

Integration of Specific Local Search Methods in Metaheuristic Algorithms for Optimizing the Feature Selection Process: A Survey

Isuwa Jeremiah¹, Mohammed Abdullahi², Sahabi Ali Yusuf³, Muhammad Nuruddeen Idris⁴,
and Baffa Shuaibu Garko⁵

^{1,5}Computer Science Department, Federal University of Kashere, Gombe Nigeria.

^{2,3}Computer Science Department, Ahmadu Bello University, Zaria Nigeria.

⁴Computer Science Department, Federal University Gusau, Zamfara Nigeria
isuwajeremiah@gmail.com¹, abdullahilwafu@abu.edu.ng², sahabiali@yahoo.com³,
minuruddeen66@gmail.com⁴, baffancy1@gmail.com⁵

Abstract

Metaheuristic algorithms have proven to be quite effective at solving global optimization issues, particularly feature selection difficulties. This class of algorithms often uses a specialized local search technique as an inner component or as a post-processing mechanism to improve the performance of their search process. This paper presents a comprehensive survey of the use of local search methods integrated into metaheuristic algorithms for optimizing the feature selection process. Based on the manner of operation, the local search methods examined in this study were classed as one-way or two-way. In addition, practical suggestions were also discussed to point out possible future directions.

Keywords—Metaheuristic Algorithms; Local Search Methods; Trajectory methods; Combinatorial Optimization; Feature Selection

1. Introduction

Optimization is the process of finding the optimal option from among a variety of alternatives. It is primarily employed in situations where decisions must be made in a specific set of circumstances (Sun et al., 2020). Optimization problems emanate from various fields and disciplines such as engineering design, agricultural services, manufacturing systems, economics, etc. Several studies have been conducted to solve optimization challenges by introducing new approaches or improving existing ones. Optimization problems come in different types but are majorly divided into continuous and discrete or combinatorial optimization (Vesselinova et al., 2020).

Combinatorial optimization involves selecting the most optimal subset of variables from a set of finite variables to minimize or maximize the objective function, which frequently entails not just single but several objectives (Ramos-figueroa et al., 2020). Combinatorial optimization has piqued the interest of scientists in fields as diverse as artificial intelligence, and operations research among others. Many of these issues are simple to define, but the process of solving them is quite difficult and complex (Ramos-figueroa et al., 2020). Such types of combinatorial optimization problems have been categorized as NP-hard problems (Vesselinova et al., 2020). Examples include the traveling salesman problem, knapsack problems, decision subset sum problem, and feature selection among others. However, in this paper, we solely discuss feature selection as a combinatorial optimization problem. The Feature Selection (FS) problem involves the selection of the most informative features from the total available features. For example, a dataset "A" with a total of "n" features, selects a feature subset B consisting of "m" relevant features, where $m < n$ and $B \subset A$ (Xue et al., 2013).

Several authors have categorized FS into three categories, namely the filter, wrapper, and embedded techniques (Adamu et al., 2021; Chaudhuri & Sahu, 2021; Huda & Banka, 2020). The filter approach assesses features using statistical information, the wrapper method evaluates the goodness of the selected feature subset using a learning algorithm (classifier), and the embedding method incorporates the benefits of both the filter and wrapper methods (Alzaqebah et al., 2020; Chaudhuri & Sahu, 2021; Huda & Banka, 2020; Tubishat et al., 2021). “Fig. 1” presents the general taxonomy of the FS methods.

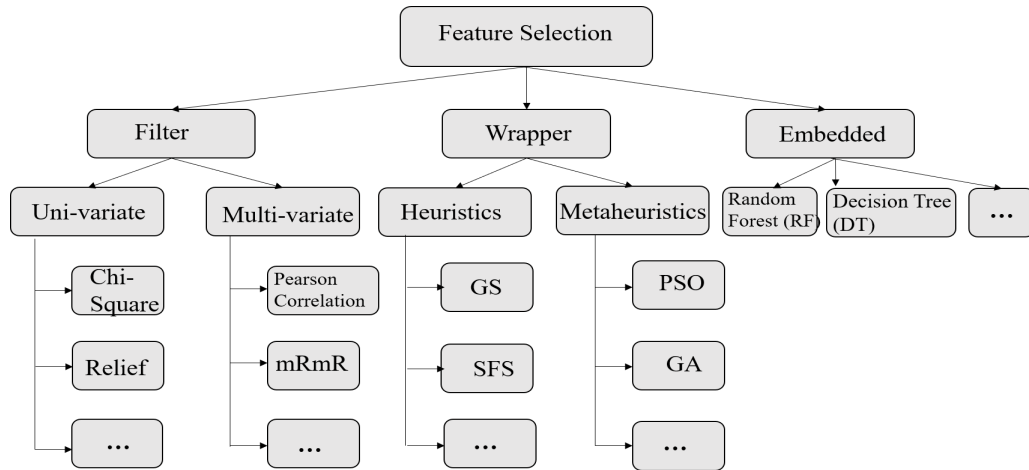


Fig. 1. General taxonomy of the feature selection method

Several studies on optimizing the FS process have been conducted, beginning with exhaustive, heuristic, and metaheuristic methodologies (Chaudhuri & Sahu, 2021). Exhaustive and heuristic approaches are regarded as impractical, computationally inefficient, and still stuck in local optima (Huda & Banka, 2020; Mafarja et al., 2020). As a result, metaheuristics methodologies were created to improve the efficiency of heuristic approaches (Sarhani et al., 2018).

Metaheuristic Algorithms (MAs) are based on global search theory and are designed to address the shortcomings of exhaustive and heuristic approaches (Mafarja et al., 2020). MAs such as the Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Salp Swarm Optimization (SSA) (Mirjalili et al., 2017), Genetic Algorithm (GA) (Holland, 1984), Moth Flame Algorithm (MFA) (Mirjalili, 2015b), Spotted Hyena Optimization (SHO) (Dhiman & Kumar, 2017), Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016), Harmony Search Algorithm (HSA) (Geem et al., 2001), Dragonfly Algorithm (DA) (Mirjalili, 2016), Coral Reef Optimization (CRO) (Salcedo-Sanz et al., 2013), Water Cycle Algorithm (WCA) (Eskandar et al., 2012), Ant Lion Optimization (ALO) (Mirjalili, 2015a), Clonal Selection Algorithm (CSA) (Castro & Zuben, 2002), Sailfish Optimization (SFO) (Shadravan et al., 2019), Harris Hawks Optimization (HHO) (Asghar et al., 2019), Ant Colony Optimization (ACO) (Dorigo & Stützle, 1999), Bat Algorithm (BA) (Yang, 2010), and Symbiotic Organisms Search (SOS) (Cheng & Prayogo, 2014) are some of the most used in solving FS problems available in the literature.

Because of the recognized issue with these MAs' exploitation phase, which causes local optima stagnation, the hybridization of these algorithms with specific local search algorithms such as the Simulated Annealing (SA) (Kirkpatrick et al., 1983), the Variable neighborhood search (VNS) (Hansen & Mladenović, 2005) among others have been studied (Ghosh et al., 2020). These local search algorithms are metaheuristics in and of themselves, but they are categorized as a trajectory or local search techniques since they deal with a single solution rather than a population of possible solutions like population-based metaheuristics (Chantar et al., 2021).

The main focus of our survey study is on integrating population-based MA with these local search algorithms to solve the FS process. To the best of our knowledge, there has been no work of this nature. However, Blot et al., (2018) presented a similar work but focuses on multi-objective combinatorial optimization not restricted to only feature selection problems. Other survey works carried out in a similar context to the proposed topic can be found in (Wang, 2019). According to Xue et al., (2016), works on MAs application for FS problems started around the early 90s, becoming most popular in 2007. As a result, our research solely covers works published between 2000 and 2021. Our goal is to provide researchers with a comprehensive survey of the state-of-the-art works on the subject matter, as well as a discussion of potential future work, which we believe will be of great benefit to prospective and contemporaneous researchers who wish to conduct a more in-depth investigation of the topic.

The remainder of the paper is laid out as follows: Section II has a brief introduction to key concepts on the issue; Section III contains a study of works in the literature arranged based on the type of local search method incorporated; Section IV presents the discussion on works in literature as well as recommendations for future works. Finally, section V contains the conclusion.

2. Preliminaries

In this section, we present the basic ideas to help understand the proposed topic. First, a brief explanation of population-based metaheuristic algorithms is presented, followed by a general overview of local search algorithms in the literature.

2.1 Population-Based Metaheuristic Algorithms

The phrase "metaheuristics" can be broken down into "meta," which means "beyond," and "heuristics," which means "to seek or direct a study or examination" (Sarhani et al., 2018). These metaheuristic algorithms provide high-quality solutions in a fraction of the time it takes to use typical traditional optimization techniques. Aside from population-based and single-point search (trajectory), Ghazali, (2009) classified MAs into several other categories based on selected characteristics i.e., nature-inspired against non-nature inspired, dynamic against static objective function, memory usage against memory-less methods, swarm Intelligence inspired against evolutionary inspired, etc.

The key idea behind MAs is to maintain a balance between exploration and exploitation during the search process. During the exploration or diversification phase, they are expected to explore un-visited regions, while in the exploitation or intensification phase, focusing on already-visited regions is required (Chakrabarti et al., 2019).

2.2 Local Search Algorithms

In this section, we choose to categorize the local search algorithms into two according to their mode of operation:

- i. One-way Local Search and
- ii. Two-way Local Search or Trajectory methods

Note that the justification behind the classification is due to the exploitation-only nature of (i) and the exploration-exploitation nature of (ii). The one-way LS is based solely on exploitation, that is, it makes local adjustments or perturbs a single solution until it improves. An example of LS methods in this class is the paired swap (Liao & Kuo, 2017), where the positions of two selected features in a solution are swapped to achieve a better result. The two-way LS, on the other hand, is based on both exploration and exploitation, which means that as a result of the changes made to the initial solution, several neighboring solutions are formed. The best among is then substituted and used as the new solution. An example is the Simulated Annealing.

2.2.1 One-way Local Search (*exploitation only*)

This category of local search algorithms operates in a most similar way to the mutation operator in the evolutionary genetic algorithm. It involves selecting one or more bits (features) in a solution and performing an operation mostly flipping or perturbation i.e., changing from 1 to 0 and vice versa (where ‘1’ represents selected features and ‘0’ for unselected features). The idea behind this is that, by flipping the bits, there is a possibility of de-selecting an irrelevant or redundant feature as well as a possibility of selecting a highly relevant feature. The presence of these irrelevant or redundant features causes a degradation in the classification performance (Alhamidi, 2020; Mafarja, 2019). Therefore, performing this perturbation provides room for possible improvement. Examples of local search algorithms in this category include the Sequential One Point Flipping (SOPF) (Ghosh et al., 2020), and NBChange (Alzaqebah et al., 2020) among others.

2.2.2 Two-way Local Search (*exploration-exploitation*)

This category of local search algorithms is also known as metaheuristics, single-point search, or trajectory methods (Ghazali, 2009). This class of local search algorithms also employs the exploration and exploitation phase (Al-betar, 2016). However, they operate by improving only a single solution (Chantar et al., 2021) unlike in the population-based metaheuristic where a population of possible solutions is considered. The two-way local search algorithm begins operation on a single initial solution and gradually evolves it into a better solution using the neighborhood concept (Ghosh et al., 2019). They are either incorporated into the inner workings of the corresponding population-based metaheuristic or used as a post-processing mechanism to refine the solution produced (Blot et al., 2018).

Hill Climbing (HC) is the most basic of all the two-way local search algorithms, and it serves as the foundation for the majority of the others (Al-betar, 2016). The HC does not involve any random component and is used to strengthen the exploitation abilities of population-based MAs (Blot et al., 2018). However, reports have been made that the HC is easily trapped in a local optimum (Blot et al., 2018). As a result of this constraint, multiple extensions to the classical HC have been developed, all to include various intelligent random operators (Al-betar, 2016). Some of the most common extensions of the HC include the Simulated Annealing (SA) (Kirkpatrick et al., 1983), the Tabu Search (TS) (Glover, 1986), the Greedy Randomized Adaptive Search Procedure (GRASP) (Feo et al., 1995), the Variable neighborhood search (VNS) (Hansen & Mladenović, 2005), the Guided Local Search (GLS) (Voudouris, 1998), the Iterated Local Search (ILS) (Kohlhardt et al., 1972), the Stochastic Local Search (SLS) (Hoos & Stützle, 2005) and their variants (Al-betar, 2016; Blot et al., 2018).

In general, both one-way and two-way local search algorithms have their advantages in the intensification/exploitation phase, since they can quickly find better solutions in the vicinity of the provided initial solution (Jia et al., 2019).

3. Existing works on the proposed topic

Following a thorough search on Google Scholar and Scopus, about 100 papers were collected with approximately 71 from Google Scholar and 29 from Scopus. Feature selection, metaheuristic algorithms, and local search are among the keywords used. However, only about 26 were deemed relevant for inclusion in the literature. The inclusion criteria restrict to only the integration of local search methods in population-based MAs for solving the FS problem from 2000 to 2021. These literary works are grouped into two parts based on the type of local search strategies employed and are sorted chronologically (from oldest to the most recent). Tables 1 and 2 present works on the topic that use one-way and two-way local search algorithms respectively, with each describing its proposed strategy, strengths, and weaknesses.

Table 1. works in literature with the one-way local search

S/N	Reference	MA(s)	Description of Local Search	Strength/Weakness
1	(Oh et al., 2004)	GA	Rem-g and Add-g local search operators. Operates by choosing g features simultaneously and moving them between feature subsets.	GA's exploitation capability was strengthened as a result of the improvements. Random particle initialization employed, on the other hand, may result in a loss of population diversity, resulting in sub-optimal results. Furthermore, microarray gene datasets were used without any suggestions for reducing the potentially high computing cost.
2	(Zhu et al., 2006)	GA	Performs the Add and Delete operation in the solution vector of an offspring using the concept of mutual information.	GA's exploitation capability was strengthened as a result of the improvements. However, The filter technique of fitness evaluation, i.e., mutual information, resulted in sub-optimal performance due to a lack of proper interaction among features.
3	(Kabir et al., 2011)	GA	Performs the Add and Delete operations using three basic steps to enhance the exploitation capability of an offspring. The Add operator selects a required number of features while the Delete operator throws away desired number of features from the feature set of an offspring.	As a result of the enhancements, GA's exploitation capabilities have improved. The use of correlation information in selecting features, on the other hand, does not guarantee proper feature interaction. As a result, performance may be sub-optimal.
4	(Meena et al., 2012)	ACO	Make a forward and a backward search in the search space on the best ant for a fixed distance to upgrade its solution	Parallelization was used to deal with the ACO's great computational power. Using forward and backward search, on the other hand, may result in a nesting effect, resulting in sub-optimal performance.
5	(Moradi & Gholampour, 2016)	PSO	Performs the Add and Delete operations using two basic steps to enhance the exploitation capability of an offspring. The Add operator selects a required number of features while Delete throws away desired number of features.	PSO exploitation capabilities have been enhanced as a result of the improvement. On the other hand, using correlation information in selecting features does not guarantee proper feature interaction. As a result, performance may fall short of expectations.
6	(Liao & Kuo, 2017)	SOS	Paired Swap: Involves swapping the values of two randomly selected features in the solution vector.	It may not be effective to select and swap simply two randomly selected features without considering the complete feature space of a solution vector.

S/N	Reference	MA(s)	Description of Local Search	Strength/Weakness
7	(Alzaqebah et al., 2020)	MFA	NBChange: Randomly select a feature from each solution in the population and change it from 1 to 0 and vice versa. NBMove: Randomly select a feature from each solution in the population and change its position. NBSwap: Randomly select two features from each solution in the population and swap their positions.	As a result of the enhancements, MFA's exploitation capability has been strengthened. Small adjustments in a solution vector, on the other hand, may only result in minor improvements.
8	(Tubishat et al., 2020)	SSA	Randomly select three features from the best solution and change from 1 to 0 and vice versa.	As a result of the enhancements, SSA's exploitation capability has been strengthened. Small adjustments in a solution vector, on the other hand, may only result in minor improvements.
9	(Ghosh et al., 2020)	PSO+ GA	Sequential One Point Flipping: Goes through the entire solution making changes from 1 to 0 and vice versa. Only accept a better solution.	Performing perturbation on a solution vector in a sequential manner may result in significant improvement. However, because of the multiple fitness evaluation, it is computationally expensive.
10	(Tubishat et al., 2021)	SSA	Randomly select two or five features from the best solution and change from 1 to 0 and vice versa	As a result of the enhancements, SSA's exploitation capability has been strengthened. Small adjustments in a solution vector, on the other hand, may only result in minor improvements

Table 2. works in literature with the two-way local search

S/N	Reference	MA(s)	Description of Local Search	Strength/Weakness
1	(Blum & Hutchison, 2013)	PSO	Variable Neighborhood Search (VNS)	Neighbor solutions are generated for each particle by the VNS in each iteration until the number of neighbors for each particle is equal to the number of particles in the population. this provides room for particle exploitation in a smaller swarm inside the whole swam. However, it incurs additional computational costs as a result of the hybridization and also, multiple fitness evaluation steps due to neighbor solutions generation.

S/N	Reference	MA(s)	Description of Local Search	Strength/Weakness
2	(Nekkaa & Boughaci, 2015b)	GA	Stochastic Local Search (SLS)	SLS is embedded in the core of GA to enhance its exploitation ability. The SLS is applied after every iteration to the best chromosome in the population to improve its fitness. However, it incurs additional computational costs as a result of the hybridization and also, multiple fitness evaluation steps due to frequent calling of the local search.
3	(Nekkaa & Boughaci, 2015a)	HSA	Stochastic Local Search (SLS)	Based on an empirically set probability, generate neighborhood solutions for the selected individuals in the population by randomly adding or dropping a feature from the feature vector of size 'n'. The neighbor solution with the best fitness is selected as the current solution for the next generation. Small adjustments in a solution vector, on the other hand, may only result in minor improvements.
4	(Marinaki & Marinakis, 2015)	CSA	Iterated Local Search (ILS) and Variable Neighborhood Search (VNS)	Hybridize the CSA+ILS+VNS to improve the search procedure and evolve a reduced feature set having high classification performance. More computational cost due to hybridization of CSA with two different local search methods.
5	(Mafarja Mirjalili, 2017a)	WOA	Crossover and Mutation operators	The crossover and mutation operators from GA were used to enhance the exploitation capability of the WOA by applying to the best or a randomly selected solution. Small adjustments in a solution vector i.e. low mutation rate, may, however, result in minor improvements.
6	(Mafarja Mirjalili, 2017b)	WOA	Simulated Annealing (SA)	Development of two FS methods based on WOA and SA. First, is the incorporation of SA in the core of WOA to enhance its exploitation. Secondly, apply SA to the best solution returned by WOA after each generation. SA aims to solve the local optima problem of SHO. However, due to the hybridization and many fitness evaluation steps required by repeated calls to the local search, it incurs significant computing costs.
7	(Yan et al., 2018)	CRO	Simulated Annealing (SA)	SA is embedded in the core of CRO to enhance its exploitation ability. The SA is applied to the best solution in the population

S/N	Reference	MA(s)	Description of Local Search	Strength/Weakness
8	(Mirjalili, 2018)	ALO	Incremental Hill-Climbing (IHC)	Proposed two FS methods based on ALO+QuickReduct and ALO+CEBARKCC. The two local search methods are hybridized in the core of the ALO to enhance the quality of the initial population and also to subsequently achieve optimum results.
9	(Alomari et al., 2018)	BA	Beta Hill Climbing (β HC)	β HC is embedded in the core of BA to enhance its exploitation ability to select the most informative subset of genes. The β HC is applied to the best solution in the population after each generation. Hybridization and frequent calls to the local search result in higher computing expenses.
10	(Jia et al., 2019)	SHO	Simulated Annealing (SA)	Development of two FS methods based on SHO and SA. First, the incorporation of SA in the core of SHO and apply to the best solution after each iteration. Secondly, apply SA on the final solution returned by SHO. SA aims to solve the local optima problem of SHO. Hybridization and frequent calls to the local search result in higher computing expenses.
11	(Al-rawashdeh et al., 2019)	WCA	Simulated Annealing (SA)	The development of three FS methods based on WCA and SA. Low-level: The SA is applied to the best solution of the WCA after a predefined number of generations. Interleave: The best solution is passed to the SA after each generation. High-level: Explorative capability of WCA and the fine-tuning capability of SA are explored at each generation. Hybridization and frequent calls to the local search result in higher computing expenses.
12	(Ghosh et al., 2019)	GA	Late Acceptance Climbing (LAHC)	LAHC is embedded in the core of GA to enhance its exploitation ability. The LAHC is applied after every generation to each chromosome in the population a specified number of times. Hybridization and frequent calls to the local search result in higher computing expenses.

S/N	Reference	MA(s)	Description of Local Search	Strength/Weakness
13	(Ghosh et al., 2020)	SFO	Adaptive Beta Hill Climbing (A-βHC)	Incorporated the A-βHC into the SFO to enhance both its exploration and exploitation capabilities. Additional computing costs as a result of hybridization.
14	(Elgamal et al., 2020)	HHO	Simulated Annealing (SA)	SA is embedded in the core of HHO to enhance its exploitation ability to escape local optima stagnation. The SA is applied to the best solution in the population after each generation. Hybridization and frequent calls to the local search result in higher computing expenses.
15	(Alzaqebah et al., 2021)	PSO	Late Acceptance Hill Climbing (LAHC)	LAHC is embedded in the core of PSO to enhance its exploitation ability. The LAHC is applied conditionally when no further improvement is possible or the solution is stuck in local optima. Additional computing costs as a result of hybridization.
16	(Chantar et al., 2021)	DA	Simulated Annealing (SA)	SA is embedded in the core of DA to enhance its exploitation ability. The SA is applied to the best solution in the population after each generation. Hybridization and frequent calls to the local search result in higher computing expenses.

4. Discussion and Future Directions

It is observed from works in the literature presented that few out of the population-based MAs recorded a fair number of occurrences in the last two decades concerning the proposed topic. Table 3 presents the entire MAs reviewed in our literature based on the type of local search algorithm employed.

Table 3. Categorization of MAs reviewed in literature based on the type of local search algorithms employed from 2000 to 2021

Metaheuristic Algorithms	Number of Times Applied		Total
	One-Way LS	Two-Way LS	
GA	4	2	6
PSO	2	2	4
SSA	2	0	2
WOA	0	2	2
MFA	1	0	1
SHO	0	1	1
HSA	0	1	1
DA	0	1	1

Metaheuristic Algorithms	Number of Times Applied		Total
	One-Way LS	Two-Way LS	
CRO	0	1	1
WCA	0	1	1
ALO	0	1	1
CSA	0	1	1
SFO	0	1	1
HHO	0	1	1
ACO	1	0	1
BA	0	1	1
SOS	1	0	1

It is observed that only PSO, GA, SSA, and WOA recorded more than one occurrence in total over the last two decades, with GA attaining the highest. One possible reason for GA recording the highest occurrence is due to its adverse limitation in exploitation i.e., the mutation operator in GA responsible for the exploitation is insufficient. PSO also recorded a fairly high number of occurrences after GA. Although the PSOs exploitation phase has been described as noteworthy (Ghosh et al., 2020), works in the literature demonstrated that further strengthening it with specific local search algorithms yields better performance. Judging from table 3, it is clear that not much work has been done in hybridizing population-based MAs with one-way local search algorithms. Therefore, this should draw the attention of researchers in this field. Contrary to this, hybridizing population-based MAs with two-way local search algorithms has been commendable with HC, SA, VNS, SLS, and their variants being the most frequently used. However, in most research, higher computing costs as a result of hybridization and frequent calls to the local search have been identified as the major downsides. Although several researchers have suggested parallelization in MAs to share the workload among CPUs or GPUs, establishing specialized approaches for reducing high computing costs, such as developing a low-cost fitness evaluation strategy, will surely assist (Xue et al., 2016). Nonetheless, none of the hybridized techniques reviewed in the literature can be said to have solved the local optimum stagnation problem maximally. Therefore, there is still a need for more superior techniques to be developed.

In the future, researchers in this field should study the effectiveness of hybridizing these population-based MAs and the local search algorithms that have yet to be tried in the literature. However, an utmost investigation should be carried out when selecting a combination as not all of them complement each other. Additionally, other new and promising population-based MAs such as the Black Widow Optimization (BWO) (Hayyolalam et al., 2020) effectiveness with either of the local search algorithms should be investigated. Finally, more focus should be placed on developing unique local search methods that are properly adapted to the peculiarities of population-based metaheuristic algorithms. This would increase their effectiveness and efficiency significantly.

5. Conclusion

The sole aim of this paper was to provide a comprehensive survey for the hybridization of population-based MAs with local search algorithms in solving feature selection problems. 17 population-based MAs that were hybridized with different specific local search algorithms to strengthen their exploitation capabilities were reviewed. The local search algorithms discussed in this paper were categorized into one-way and two-way local search algorithms based on their mode of operation. This survey reveals that different population-based MAs have over the years drawn the attention of several researchers to address feature selection problems. The most profound approaches are the PSO and GA which are among the oldest population-based MAs in literature. Concerning the local search methods, the HC, SA, SLS, VNS, and their variants were the most employed methods in the literature mostly because of their strength in exploitation thereby achieving optimum results.

References

- Adamu, A., Abdullahi, M., Junaidu, S. B., & Hassan, I. H. (2021). An hybrid particle swarm optimization with crow search algorithm for feature selection. *Machine Learning with Applications*, 6, 100108.
- Al-betar, M. A. (2016). b -Hill climbing : an exploratory local search. *Neural Computing and Applications*.
- Al-rawashdeh, G., Mamat, R., Hafhizah, N., & Abd, B. (2019). Hybrid Water Cycle Optimization Algorithm With Simulated Annealing for Spam E-mail Detection. *IEEE Access*, 7, 143721–143734.
- Alhamidi, M. R. (2020). *Optimal Feature Aggregation and Combination for Two-Dimensional Ensemble Feature Selection*.
- Alomari, O. A., Tajudin, A., Mohammed, K., & Mohammed, A. A. (2018). *A novel gene selection method using modified MRMR and hybrid bat-inspired algorithm with β -hill climbing*.
- Alzaqebah, M., Alrefai, N., Ahmed, E. A. E., Jawarneh, S., & Alsmadi, M. K. (2020). Neighborhood search methods with moth optimization algorithm as a wrapper method for feature selection problems. *International Journal of Electrical and Computer Engineering*, 10(4), 3672–3684.
- Alzaqebah, M., Jawarneh, S., Mohammad, R. M. A., Alsmadi, M. K., Al-marashdeh, I., Ahmed, E. A. E., Alrefai, N., & Alghamdi, F. A. (2021). *Hybrid feature selection method based on particle swarm optimization and adaptive local search method*.
- Angeline, P. J. (1998). Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1447, 601–610.
- Asghar, A., Mirjalili, S., Faris, H., & Aljarah, I. (2019). Harris hawks optimization : Algorithm and applications. *Future Generation Computer Systems*, 97, 849–872.
- Blot, A., Kessaci, M. É., & Jourdan, L. (2018). Survey and unification of local search techniques in metaheuristics for multi-objective combinatorial optimisation. *Journal of Heuristics*, 24(6), 853–877.
- Blum, C., & Hutchison, D. (2013). *LNCS 7919 - Hybrid Metaheuristics* (Issue May).
- Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, 237(February), 82–117.
- Castro, L. N. De, & Zuben, F. J. Von. (2002). *Learning and Optimization Using the Clonal Selection Principle*. 6(3), 239–251.
- Chakrabarti, S., Saha, H. N., University of British Columbia, Institute of Electrical and Electronics Engineers. Vancouver Section, Institute of Engineering & Management, University of Engineering & Management, & Institute of Electrical and Electronics Engineers. (2019). *Particle Swarm Optimization for Feature Selection: A Review of Filter-based Classification to Identify Challenges and Opportunities*.
- Chantar, H., Tubishat, M., Essgaer, M., & Mirjalili, S. (2021). Hybrid Binary Dragonfly Algorithm with Simulated Annealing for Feature Selection. *SN Computer Science*, 1–11.

- Chaudhuri, A., & Sahu, T. P. (2021). Feature selection using Binary Crow Search Algorithm with time varying flight length. *Expert Systems with Applications*, 168.
- Cheng, M., & Prayogo, D. (2014). Symbiotic Organisms Search : A new metaheuristic optimization algorithm. *COMPUTERS AND STRUCTURES*.
- Dhiman, G., & Kumar, V. (2017). Spotted hyena optimizer : A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software*, 0, 1–23.
- Dorigo, M., & Stützle, T. (1999). *Chapter 9 THE ANT COLONY OPTIMIZATION METAHEURISTIC : ALGORITHMS , APPLICATIONS , AND ADVANCES*.
- Elgamal, Z. M., Binti, N., Yasin, M., Tubishat, M., Alswaitti, M., & Mirjalili, S. (2020). *An Improved Harris Hawks Optimization Algorithm With Simulated Annealing for Feature Selection in the Medical Field*. 8.
- Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers and Structures*, 110–111, 151–166.
- Feo, T. ., Resende, M. G. ., & Smith, S. . (1995). *Greedy Randomized Adaptive Search Procedures*. 109–133.
- Geem, Z. W., K, J. H., & Loganathan, G. V. (2001). *Optimization Algorithm :*
- Ghazali, T. (2009). *Metaheuristics: From Design to Implementation | Wiley*. <https://www.wiley.com/en-us/Metaheuristics%3A+From+Design+to+Implementation+-p-9780470278581>
- Ghosh, K. K., Ahmed, S., Singh, P. K., Geem, Z. W. O. O., Sarkar, R. A. M., & Member, S. (2020). *Improved Binary Sailfish Optimizer Based on Adaptive β -Hill Climbing for Feature Selection*. 8.
- Ghosh, M., Guha, R., Alam, I., Lohariwal, P., Jalan, D., & Sarkar, R. (2020). Binary genetic swarm optimization: A combination of ga and pso for feature selection. *Journal of Intelligent Systems*, 29(1), 1598–1610.
- Ghosh, M., Kundu, T., Ghosh, D., & Sarkar, R. (2019). *Feature selection for facial emotion recognition using late hill-climbing based memetic algorithm*.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5), 533–549.
- Hansen, P., & Mladenović, N. (2005). Variable Neighborhood Search. *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, 211–238.
- Hayyolalam, V., Asghar, A., & Kazem, P. (2020). Engineering Applications of Artificial Intelligence Black Widow Optimization Algorithm : A novel meta-heuristic approach for solving engineering optimization problems ☆. *Engineering Applications of Artificial Intelligence*, 87(November 2018), 103249.
- Holland, J. H. (1984). *GENETIC ALGORITHMS AND ADAPTATION*. 317–333.
- Hoos, H. H., & Stützle, T. (2005). Stochastic Local Search. *Stochastic Local Search*.
- Huda, R. K., & Banka, H. (2020). New efficient initialization and updating mechanisms in PSO for feature selection

- and classification. *Neural Computing and Applications*, 32(8), 3283–3294.
- Jia, H., Li, J., Song, W., Peng, X., & Lang, C. (2019). Spotted Hyena Optimization Algorithm With Simulated Annealing for Feature Selection. *IEEE Access*, 7, 71943–71962.
- Kabir, M. M., Shahjahan, M., & Murase, K. (2011). A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing*, 74(17), 2914–2928.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942–1948.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680.
- Kohlhardt, M., Bauer, B., Krause, H., & Fleckenstein, A. (1972). Differentiation of the transmembrane Na and Ca channels in mammalian cardiac fibres by the use of specific inhibitors. *Pflügers Archiv European Journal of Physiology*, 335(4), 309–322.
- Liao, T. W., & Kuo, R. J. (2017). SC. *Applied Soft Computing Journal*.
- Mafarja, M. (2019). Boosting Salp Swarm Algorithm by Sine Cosine algorithm and Disrupt Operator for Feature Selection. *Expert Systems With Applications*, 113103.
- Mafarja, M. M., & Mirjalili, S. (2017a). Graphical abstract. *Applied Soft Computing Journal*.
- Mafarja, M. M., & Mirjalili, S. (2017b). *Hybrid Whale Optimization Algorithm with simulated annealing for feature selection*. 0, 1–11.
- Mafarja, M., Qasem, A., Heidari, A. A., Aljarah, I., Faris, H., & Mirjalili, S. (2020). Efficient Hybrid Nature-Inspired Binary Optimizers for Feature Selection. *Cognitive Computation*, 12(1), 150–175.
- Marinaki, M., & Marinakis, Y. (2015). REGULAR RESEARCH PAPER A hybridization of clonal selection algorithm with iterated local search and variable neighborhood search for the feature selection problem. *Memetic Computing*, 181–201.
- Martínez-Estudillo, A. C., Hervás-Martínez, C., Martínez-Estudillo, F. J., & García-Pedrajas, N. (2006). Hybridization of evolutionary algorithms and local search by means of a clustering method. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(3), 534–545.
- Meena, M. J., Chandran, K. R., Karthik, A., & Samuel, A. V. (2012). Expert Systems with Applications An enhanced ACO algorithm to select features for text categorization and its parallelization. *Expert Systems With Applications*, 39(5), 5861–5871.
- Mernik, M. (2013). *Exploration and Exploitation in Evolutionary Algorithms : A Survey*. 45(3), 1–33.
- Mirjalili, S. (2015a). Advances in Engineering Software The Ant Lion Optimizer. *Advances in Engineering Software*, 83, 80–98.
- Mirjalili, S. (2015b). Moth-Flame Optimization Algorithm : A Novel Nature-inspired Heuristic Paradigm. *KNOWLEDGE-BASED SYSTEMS*, July.

- Mirjalili, S. (2016). Dragonfly algorithm : a new meta-heuristic optimization technique for solving single-objective , discrete , and multi-objective problems. *Neural Computing and Applications*, 27(4), 1053–1073.
- Mirjalili, S. (2018). Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. *Soft Computing*.
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, 163–191.
- Mirjalili, S., & Lewis, A. (2016). *Advances in Engineering Software The Whale Optimization Algorithm*. 95, 51–67.
- Moradi, P., & Gholampour, M. (2016). A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing Journal*, 43, 117–130.
- Nekkaa, M., & Boughaci, D. (2015a). Hybrid Harmony Search Combined with Stochastic Local Search for Feature Selection. *Neural Processing Letters*.
- Nekkaa, M., & Boughaci, D. (2015b). *REGULAR RESEARCH PAPER A memetic algorithm with support vector machine for feature selection and classification*. 59–73.
- Oh, I., Lee, J., & Moon, B. (2004). *Hybrid Genetic Algorithms for Feature Selection*. 26(11), 1424–1437.
- Petalas, Y. G., Parsopoulos, K. E., & Vrahatis, M. N. (2007). Memetic particle swarm optimization. *Annals of Operations Research*, 156(1), 99–127.
- Ramos-figueroa, O., Quiroz-castellanos, M., & Mezura-montes, E. (2020). Metaheuristics to solve grouping problems : A review and a case study. *Swarm and Evolutionary Computation*, 53(January), 100643.
- Salcedo-Sanz, S., Pastor-Sánchez, A., Gallo-Marazuela, D., & Portilla-Figueras, A. (2013). *A Novel Coral Reefs Optimization Algorithm for Multi-objective Problems*. 326–333.
- Sarhani, M., El Afia, A., & Faizi, R. (2018). Facing the feature selection problem with a binary PSO-GSA approach. In *Operations Research/ Computer Science Interfaces Series* (Vol. 62, pp. 447–462). Springer New York LLC.
- Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). Engineering Applications of Artificial Intelligence The Sailfish Optimizer : A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems ☆. *Engineering Applications of Artificial Intelligence*, 80(January), 20–34.
- Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2020). A Survey of Optimization Methods from a Machine Learning Perspective. *IEEE Transactions on Cybernetics*, 50(8), 3668–3681.
- Tubishat, M., Idris, N., Shuib, L., Abushariah, M. A. M., & Mirjalili, S. (2020). Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Systems with Applications*, 145, 113122.
- Tubishat, M., Ja’afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M. A., & Omar, M. S. (2021). Dynamic Salp swarm algorithm for feature selection. *Expert Systems with Applications*, 164.

- Vesselinova, N., Steinert, R., Perez-Ramirez, D. F., & Boman, M. (2020). Learning Combinatorial Optimization on Graphs: A Survey with Applications to Networking. *IEEE Access*, 8(Section II), 120388–120416.
- Voudouris, C. (1998). Guided Local Search and Its Application to the Traveling Salesman Problem. *European Journal of Operational Research*, 113.
- Wang, J. (2019). A Brief Survey on Nature-Inspired Metaheuristics for Feature Selection in Classification in this Decade. *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, 424–429.
- Xue, B., Zhang, M., & Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, 43(6), 1656–1671.
- Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2016). A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation*, 20(4), 606–626.
- Yan, C., Ma, J., Luo, H., & Ashutosh, P. (2018). AC. *Chemometrics and Intelligent Laboratory Systems*.
- Yang, X. (2010). *A New Metaheuristic Bat-Inspired Algorithm*. 65–74.
- Zhu, Z., Ong, Y., & Dash, M. (2006). *Markov Blanket-Embedded Genetic Algorithm for Gene Selection*. December 2006.