 13057–13063, 2011.
- [86] N. Boumedine and S. Bouroubi, “Protein structure prediction in the HP model using scatter search algorithm,” *Bulletin du Laboratoire*, vol. 4, pp. 73–85, 2021.
- [87] N. Mansour, C. Kehyayan, and H. Khachfe, “Scatter search algorithm for protein structure prediction,” *International Journal of Bioinformatics Research and Applications*, vol. 5, no. 5, pp. 501–515, 2009.
- [88] E. K. Burke, T. Curtois, R. Qu, and G. V. Berghe, “A scatter search for the nurse rostering problem,” Computer Science Technical report:NOTTCS-TR-2007-7, 2007.
- [89] B. Maenhout and M. Vanhoucke, “New computational results for the nurse scheduling problem: a scatter search algorithm,” in *Lecture Notes in Computer Science*, vol. 3906, pp. 159–170, Springer, Berlin, Germany, 2006.

- [90] J. A. Egea, D. Henriques, T. Cokelaer et al., "MEIGO: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics," *BMC Bioinformatics*, vol. 15, no. 1, p. 136, 2014.
- [91] O. Cerdón, S. Damas, and J. Santamaría, "A fast and accurate approach for 3D image registration using the scatter search evolutionary algorithm," *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1191–1200, 2006.
- [92] O. Cerdón, S. Damas, J. Santamaría, and R. Martí, "Scatter search for the point-matching problem in 3D image registration," *INFORMS Journal on Computing*, vol. 20, no. 1, pp. 55–68, 2008.
- [93] J. Santamaría, O. Cerdón, S. Damas, I. Alemán, and M. Botella, "A scatter search-based technique for pair-wise 3D range image registration in forensic anthropology," *Soft Computing*, vol. 11, no. 9, pp. 819–828, 2007.
- [94] J. Euchí, "Genetic scatter search algorithm to solve the one-commodity pickup and delivery vehicle routing problem," *Journal of Modelling in Management*, vol. 12, no. 1, pp. 2–18, 2017.
- [95] P. c. Belfiore and H. T. Y. Yoshizaki, "Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil," *European Journal of Operational Research*, vol. 199, no. 3, pp. 750–758, 2009.
- [96] J. Tang, J. Zhang, and Z. Pan, "A scatter search algorithm for solving vehicle routing problem with loading cost," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4073–4083, 2010.
- [97] R. A. Russell and W.-C. Chiang, "Scatter search for the vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 169, no. 2, pp. 606–622, 2006.
- [98] T. Zhang, W. A. Chaovaitwongse, and Y. Zhang, "Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries," *Computers & Operations Research*, vol. 39, no. 10, pp. 2277–2290, 2012.
- [99] J. Zhang, J. Tang, and R. Y. K. Fung, "A scatter search for multi-depot vehicle routing problem with weight-related cost," *Asia-Pacific Journal of Operational Research*, vol. 28, no. 3, pp. 323–348, 2011.
- [100] J. Ge, X. Liu, and G. Liang, "Research on vehicle routing problem with soft time windows based on hybrid tabu search and scatter search algorithm," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1945–1958, 2020.
- [101] J. T. Soman and R. J. Patil, "A scatter search method for heterogeneous fleet vehicle routing problem with release dates under lateness dependent tardiness costs," *Expert Systems with Applications*, vol. 150, Article ID 113302, 2020.
- [102] B. B. Keskin and H. Üster, "A scatter search-based heuristic to locate capacitated transshipment points," *Computers & Operations Research*, vol. 34, no. 10, pp. 3112–3125, 2007.
- [103] H. Hakli and Z. Ortacay, "An improved scatter search algorithm for the uncapacitated facility location problem," *Computers & Industrial Engineering*, vol. 135, pp. 855–867, 2019.
- [104] R. Kothari and D. Ghosh, "Scatter search algorithms for the single row facility layout problem," *Advances in Production Engineering and Management*, vol. 3, no. 4, pp. 193–204, 2012.
- [105] J. Xue and Q. Li, "Scatter search algorithm to multiple container loading problem," in *Proceedings of the International Conference on Transportation Engineering*, Chengdu, China, July 2007.
- [106] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, J. L. González-Velarde, and J. Molina, "Multiobjective scatter search for a commercial territory design problem," *Annals of Operations Research*, vol. 199, no. 1, pp. 343–360, 2012.
- [107] F. Yang and L. X. Tang, "A scatter search optimization for ship consolidation plan for strip coils to reduce the number of shuffling," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems*, Shanghai, China, November 2009.
- [108] B. González and B. Adenso-Díaz, "A scatter search approach to the optimum disassembly sequence problem," *Computers & Operations Research*, vol. 33, pp. 1776–1793, 2006.
- [109] A. R. Rahimi-Vahed, M. Rabbani, R. Tavakkoli-Moghaddam, S. A. Torabi, and F. Jolai, "A multi-objective scatter search for a mixed-model assembly line sequencing problem," *Advanced Engineering Informatics*, vol. 21, pp. 85–99, 2007.
- [110] X. Guo and S. Liu, "A scatter search approach for multi-objective selective disassembly sequence problem," *Discrete Dynamics in Nature and Society*, vol. 2014, Article ID 756891, 9 pages, 2014.
- [111] Q. Liu, W.-X. Wang, J.-R. Zhu, C.-Y. Zhang, and Y.-Q. Rao, "Advanced scatter search approach and its application in a sequencing problem of mixed-model assembly lines in a case company," *Engineering Optimization*, vol. 46, no. 11, pp. 1485–1500, 2014.
- [112] J. Cano-Belmán, R. Ríos-Mercado, and J. Bautista, "A scatter search based hyper-heuristic for sequencing a mixed-model assembly line," *Journal of Heuristics*, vol. 16, no. 6, pp. 749–770, 2010.
- [113] H. Rezazadeh, "A rough sets based modified scatter search algorithm for solving 0-1 knapsack problem," *Decision Science Letters*, vol. 4, no. 3, pp. 425–440, 2015.
- [114] S. Hanafi and C. Wilbaut, "Scatter search for the 0–1 multidimensional knapsack problem," *Journal of Mathematical Modelling and Algorithms*, vol. 7, no. 2, pp. 143–159, 2008.
- [115] C. G. da Silva, J. Climaco, and J. Figueira, "A scatter search method for bi-criteria{0, 1}-knapsack problems," *European Journal of Operational Research*, vol. 169, no. 2, pp. 373–391, 2006.
- [116] J. Climaco, C. G. da Silva, and J. Figueira, "On the potentialities of scatter search to approximate the non-dominated solutions set of bicriteria problems-the knapsack case," in *Proceedings of the IISR-IIASA Workshop on Methodologies and Tools for Complex System Modelling and Integrated Policy Assessment*, Laxenburg, Austria, August 2003.
- [117] A. Mizutani, T. Yukawa, K. Numa et al., "Improvement of input-output correlations of electric power load forecasting by scatter search," in *Proceedings of the International Conference on Intelligent Systems Application to Power Systems*, Arlington, VA, USA, November 2005.
- [118] M. D. A. Costa e Silva, C. E. Klein, V. C. Mariani, and L. dos Santos Coelho, "Multiobjective scatter search approach with new combination scheme applied to solve environmental/economic dispatch problem," *Energy*, vol. 53, pp. 14–21, 2013.
- [119] A. F. P. Posada, J. G. Villegas, and J. M. Lopez-Lezama, "A scatter search heuristic for the optimal location, sizing and contract pricing of distributed generation in electric distribution systems," *Energies*, vol. 10, no. 10, p. 1449, 2017.
- [120] J. Blazewicz, F. Glover, and M. Kasprzak, "DNA sequencing-tabu and scatter search combined," *INFORMS Journal on Computing*, vol. 16, no. 3, pp. 232–240, 2004.

- [121] J. A. Nepomuceno, A. Troncoso, and J. S. Aguilar-Ruiz, "Biclustering of gene expression data by correlation-based scatter search," *BioData Mining*, vol. 4, no. 1, 2011.
- [122] J. A. Egea, M. Rodríguez-Fernández, J. R. Banga, and R. Martí, "Scatter search for chemical and bio-process optimization," *Journal of Global Optimization*, vol. 37, pp. 481–503, 2007.
- [123] P. Greistorfer, "A tabu scatter search metaheuristic for the arc routing problem," *Computers & Industrial Engineering*, vol. 44, no. 2, pp. 249–266, 2003.
- [124] Y. Xu and R. Qu, "A hybrid scatter search meta-heuristic for delay-constrained multicast routing problems," *Applied Intelligence*, vol. 36, pp. 229–241, 2012.
- [125] S. Chunxin, Z. Xiaoxia, C. Hongyang, Y. Jiao, and W. Wangpeng, "A hybrid scatter search algorithm for QoS multicast routing problem," in *Proceedings of the Chinese Control and Decision Conference (CCDC)*, pp. 4875–4878, Shenyang, China, June 2018.
- [126] A. M. Alvarez, J. L. González-Velarde, and K. De Alba, "Grasp embedded scatter search for the multicommodity capacitated network design problem," *Journal of Heuristics*, vol. 11, no. 3, pp. 233–257, 2005.
- [127] H. Mori and K. Shimomugi, "Transmission network expansion planning with scatter search," in *Proceedings of the IEEE International Conference in Systems, Man and Cybernetics*, Montreal, QC, Canada, October 2007.
- [128] J. A. M. Perez, B. M. Batista, and M. Laguna, "Scatter search based metaheuristic for robust optimization of the deploying of "DWDM" technology on optical networks with survivability," *Yugoslav Journal of Operations Research*, vol. 15, no. 1, pp. 65–77, 2005.
- [129] A. M. Alvarez, J. L. González-Velarde, and K. De-Alba, "Scatter search for network design problem," *Annals of Operations Research*, vol. 138, no. 1, pp. 159–178, 2005.
- [130] R. Blanco, J. Tuya, and B. Adenso-Diaz, "Automated test data generation using a scatter search approach," *Information and Software Technology*, vol. 51, no. 4, pp. 708–720, 2009.
- [131] R. Sagarna and J. A. Lozano, "Scatter search in software testing, comparison and collaboration with estimation of distribution algorithms," *European Journal of Operational Research*, vol. 169, no. 2, pp. 392–412, 2006.
- [132] R. B. Aguirre, P. J. T. González, M. E. D. Fernandez, and B. A. D. Fernandez, "A scatter search approach for automated branch coverage in software testing," *Engineering Intelligent Systems*, vol. 15, no. 3, pp. 135–142, 2007.
- [133] F. Liu, H. Huang, Z. Yang, Z. Hao, and J. Wang, "Search-based algorithm with scatter search strategy for automated test case generation of NLP toolkit," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 99, 2019.
- [134] A. Bouridah and H. Belhadef, "A scatter search algorithm to configure service function chaining," *International Journal of Internet Technology and Secured Transactions*, vol. 10, no. 6, pp. 654–674, 2020.
- [135] D. S. Yamashita, V. A. Armentano, and M. Laguna, "Scatter search for project scheduling with resource availability cost," *European Journal of Operational Research*, vol. 169, no. 2, pp. 623–637, 2006.
- [136] T. Chinnusamy, P. Kammar, F. Praveen et al., "Scheduling of flexible manufacturing system by hybridizing petri net with improved scatter search algorithm," in *Recent Trends in Mechanical Engineering*, pp. 305–332, Springer, Berlin, Germany, 2020.
- [137] A. G. Krishna and K. M. Rao, "Multi-objective optimization of surface grinding operations using scatter search approach," *International Journal of Advanced Manufacturing Technology*, vol. 29, no. 5, pp. 475–480, 2006.
- [138] P. M. Prabu, K. P. Padmanaban, R. Rajasekar, R. D. Vignesh, and S. Navaethasanthakumar, "Scatter search optimization for multi node machining fixture layout," *International Journal of Engineering and Science*, vol. 3, no. 1, pp. 30–37, 2014.
- [139] J. L. Francois, C. Martin-del-Campo, and L. B. Morales, "Development of a scatter search optimization algorithm for boiling water reactor fuel lattice design," *Nuclear Science and Engineering*, vol. 155, no. 3, pp. 367–377, 2007.
- [140] Y. Tan, M. C. Zhou, Y. Zhang, X. Guo, L. Qi, and Y. Wang, "Hybrid scatter search algorithm for optimal and energy-efficient steelmaking-continuous casting," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1814–1828, 2020.
- [141] P. Greistorfer, "Experimental pool design: input, output and combination strategies for scatter search," in *Metaheuristics: Computer Decision-Making*, pp. 279–300, Springer, Berlin, Germany, 2003.
- [142] F. Glover, A. Løkketangen, and D. L. Woodruff, "Scatter search to generate diverse MIP solutions," in *OR Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, M. Laguna and J. L. González-Velarde, Eds., Kluwer Academic Publishers, Berlin, Germany, pp. 299–317, 2000.
- [143] C.-T. Su, M.-C. Chen, and H.-L. Chan, "Applying neural network and scatter search to optimize parameter design with dynamic characteristics," *Journal of the Operational Research Society*, vol. 56, no. 10, pp. 1132–1140, 2005.
- [144] Q. Le, G. Yang, N. N. W. Hung, X. Zhang, and F. Fan, "A multiobjective scatter search algorithm for fault-tolerant NoC mapping optimization," *International Journal of Electronics*, vol. 101, no. 8, pp. 1056–1073, 2014.
- [145] J. L. González-Velarde, J.-F. Camacho-Vallejo, and G. P. Serrano, "A scatter search algorithm for solving a bilevel optimization model for determining highway tolls," *Computación y Sistemas*, vol. 19, no. 1, 2015.
- [146] Z. Zhang, L. Mao, C. Guan, L. Zhu, and Y. J. S. C. Wang, "An improved scatter search algorithm for the corridor allocation problem considering corridor width," *Soft Computing*, vol. 24, no. 1, pp. 461–481, 2020.
- [147] J. E. Beasley, "Scatter search and bionomic algorithms for the aircraft landing problem," in *Proceedings of the International Conference on Applied Mathematics and Computer Science*, vol. 2, pp. 1–5, Athens, Greece, 2005.
- [148] B. Soykan, *A hybrid tabu/scatter search algorithm for simulation-based optimization of multi-objective runway operations scheduling*, Doctor of philosophy (PhD), dissertation, Engineering Management, Old Dominion University, Norfolk, UK, 2016.
- [149] V. Campos, M. Laguna, and R. Martí, "Scatter search for the linear ordering problem," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds., pp. 331–341, McGraw-Hill, New York, NY, USA, 1999.
- [150] V. Campos, F. Glover, M. Laguna, and R. Martí, "An experimental evaluation of a scatter search for the linear ordering problem," *Journal of Global Optimization*, vol. 21, no. 4, pp. 397–414, 2001.
- [151] J. P. Hamiez and J. K. Hao, "Scatter search for graph coloring," in *Lecture Notes in Computer Science*, vol. 2310, pp. 168–179, Springer, Berlin, Germany, 2002.
- [152] V.-D. Cung, T. Mautor, P. Michelon, and A. Tavares, "A scatter search based approach for the quadratic assignment

- problem,” in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 165–169, Indianapolis, IN, USA, April 1997.
- [153] F. García-López, B. Melián-Batista, J. A. Moreno-Pérez, and J. M. Moreno-Vega, “Parallelization of the scatter search for the p-median problem,” *Parallel Computing*, vol. 29, no. 5, pp. 575–589, 2003.
- [154] V. Campos, M. Laguna, and R. Marti, “Context-independent scatter and tabu search for permutation problems,” *INFORMS Journal on Computing*, vol. 17, no. 1, pp. 111–122, 2005.
- [155] S. Ibri, M. E. A. Cherabrab, and N. Abdoune, “Parallel scatter search approach for the minmax regret location problem,” *International Journal of Applied Metaheuristic Computing*, vol. 9, no. 2, pp. 1–17, 2018.
- [156] D. Habiba and A. Nabila, “Scatter search for SAT and W-MAX-SAT problems,” in *Proceedings of the 33rd Southeastern Symposium on System Theory*, pp. 105–109, Athens, OH, USA, March 2001.
- [157] H. Drias and M. Khabzaoui, “Scatter search with random walk strategy for SAT and MAX-W-SAT problems,” in *Lecture Notes in Computer Science*, vol. 2070, pp. 35–44, Springer, Berlin, Germany, 2001.
- [158] L. Cavique, C. Rego, and I. Themido, *A Scatter Search Algorithm for the Maximum Clique Problem*, Springer, Boston, MA, USA, 2002.
- [159] J. Egea and E. Balsa-Canto, “Dynamic optimization of nonlinear processes with an enhanced scatter search method,” *Industrial & Engineering Chemistry Research*, vol. 48, pp. 4388–4401, 2009.
- [160] M. D. Amico, M. Iori, and S. Martello, “Heuristic algorithms and scatter search for the cardinality constrained  $P||C_{\max}$  problem,” *Journal of Heuristics*, vol. 10, pp. 169–204, 2004.
- [161] J. A. Egea, E. Vazquez, J. R. Banga, and R. Martí, “Improved scatter search for the global optimization of computationally expensive dynamic models,” *Journal of Global Optimization*, vol. 43, pp. 175–190, 2009.
- [162] W. N. N. Hung, X. Song, E. M. Aboulhamid, and M. A. Driscoll, “BDD minimization by scatter search,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 8, pp. 974–979, 2002.
- [163] O. Ibanez, O. Cordon, S. Damas, and J. Santamaria, “An advanced scatter search design for skull-face overlay in craniofacial superimposition,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 1459–1473, 2012.
- [164] M. A. Gutierrez, A. S. de-los-Cobos, and J. Goddard, “A comparison between scatter search and the RAND method for solving the joint replenishment problem,” in *Proceedings of the 5th Mexican International Conference on Artificial Intelligence*, pp. 287–295, Lecture Notes in Artificial Intelligence, Leipzig, Apizaco, Mexico, November 2005.
- [165] M. A. Remil, M. S. Mohamad, S. Deris, R. Sinnott, and S. Napis, “An improved scatter search algorithm for parameter estimation in large-scale kinetic models of biochemical systems,” *Current Proteomics*, vol. 16, no. 5, pp. 427–438, 2019.
- [166] A. Rusdiansyah, D. Musthofa, N. Siswanto, and M. M. Putri, “A hybrid scatter search heuristic algorithm for solving cockpit crew rostering,” *Jurnal Teknik Industri*, vol. 20, no. 2, 2018.
- [167] B. Maenhout and M. Vanhoucke, “A hybrid scatter search heuristic for personalized crew rostering in the airline industry,” *European Journal of Operations Research*, vol. 206, pp. 155–167, 2010.
- [168] F. S. Gharehchopogh and S. Amjad, “A novel hybrid approach for email spam detection based on scatter search algorithm and k-nearest neighbors,” *Journal of Advances in Computer Engineering and Technology*, vol. 5, no. 3, pp. 169–178, 2019.
- [169] J. Wang, A. Hedar, G. H. Zheng, and S. Y. Wang, “Scatter search for rough set attribute reduction,” in *Proceedings of the International Joint Conference on Computational Sciences and Optimization*, Sanya, China, April 2009.