Received: 23 December 2020 | Revised: 30 January 2021 | Accepted: 31 January 2021
DOI: 10.37917/ijeee.17.1.1 Vol. 17| Issue 1| June 2021

Open Access

Iraqi Journal for Electrical and Electronic Engineering
Review Article

Stochastic Local Search Algorithms for Feature Selection: A Review

Hayder Naser Khraibet Al-Behadili
Department of Computer Science, Shatt Alarab University College, Basra, Iraq

Correspondence
* Hayder Naser Khraibet Al-Behadili
Department of Computer Science,
Shatt Alarab University College, Basra, Iraq
Email: hayderkhraibet@sa-uc.edu.iq

Abstract

In today's world, the data generated by many applications are increasing drastically, and finding an optimal subset of features from the data has become a crucial task. The main objective of this review is to analyze and comprehend different stochastic local search algorithms to find an optimal feature subset. Simulated annealing, tabu search, genetic programming, genetic algorithm, particle swarm optimization, artificial bee colony, grey wolf optimization, and bat algorithm, which have been used in feature selection, are discussed. This review also highlights the filter and wrapper approaches for feature selection. Furthermore, this review highlights the main components of stochastic local search algorithms, categorizes these algorithms in accordance with the type, and discusses the promising research directions for such algorithms in future research of feature selection.

KEYWORDS: Feature selection, Stochastic Local Search, Machine learning, Data mining.

I. INTRODUCTION

Classification is a data-mining task used to classify the unknown class for each data accurately [1], [2]. In classification, the data oftentimes have a huge number of attributes (features); many of these attributes are not helpful for data classification. Thus, redundant and irrelevant features in the data could decrease the classification performance. Feature selection is a complex process that automatically selects a subset of features that improve the classification accuracy, shorten the data dimensionality, and decrease the running time [3]. Feature selection has two main approaches: wrapper and filter approaches [4]. The wrapper approaches use a classifier to test the quality during the process of feature selection. The filter approaches do not depend on any classification algorithm and use fitness functions to evaluate a subset of features [5]. However, feature selection is not a trivial task due to the large search space and the interactions among features. Therefore, feature selection approaches suffer from the problems of high computational cost, stagnation, and local optima [6]. Stochastic local search algorithms have been used widely for solving computationally optimization and hard decision problems, including engineering problems, medical diagnosis, chemistry, physics, biology, and computer science [7]–[11]. The stochastic local search algorithms contain a spectrum of methods within the range of simple iterative improvement and constructive procedures to more complex methods to solve the above-mentioned issues [12]. Stochastic local search algorithms have been successfully implemented in various data mining applications[13]–[16], but in feature selection, they have not been completely investigated. Thus, the main objective of this review is to review and comprehend the difference of stochastic local search algorithms for feature selection to select a minimum number of features and obtain similar or better classification accuracy than that when using all features in the data. This review investigates the differences between wrapper and filter approaches for multi-objective and single-objective feature selection. Lastly, this review focuses on the considered directions in enhancing such algorithms in feature selection approaches, developing applications, and establishing potential systems for next-generation data classification.

The rest of this paper is structured as follows. Section 3 provides background on filter and wrapper approaches. The previously introduced stochastic local search algorithms for feature selection were extensively reviewed in Section 4.". Then, discussion and future research directions are indicated in Section 5. Lastly, the review concludes with general remarks in Section 6.

II. Feature Selection

This section displays conventional feature selection approaches, which are the filter and wrapper approaches.

A. Filter methods

The filter approaches aim at finding the optimal number of features in a dataset on the basis of evaluation functions. The evaluation functions are independent of any classification algorithm. In the literature, various criteria, including information measures [17], distance measures [18], consistency measures [19], dependency measures [20], and confidence and coverage measures [21], have been produced. These criteria are used to develop filter feature selection algorithms. The filter algorithms are computationally less expensive than others [22] but lead to poor classification performance because they ignore the interaction between the selected subset of features and the classification algorithm during the process of feature selection [23].

B. Wrapper methods

In the wrapper type of feature selection, the algorithm occurs as a wrap around the classifier and uses it as a “black box” via the feature selection algorithm to evaluate the quality of the selected features and guide the search process [24]. Wrapper feature selection approaches are generally more expensive than filter approaches because they include training and test processes. Therefore, most of the existing wrappers in the literature use greedy or stochastic search strategies [25].

Sequential backward selection and sequential forward selection are two popular algorithms used in wrapper feature selection approaches. Both selection methods search for the optimal feature subset by using a greedy hill-climbing search strategy [26]. The sequential backward selection removes one feature from the feature subset until the further deletion of any data attribute will not improve the classification accuracy. By contrast, sequential forward selection sequentially adds a feature to an empty set of features until any addition has no further improvement in classification accuracy. However, both algorithms suffer from the problem of nesting effect, i.e., when a feature is deleted (selected), it cannot be deleted (selected) later. Hence, these methods are easily trapped in local optima and require long computation, especially with high-dimensional data [27]. Recently, stochastic local search algorithms for feature selection have been utilized to wrapper and filter feature selection models. The next section will elucidate these algorithms.

III. Stochastic Local Search Algorithms for Feature Selection

A. Simple stochastic local search algorithms

The simple stochastic local search algorithms have iterative improvement concepts to enhance the neighbourhood, produce new candidate solutions, and escape optimal local problems. Examples of simple stochastic local search algorithms are simulated annealing (SA) and tabu search (TS).

SA is a simple stochastic local search algorithm proposed by Kirkpatrick et al. (1983) [28]. It is based on the slow cooling of metals. It is an adaptation for optimization methods for approximating the global optimum for large search space in the given problem. The objective function is based on the energy state variation. This variation leads to a new potential solution (neighbor of the current solution) by altering the current solution on the basis of predefined rules. In the SA algorithm, the optimization mimics allows the acceptance of a less-quality solution in accordance with the temperature T value and two algorithmic components (i.e., acceptance criterion and cooling schedule) [9], [29]. SA initially sets T to a high temperature and then gradually decreases it. Thus, SA can accept worsening candidate solutions. As the temperature value is reduced, the search process of SA becomes increasingly greedy and only allows improving solutions or solutions with a fitness function value equal to the current candidate solution. The SA-based feature selection uses its algorithmic power to find the best features from the given problem. The pseudocode of SA-based feature selection is shown in Fig. 1.

SA-based feature selection

Input Original dataset
Output The selected features of dataset
Generate empty set of Attributes {};
$S \longleftarrow$ CreateInitialAttributesSet {};
$T \longleftarrow T_0$. // cooling schedule
While Acceptance criterion not met **do**
  $S' \longleftarrow$ ChangeNeighborhood ($N(S)$);
  If quality (S') > (S) ***Then***
    $S \longleftarrow S'$ *//replacement*
else
  Accept S' as new subset of features with probability
End if
Update (T) // temperature value is reduced
End While

Fig. 1: Pseudocode of SA-based feature selection.

Research on depression has evaluated the performance of the SA-based feature selection algorithm. The algorithm can find the features that could detect depression disease [30]. Another paper on SA-based feature selection is proposed on large commercial databases. In this article, an entropic measure is developed to select the high-quality subsets of features from the databases [31].

Another simple stochastic local search algorithm is TS. It is a metaheuristic algorithm used for local search optimization to improve a solution iteratively to another potential solution in its neighborhood [32]. In the classical feature selection problem, the TS concepts utilize the prohibiting principle for the already visited region or by other predetermined rules (see Fig.2). The basic TS is composed of tabu lists and an aspiration criterion [33]. Tabu lists tend to be a short/medium memory in the algorithm. The tabu lists disable movements to the previously selected features, referred to as tabu moves. The aspiration criterion could be a number of iterations (i.e., period) for selecting the remaining features in the tabu lists [34], [35].

TS-based feature selection

Input Original dataset
Output The selected features of dataset
Generate empty set of Attributes {};
$S \longleftarrow$ CreateInitialAttributesSet {};
TabuList = { }
While Aspiration criterion not met **do**
$S' \longleftarrow$ SelectBestFeature ($N(S)$/ *TabuList*);
  Update *(TabuList);*
End While

Fig. 2: Pseudocode of TS-based feature selection.

B. Evolutionary-based local search algorithms

Another type of stochastic local search algorithms is evolutionary algorithms (EAs), such as genetic programming (GP) and genetic algorithm (GA) [36]. EAs use the principle of evolution of Charles Darwin, namely, the survival of the fittest, which leads to natural selection and an increase in the quality of the population [37], [38]. In GP, an evolutionary learning algorithm, the system extracts combinations of attributes with potential predictive power and comprehensibility. An individual in the population is represented as a tree. Each individual undergoes GP. Then, the individual is evaluated using a fitness function with respect to its accuracy to solve the target problem. The size of the new generation may exceed that of the parent; hence, an evaluation function that considers the size of the discovered features should be incorporated. The first component of PG is a reproduction operator that is responsible for selecting one individual in accordance with its evaluation value. An individual with a high value will participate in the next generation of individuals. After the selection, the individual will be going into the new generation on the basis of the principles of natural selection and survival of the fittest. The second component is a crossover operator that selects two individuals from the population and marriages them to introduce two new individuals. The crossover process comprises several ways, but the common one is single-point crossover. This operator selects random features from two individuals, called parents, and then swaps the features between them to produce new individuals. The third component is a mutation operator to maintain the diversity of the solutions and help guide the search in all solution spaces. The mutation operator selects a random node (internal node) from the solutions, removes it with its children, and replaces it with the randomly generated solutions [39], [40]. The pseudocode of GP-based feature selection algorithm is shown in Fig 3.

GP-based feature selection

Input Original dataset
Output The selected features of dataset
$\boldsymbol{P} \longleftarrow$ GenerateInitialPopulation ();
Evaluation ($\boldsymbol{P}$)
While Aspiration criterion not met **do**
  $\boldsymbol{S'} \longleftarrow$ SelectBestFeature set
  $\boldsymbol{P'} \longleftarrow$ Crossover (S');
  $\boldsymbol{P''} \longleftarrow$ Mutation (S');
Evaluation ($\boldsymbol{P''}$);
$\boldsymbol{P} \longleftarrow$ Select ($\boldsymbol{P''} \cup \boldsymbol{P}$);
End While

Fig. 3: Pseudocode of GP-based feature selection.

In GA, the feature selection is conducted using a chromosome to encode a set of selected features [41]. As shown in Fig. 4, the first component of GA in feature selection is an encoding component, in which the data have to be encoded in the chromosome or individual. Then, the chromosome or individual will represent a candidate set of features.

The second component of GA is a crossover operator to exchange random pieces of two individuals. The classical crossover operation is implemented by selecting two chromosomes and random attributes from them and exchanging them. The third component of GA is a mutation operator, which aims at avoiding the local optimum problem. The mutation applies for a single chromosome at a time. It randomly removes an attribute with other value belonging to the domain of that attribute. The mutation does not always produce an enhanced result, but it is an important step in global optimization [42], [43].

GA-based feature selection

Input Original dataset
Output The selected features of dataset
$\boldsymbol{P} \longleftarrow$ GenerateInitialPopulation ();
Evaluation ($\boldsymbol{P}$)
While termination conditions not met **do**
  SelectBestFeature set ();
  Crossover ();
  Mutation ();
  Update **($\boldsymbol{P}$);**
End While

Fig. 4: Pseudocode of GA-based feature selection algorithm.

C. Swarm-based local search algorithms

Swarm intelligence refers to biologically inspired algorithms that demonstrate their potential power to solve different real-world applications. It uses the collective behavior of self-organized systems. Examples of swarm intelligence algorithms are particle swarm optimization (PSO) [44], ant colony optimization [45], artificial bee colony (ABC) [46], grey wolf optimization (GWO) [47], and bat algorithm (BA) [48].

PSO is a metaheuristic swarm optimization algorithm that searches for an optimal solution using a swarm of particles that is updated from iteration to iteration [49]. It finds the best features from data by using a population of candidate solutions (subsets of features) and dynamically moves around the search space on the basis of a mathematical equation over the position and velocity of the particles. Thus, PSO keeps tending in the direction of its previously best (pbest) set of features and the global best (gbest) features in the swarm. This process is expected to move the particles to find the global best subset of features [50], [51]. The pseudocode of the PSO-based feature selection is shown in Fig. 5 below.

PSO-based feature selection

Input Original dataset
Output The selected features of dataset
P ⟵ GenerateInitialPopulation ();
repeat
Evaluation (***P***)
For each particle *i* in ***P* do**
    Apply LocalSearch ();
    Update the particle best features ();
    Update the global best features ();
End For
    Update **(*P*);**
Until max iteration

Fig. 5: Pseudocode of PSO-based feature selection algorithm.

ABC is one of the most recent optimization algorithms proposed by Dervis Karaboga in 2005; it is based on the intelligent foraging behavior of honeybees. ABC is a swarm-based algorithm that provides good search capabilities in various optimization problems and feature selection [52]. As shown in Fig. 6, the ABC uses three groups of bees (i.e., search procedure): employed, onlooker, and scout bees. The employed bees find a set of solutions (e.g., a subset of features). The solutions are evaluated and high nectar amounts are detected. The employed bee who is stagnated in a single solution becomes a scout and randomly finds a new subset of features. Onlookers watch the dances of employed bees and improve the solutions further. The ABC algorithm combines two search methods. First, local search is used by employed and onlooker bees. Second, the global search method is controlled by onlookers and scouts. In this way, ABC balances between the exploitation and exploration processes to find the best feature subset from the data [53], [54].

ABC-based feature selection

Input Original dataset
Output The selected features of dataset
P ⟵ GenerateInitialPopulation ();
Evaluation (***P***)
repeat
    Apply Employed bees ();
    Apply Onlooker bees ();
    Apply Scout bees ();
    Update the Best set of features ();
    Update **(*P*);**
Until max iteration

Fig. 6: Pseudocode of ABC-based feature selection algorithm.

In feature selection and classification, BA is a probabilistic technique for solving computational problems and finding the best features from data [55], [56]. BA is a metaheuristic algorithm that uses the echolocation behavior of microbats to find solutions to different combinatorial optimization problems. It is based on the foraging behavior and the echolocation principle of bats and has an intelligent ability to remember the past solutions and knowledge about the distance of other regions in the search space. It generates a population of solutions on the basis of velocity and frequency [57]. Then, it selects the best one among the generated solutions. The selected solution undergoes a local search stage for further improvement by creating a local solution around the best solution. If the solution is improved (feature subset), then the new solution will replace the old one. The algorithm repeats this task a certain amount of times and ranks the current best bat [58], [59]. The pseudocode of the BA-based feature selection is shown in Fig.7 below.

BA-based feature selection

Input Original dataset
Output The selected features of dataset
P ⟵ GenerateInitialPopulation ();
Evaluation (***P***)
repeat
    Generate feature subset ();
    Update velocity and position ();
    Update the Best set of features ();
    Apply Local Search method ();
    Update **(*P*);**
Until max iteration

Fig. 7: Pseudocode of BA-based feature selection algorithm.

GWO is a nature inspired swarm-based metaheuristic optimization algorithm. It is inspired by the hunting and leadership attitude of grey type wolves [60]. It utilizes the three main steps of hunting: searching for prey, encircling prey, and attacking prey (see Fig. 8). It simulates the leadership hierarchy by using three types of wolves: alpha, beta, and delta. In feature selection, GWO first explores the search space for a prey (set of features) in accordance with the position in the search space. Hunting is then performed by recognizing the location of the best features and encircling them. The hunt is usually guided by the best wolf position, called alpha. The beta and delta (the second and third-best feature subsets, respectively) can participate in the hunting occasionally. The GWO algorithm enables its search agents to revise their position on the basis of the location of the alpha, beta, and delta to become close to the prey (exploitation) [61], [62]. Furthermore, the GWO algorithm has been implemented in different FS domains: such as disease diagnosis, anomaly detection, and gene selection.

GWO-based feature selection

```
Input     Original dataset
Output    The selected features of dataset
          P ◄── GenerateInitialPopulation ();
          Initailaze the parameters ();
          Evaluation the quality of each wolf ();
         // Searching for prey
                  //Encircling prey start here
          Select the best wolf ();
          Select the second-best wolf ();
          Select the third-best wolf ();
                  //Encircling prey end here
          While termination criteria not met do
             For each wolf i in P do
                  Apply LocalSearch (); // Attacking prey
             End
             Update the parameters ();
             Update the (first, second and third) best wolf ();
          End While
```

Fig. 8: Pseudocode of GWO-based feature selection algorithm.

Table 1 summarizes the stochastic local search algorithms, basic algorithmic components, local search types, and application domain used in feature selection. Besides, a taxonomy is proposed and presented in Fig. 9 below based on the number of perspectives and views in the literature.

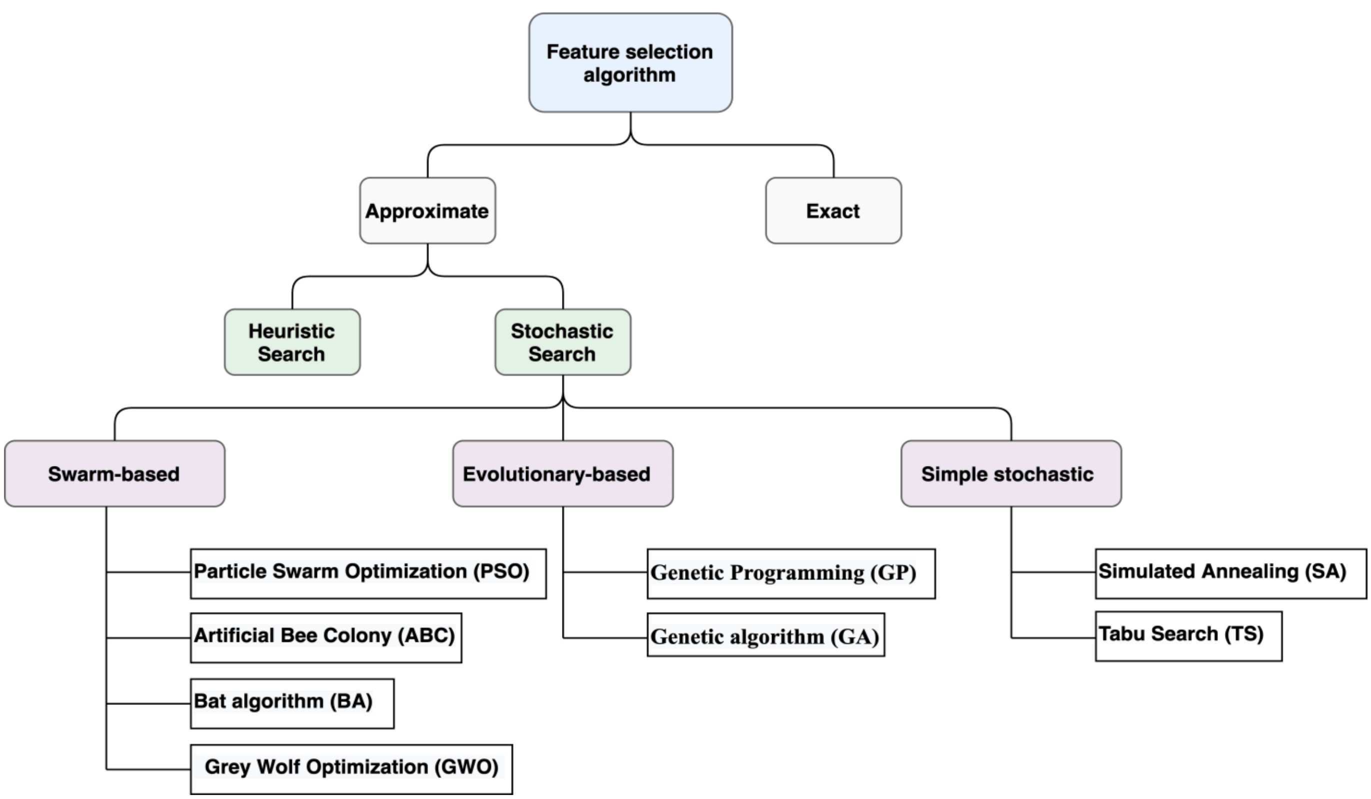

Fig. 9: Proposed taxonomy of feature selection algorithms.

TABLE I
CURRENT APPLICATIONS OF STOCHASTIC LOCAL SEARCH ALGORITHMS IN FEATURE SELECTION LISTED IN ACCORDANCE WITH TYPES

No.	Reference	Algorithm	Algorithm components	Application domain	Local search type
1	Ref [30] 2019	SA	Acceptance criterion Cooling schedule	Depression dataset	Simple stochastic local search
2	Ref [31] 1997	SA	Acceptance criterion Cooling schedule	Large commercial databases	Simple stochastic local search
3	Ref [33] 2020	TS	Tabu lists, and Aspiration criterion	Breast cancer	Simple stochastic local search
4	Ref [34] 2009	TS	Tabu lists, and Aspiration criterion	UCI datasets	Simple stochastic local search
5	Ref [35] 2002	TS	Tabu lists, and Aspiration criterion	High-dimensional data	Simple stochastic local search
6	Ref [39] 2016	GP	Evaluation Selection Crossover Mutation	High-dimensional data	EAs
7	Ref [40] 2019	GP	Evaluation Selection Crossover Mutation	Skin cancer	EAs
8	Ref [42] 2018	GA	Encoding Crossover Mutation	Credit risk analysis	EAs
9	Ref [43] 2014	GA	Encoding Crossover Mutation	Flavia image dataset	EAs
10	Ref [50] 2018	PSO	Initialization Evaluation Local search Selection (global subset)	Document clustering	Swarm intelligence
11	Ref [51] 2018	PSO	Initialization Evaluation Local search Selection (global subset)	Logistic regression datasets	Swarm intelligence
12	Ref [53] 2020	ABC	Employed bees Onlooker bees Scout bees	Grape leaf disease	Swarm intelligence
13	Ref [54] 2015	ABC	Employed bees Onlooker bees Scout bees	10 benchmark datasets	Swarm intelligence
14	Ref [57] 2017	BA	Generate solutions Update velocity and position Select the best solution Local search	Microarray cancer data	Swarm intelligence
15	Ref [58] 2012	BA	Generate solutions Update velocity and position Select the best solution Local search	UCI datasets	Swarm intelligence
16	Ref [59] 2017	BA	Generate solutions Update velocity and position Select the best solution Local search	Cancer classification	Swarm intelligence
17	Ref [61] 2016	GWO	Searching, Encircling, and Attacking (prey)	UCI datasets	Swarm intelligence
18	Ref [62] 2015	GWO	Searching, Encircling, and Attacking (prey)	UCI datasets	Swarm intelligence

In this review, primary studies of stochastic local search algorithms in feature selection were provided. It was observed that the algorithms have the advantages of selecting the optimal subsets of the attribute from the original dataset. This type of algorithms enables to simplify the dataset and produce a comprehensible subset, increase generalization and enhance the classification accuracy. However, the main drawback of these algorithms is the computations cost required to obtain the features. In each subset selection, the classification model will be trained and tested for each subset to achieve classification accuracy. Thus, if the dataset consists of a wide number of features, most of the algorithm execution time is spent in training and testing phases. Another drawback is that each subset of feature (i.e., the gene in evolutionary based algorithms or agents in swarm-based algorithms) could be evaluated multiple times since the classification model qualities for evaluation subsets of features are not indicated for future retrieval. Another disadvantage is that using the classification algorithm performance as the fitness function could lead to an overfitting problem. Furthermore, Fig. 10 demonstrates the distribution of publications published on stochastic local search algorithms in feature selection (as indexed by google scholar) which indicates a gradual increase since early 2010. The articles count clearly indicate that evolutionary-based feature selection seems most common. Although the swarm-based feature selection has become a promising area of research, simple models are showing growth in popularity as well.

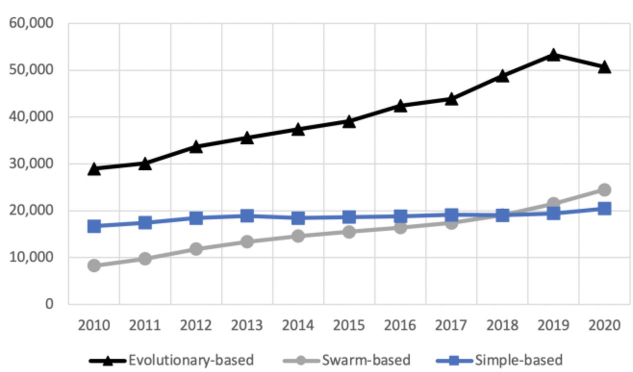

Fig. 10: Number of articles written on stochastic local search algorithms in feature selection, from google scholar.

IV. Discussion and Future Research

This section discusses four research directions that are particularly relevant and promising: multi-objective local search, hybridization of different stochastic local search algorithms, online configuration, and type of attributes.

The existing feature selection approaches are implemented to find the smallest number of features with better or similar classification performance than that when using all features in the data. However, the main purpose of feature selection approaches is to intensify and increase the classification accuracy. A subset of features with a low degree of repetition and the same or even higher classification accuracy can be achieved with a smaller number of features [63], [64]. Developing feature selection-based wrapper or filter approaches accordingly becomes necessary to optimize the two main objectives: the number of features and classification accuracy.

The stochastic local search algorithms are high-level strategies that have the ability to explore a dataset and find a high-quality subset of features. These algorithms can be classified into two groups: constructive or iterative [43], [65]. Constructive local search algorithms build an optimal subset of features from scratch. They add the best available features to the solution. The iterative algorithms find optimal features by iteratively replacing the current features with their neighbors, while the classification accuracy is improved. These algorithms use stochastic moves, memory in the search process, and accepted solutions worse than previous ones. Thus, hybridization between the constructive and iterative local search algorithms is required to find the optimal subset of features from a given dataset. The constructive algorithms are used to explore the search space and produce the initial feature subset, while the iterative algorithms utilize this subset to exploit the neighborhood space and improve it. Therefore, using those capabilities will allow exploring large search spaces without becoming trapped in local minima.

The performance of stochastic local search algorithms depends strongly on the appropriate parameter setting. In manual cases, different values of user-specifiable parameters that influence the search behavior exist. With different categorical and numerical parameters, many techniques and strategies are proposed. In categorical parameter situations, such as neighborhoods or mutation mechanisms, a choice needs to be taken from a discrete set of design variants to improve solution quality or computation time. In the numerical parameter case, the search behavior of the algorithms, such as the tabu list in the TS algorithm and the cooling schedule in the SA algorithm, is controlled. The parameter configuration techniques are a promising research area and have improved the performance of algorithms in many application domains and different problems. In the literature, two configuration methods, offline and online, are available. The offline configuration is used to determine the parameter settings during a training phase before algorithm deployment. In this method, the preselected parameter value does not consider the different stages of search space and does not benefit from feedback from search behavior. By contrast, the online configuration modifies the parameter values, while the algorithm searches for an optimal solution. Nevertheless, the parameter configuration in feature selection is still an undiscovered research area. Hence, a successful method used in the area of EAs [66] and reactive search [67] could be adopted in feature selection algorithms. Another research direction is to adopt undiscovered stochastic local search algorithms (e.g., iterated greedy and local search) algorithms in the feature selection domain [12].

The features in the data consist of categorical and continuous types. The difference between the two types is by the number of values they can take. The categorical features have a finite number of particular values while the continuous features have infinite possibilities of the number values, for example, temperature or weight [58], [68]. The traditional stochastic local search algorithms failed to cope with both types in the feature selection process. Therefore, an interesting research direction is to adapt these algorithms to deal with both types.

V. Conclusions

Stochastic local search is a relatively new domain within feature selection research. It has drawn increasing attention from the research community, with various applications. The proposed feature selection algorithms that incorporate these stochastic local search principles often show performance results that outperform those of the traditional approaches. Yet, many opportunities and trends exist. While the stochastic local search studies are still in the early stage, the research on the application thereof is continuously in motion. In this review, the existing algorithms are categorized in three approaches: simple, evolutionary, and swarm. Furthermore, this review provides future directions for researchers in comparing existing or producing new stochastic local search-based feature selection methods.

Conflict of Interest

The authors have no conflict of relevant interest to this article.

References

[1] T. R. Patil and S. Sherekar, “Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification,” *Int. J. Comput. Sci. Appl.*, vol. 6, no. 2, pp. 256–261, 2013.

[2] H. N. K. Al-behadili, “Classification Algorithms for Determining Handwritten Digit,” *Iraqi Journal for Electrical and Electronic Engineering*, vol. 12, no. 1, pp. 96–102, 2016.

[3] N. K. Verma and A. Salour, “Feature selection,” in *Studies in Systems, Decision and Control*, 2020.

[4] M. M. Kabir, M. Shahjahan, and K. Murase, “A new hybrid ant colony optimization algorithm for feature selection,” *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3747–3763, 2012.

[5] Y. B. W. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, “Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy,” *Pertanika J. Sci. Technol.*, vol. 26, no. 1, pp. 329–340, 2018.

[6] J. Li and H. Liu, “Challenges of Feature Selection for Big Data Analytics,” *IEEE Intell. Syst.*, vol. 32, no. 2, pp. 9–15, 2017.

[7] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, “Hybrid Ant Colony Optimization and Iterated Local Search for Rules-Based Classification,” *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 04, pp. 657–671, 2020.

[8] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, “Modified ACS Centroid Memory for Data Clustering,” *J. Comput. Sci.*, vol. 15, no. 10, pp. 1439–1449, 2019.

[9] A. Franzin and Thomas Stutzle, “Revisiting Simulated Annealing: a Component-Based Analysis,” Bruxelles, Belgium, 2018.

[10] H. H. Hoos and T. Stutzle, “Stochastic Local Search Algorithms: An Overview,” in *Springer Handbook of Computational Intelligence*, Springer Berlin Heidelberg, 2015, pp. 1085–1105.

[11] A. Albaghdadi and A. Ali, “An Optimized Complementary Filter For An Inertial Measurement Unit Contain MPU6050 Sensor,” *Iraqi Journal for Electrical and Electronic Engineering*, vol. 15, no. 2, pp. 71–77, 2019.

[12] T. Stützle and R. Ruiz, “Iterated Local Search,” in *Handbook of Heuristics*, Springer International Publishing, 2017, pp. 1–27.

[13] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, “Ant colony optimization algorithm for rule-based classification: Issues and potential solutions,” *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 21, pp. 7139–7150, 2018.

[14] H. N. K. Al-Behadili, R. Sagban, and K. R. Ku-Mahamud, “Adaptive Parameter Control Strategy for Ant-Miner Classification Algorithm,” *Indones. J. Electr. Eng. Informatics*, vol. 8, no. 1, pp. 149–162, 2020.

[15] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, “An improved ACS algorithm for data clustering,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1506–1515, 2020.

[16] A. M. Jabbar and K. Ku-Mahamud, “Ant-based sorting and ACO-based clustering approaches: A review,” in *In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018, pp. 217–223.

[17] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, “Normalized mutual information feature selection,” *IEEE Trans. Neural Networks*, vol. 20, no. 2, pp. 189–201, 2009.

[18] N. Kwak and C. H. Choi, “Input feature selection for classification problems,” *IEEE Trans. Neural Networks*, vol. 13, no. 1, pp. 143–159, 2002.

[19] K. Shin, D. Fernandes, and S. Miyazaki, “Consistency measures for feature selection: A formal definition, relative sensitivity comparison and a fast algorithm,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2011, pp. 1491–1497.

[20] L. Yu and H. Liu, “Efficient feature selection via analysis of relevance and redundancy,” *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, 2004.

[21] D. Martens, M. Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, “Classification with ant colony optimization,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 651–665, 2007.

[22] C. S. Yang, L. Y. Chuang, J. C. Li, and C. H. Yang, “Chaotic maps in binary particle swarm optimization for feature selection,” *SMCia/08 - Proc. 2008 IEEE Conf. Soft Comput. Ind. Appl.*, pp. 107–112, 2008.

[23] B. Venkatesh and J. Anuradha, “A review of feature selection and its methods,” *Cybern. Inf. Technol.*, vol. 19, no. 1, pp. 3–26, 2019.

[24] Y. Wah, N. Ibrahim, H. Hamid, S. Abdul-Rahman, and S. Fong, “Feature selection methods: case of filter and wrapper approaches for maximising classification accuracy,” *Pertanika J. Sci. Technol.*, vol. 26, no. 1, pp. 329–340, 2018.
[25] I. A. Gheyas and L. S. Smith, “Feature subset selection in large dimensionality domains,” *Pattern Recognit.*, vol. 43, no. 1, pp. 5–13, 2010.
[26] A. A. Abdoos, P. K. Mianaei, and M. R. Ghadikolaei, “Combined VMD-SVM based feature selection method for classification of power quality events,” *Appl. Soft Comput. J.*, vol. 38, pp. 637–646, 2016.
[27] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997.
[28] B. Chopard and M. Tomassini, “Simulated annealing,” in *Natural Computing Series*, 2018.
[29] S. Ledesma, G. Avia, and R. Sanchez, “Practical Considerations for Simulated Annealing Implementation,” in *Simulated Annealing*, 2008.
[30] M. Arokkiya and Jabasheela, “Performance Evaluation of Simulated Annealing based Feature Selection Approach on Depression Dataset,” *J. Appl. Sci. Comput.*, vol. VI, no. I, pp. 3061–3068, 2019.
[31] J. C. W. Debuse and V. J. Rayward-Smith, “Feature Subset Selection within a Simulated Annealing Data Mining Algorithm,” *J. Intell. Inf. Syst.*, vol. 9, no. 1, pp. 57–81, 1997.
[32] M. Laguna, “Tabu search,” in *Handbook of Heuristics*, 2018.
[33] H. Dhahri, I. Rahmany, A. Mahmood, E. Al Maghayreh, and W. Elkilani, “Tabu Search and Machine-Learning Classification of Benign and Malignant Proliferative Breast Lesions,” *Biomed Res. Int.*, vol. 2020, pp. 1–10, 2020.
[34] Y. Wang, L. Li, J. Ni, and S. Huang, “Feature selection using tabu search with long-term memories and probabilistic neural networks,” *Pattern Recognit. Lett.*, vol. 30, no. 7, pp. 661–670, 2009.
[35] H. Zhang and G. Sun, “Feature selection using tabu search method,” *Pattern Recognit.*, vol. 35, no. 3, pp. 701–711, 2002.
[36] R. S. Parpinelli, G. F. Plichoski, R. S. Da Silva, and P. H. Narloch, “A review of techniques for online control of parameters in swarm intelligence and evolutionary computation algorithms,” *Int. J. Bio-Inspired Comput.*, vol. 13, no. 1, pp. 1–20, 2019.
[37] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, “A survey of multiobjective evolutionary algorithms for data mining: Part I,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 4–19, 2014.
[38] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, “Parameter Control in Evolutionary Algorithms: Trends and Challenges,” *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 167–187, 2015.
[39] B. Tran, B. Xue, and M. Zhang, “Genetic programming for feature construction and selection in classification on high-dimensional data,” *Memetic Comput.*, vol. 8, no. 1, pp. 3–15, 2016.
[40] Q. U. Ain, B. Xue, H. Al-Sahaf, and M. Zhang, “Genetic Programming for Multiple Feature Construction in Skin Cancer Image Classification,” in *International Conference Image and Vision Computing New Zealand*, 2019, vol. 2019-Decem, pp. 732–745.
[41] A. Bakirtzis and S. Kazarlis, “Genetic algorithms,” in *Advanced solutions in power systems: HVDC, facts, and artificial intelligence*, 2016, pp. 845–902.
[42] S. Jadhav, H. He, and K. Jenkins, “Information gain directed genetic algorithm wrapper feature selection for credit rating,” *Appl. Soft Comput. J.*, vol. 69, no. August, pp. 541–553, 2018.
[43] O. Babatunde, L. Armstrong, J. Leng, and D. Diepeveen, “A Genetic Algorithm-Based Feature Selection,” *Int. J. Electron. Commun. Comput. Eng.*, vol. 5, no. 4, pp. 899–905, 2014.
[44] F. Ebadifard and S. M. Babamir, “A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment,” *Concurr. Comput.* , vol. 30, no. 12, pp. 1–16, 2018.
[45] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, “Rule pruning techniques in the ant-miner classification algorithm and its variants: A review,” in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018, pp. 78–84.
[46] Y. Kabalci, S. Kockanat, and E. Kabalci, “A modified ABC algorithm approach for power system harmonic estimation problems,” *Electr. Power Syst. Res.*, vol. 154, no. January, pp. 160–173, 2018.
[47] Y. Ren, T. Ye, M. Huang, and S. Feng, “Gray Wolf Optimization algorithm for multi-constraints second-order stochastic dominance portfolio optimization,” *Algorithms*, vol. 11, no. 5, 2018.
[48] T. M. Adis ALIHODZIC, “Bat Algorithm (BA) for Image Thresholding,” in *Recent Researches in Telecommunications, Informatics, Electronics and Signal Processing*, 2015, no. March 2015, pp. 364–369.
[49] D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm: an overview,” *Soft Comput.*, vol. 22, no. 2, pp. 387–408, 2018.
[50] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, “A new feature selection method to improve the document clustering using particle swarm optimization algorithm,” *J. Comput. Sci.*, vol. 25, no. September, pp. 456–466, 2018.
[51] O. S. Qasim and Z. Y. Algamal, “Feature selection using particle swarm optimization-based logistic regression model,” *Chemom. Intell. Lab. Syst.*, vol. 182, pp. 41–46, 2018.
[52] S. Aslan, “A transition control mechanism for artificial bee colony (ABC) algorithm,” *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–24, 2019.
[53] A. D. Andrushia and A. T. Patricia, “Artificial bee colony optimization (ABC) for grape leaves disease detection,” *Evol. Syst.*, vol. 11, no. 1, pp. 105–117, 2020.
[54] E. Hancer, B. Xue, D. Karaboga, and M. Zhang, “A binary ABC algorithm based on advanced similarity scheme for feature selection,” *Appl. Soft Comput. J.*, vol. 36, pp. 334–348, 2015.
[55] X. S. Yang and X. He, “Bat algorithm: literature review and applications,” *Int. J. Bio-Inspired Comput.*, vol. 5,

no. 3, p. 141, 2013.
[56] I. Gagnon, A. April, and A. Abran, "A critical analysis of the bat algorithm," *Eng. Reports*, vol. 2, no. 8, pp. 1–20, 2020.
[57] O. Ahmad Alomari, A. Tajudin Khader, M. Azmi Al-Betar, and L. Mohammad Abualigah, "Gene selection for cancer classification by combining minimum redundancy maximum relevancy and bat-inspired algorithm," *Int. J. Data Min. Bioinform.*, vol. 19, no. 1, pp. 32–51, 2017.
[58] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X. S. Yang, "BBA: A binary bat algorithm for feature selection," *Brazilian Symp. Comput. Graph. Image Process.*, pp. 291–297, 2012.
[59] O. A. Alomari, A. T. Khader, M. A. Al-Betar, and L. M. Abualigah, "MRMR BA: A hybrid gene selection algorithm for cancer classification," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 12, pp. 2610–2618, 2017.
[60] S. Gupta and K. Deep, "A novel Random Walk Grey Wolf Optimizer," *Swarm Evol. Comput.*, vol. 44, pp. 101–112, 2019.
[61] W. Yamany, E. Emary, and A. E. Hassanien, "New rough set attribute reduction algorithm based on grey wolf optimization," *Adv. Intell. Syst. Comput.*, vol. 407, no. November, pp. 241–251, 2016.
[62] E. Emary, W. Yamany, A. E. Hassanien, and V. Snasel, "Multi-Objective Gray-Wolf Optimization for Attribute Reduction," *Procedia Comput. Sci.*, vol. 65, pp. 623–632, 2015.
[63] Y. Fang, Z. H. Liu, and F. Min, "A PSO algorithm for multi-objective cost-sensitive attribute reduction on numeric data with error ranges," *Soft Comput.*, vol. 21, no. 23, pp. 7173–7189, 2017.
[64] W. Li, X. Jia, L. Wang, and B. Zhou, "Multi-objective attribute reduction in three-way decision-theoretic rough set model," *Int. J. Approx. Reason.*, vol. 105, pp. 327–341, 2019.
[65] G. Wang *et al.*, "Multiple parameter control for ant colony optimization applied to feature selection problem," *Neural Comput. Appl.*, vol. 26, no. 7, pp. 1693–1708, 2015.
[66] A. E. Eiben, Z. Michalewicz, M. Schoenauer, and J. E. Smith, "Parameter control in evolutionary algorithms," *Stud. Comput. Intell.*, vol. 54, no. 2, pp. 19–46, 2007.
[67] T. Stutzle *et al.*, "Parameter Adaptation in Ant Colony Optimization," in *Autonomous search*, 2012, pp. 191–215.
[68] R. Kamala and R. Thangaiah, "An improved hybrid feature selection method for huge dimensional datasets," *IAES Int. J. Artif. Intell.*, vol. 8, no. 1, pp. 77–86, 2019.