computers

MDPI

Article

Framework of Meta-Heuristic Variable Length Searching for Feature Selection in High-Dimensional Data

Tara Othman Qadir Saraf [1,2,3,*], Norfaiza Fuad [2] and Nik Shahidah Afifi Md Taujuddin [2]

[1] Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja 86400, Johor, Malaysia
[2] Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja 86400, Johor, Malaysia
[3] Faculty of Software and Informatics Engineering, College of Engineering, Salahaddin University, Erbil 44001, Kurduistan, Iraq
* Correspondence: alsaraftara@gmail.com; Tel.: +60-0113-7065-258

Abstract: Feature Selection in High Dimensional Space is a combinatory optimization problem with an NP-hard nature. Meta-heuristic searching with embedding information theory-based criteria in the fitness function for selecting the relevant features is used widely in current feature selection algorithms. However, the increase in the dimension of the solution space leads to a high computational cost and risk of convergence. In addition, sub-optimality might occur due to the assumption of a certain length of the optimal number of features. Alternatively, variable length searching enables searching within the variable length of the solution space, which leads to more optimality and less computational load. The literature contains various meta-heuristic algorithms with variable length searching. All of them enable searching in high dimensional problems. However, an uncertainty in their performance exists. In order to fill this gap, this article proposes a novel framework for comparing various variants of variable length-searching meta-heuristic algorithms in the application of feature selection. For this purpose, we implemented four types of variable length meta-heuristic searching algorithms, namely VLBHO-Fitness, VLBHO-Position, variable length particle swarm optimization (VLPSO) and genetic variable length (GAVL), and we compared them in terms of classification metrics. The evaluation showed the overall superiority of VLBHO over the other algorithms in terms of accomplishing lower fitness values when optimizing mathematical functions of the variable length type.

Keywords: feature selection; high dimensional space; meta-heuristic; solution space; variable length

check for updates

Citation: Saraf, T.O.Q.; Fuad, N.; Taujuddin, N.S.A.M. Framework of Meta-Heuristic Variable Length Searching for Feature Selection in High-Dimensional Data. *Computers* **2023**, *12*, 7. https://doi.org/10.3390/computers12010007

Academic Editors: Phivos Mylonas, Katia Lida Kermanidis and Manolis Maragoudakis

Received: 30 October 2022
Revised: 16 December 2022
Accepted: 17 December 2022
Published: 27 December 2022

1. Introduction

Feature Selection becomes a significant process in building most machine learning systems. The role of feature selection is to exclude non-relevant features and to preserve only relevant features for the goals of training and prediction [1]. Feature selection appears in different areas, such as pattern recognition, data mining and statistical analysis [2]. The process of feature selection is regarded as important for improving the performance of prediction because less relevant features are excluded, and for increasing both memory and computation efficiency when the data are classified as high-dimensional data [3]. The literature contains three main classes of methods for feature selection [4]; the first one is the wrapper [5] and it measures the usefulness of features based on the classifier performance, such as information gain, the chi-square test, fisher score, correlation and variance threshold.

The second one is the filter [6], and it measures the statistical properties of features and their relevance without relying on the classifier for the repeated steps of training and cross-validation for enabling wrapper-based feature selection such as recursive feature elimination, sequential feature selection and meta-heuristic algorithms. It is regarded as efficient, but it is less accurate than the wrapper method. The third one is the embedded method [7], which differs in its use as an intrinsic model building during learning, such as decision tree and L1 regularization. We present the three classes in Figure 1. The usage of meta-heuristic algorithms in the wrapper methods is observed in the literature. However, there is a need to study their characteristics and the differences in their performance in terms of feature selection. One of the recent developments of meta-heuristics that serves the feature selection is variable length searching.

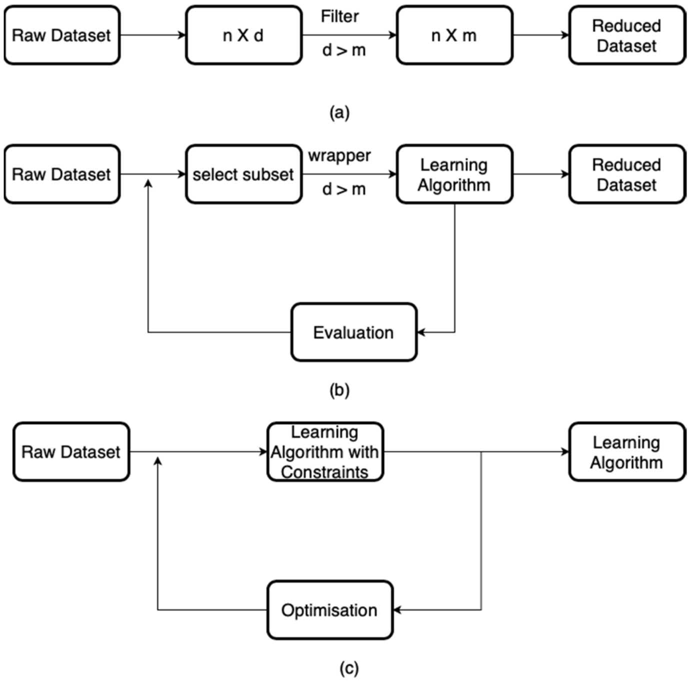

Figure 1. Conceptual diagram of (**a**) filter, (**b**) wrapper, (**c**) embedded method.

Meta-heuristic optimization algorithms are used by researchers for solving optimization problems [8]. They use the concept of generating random solutions and incorporating heuristic knowledge to develop them until reaching a convergence level in the improvement made over the solutions. The term used to describe the solution in the meta-heuristic algorithm varies from one algorithm to another. It is named a chromosome in genetics, a particle in particle swarm optimization, a star in black hole optimization, etc. Traditional meta-heuristic searching algorithms suffer from the limitation of fixed solution space. This means that the algorithms have an assumption of a fixed solution structure that does not apply to many research and real word problems. As an example, the clustering or segmentation problem cannot work on the pre-assumption of the number of clusters or segments in the image that make it a variable length optimization problem. Another example is the wireless sensor network deployment problem (WSND), which should work on the variable length of sensors before selecting the best deployment (number of them, their localization and configuration). A similar example is constellation optimization, which aims at searching over the space of satellite constellations to optimize coverage-related metrics [9]. A third example is an optimization of a convolutional neural network (CNN), which should also operate based on a variable length optimization algorithm because the number of layers that need to be optimized is fixed [10,11].

Variable length optimization is a sub-field of research with a focus on solving problems where the number of variables in the optimal solution is not known in advance [12]. The majority of approaches have been proposed in the literature, such as wind farm layout problems [13], wireless network design [14] and laminate stacking [15]. Researchers state that the research of fixed length optimization is mature; however, the research in variable length optimization is still in its infancy [16]. Some of the questions that are addressed include the following. Which is the more effective: the fixed or variable length meta-heuristic searching algorithm? How are effective operators designed for a variable length meta-heuristic searching algorithm? How are solutions handled for intra- and inter-class mobility? Selection methods that aim for some length variety in the set of parent solutions outperform selection methods that focus purely on objective value, according to [17]. Furthermore, due to the disruptive effects of altering solution lengths, it was shown that some operators were highly prone to producing an unwanted amount of badly performing solutions. In the work of [18], length niching selection was proposed. First, the population is divided into a number of niches based on the length of the solutions. To produce the parent population for the next generation, a local selection operator is applied independently to each niche. By choosing solutions from a variety of niches, the population remains diversified in terms of length. The term metameric was proposed to describe the segmented structure of the solution that contains similar variables [17]. When dealing with variable length optimization algorithms, it is important to define the metameric template of the problem. The meta-variables indicate the decision variables that combine the metameric variable. The variable length nature of the problem might occur from having solutions combined of different lengths of metameric variables and/or metavariables.

This article aims to study the recent development of meta-heuristic methods for serving the application of feature selection in high-dimensional space and the emergence of the class of variable length searching methods for feature selection. We are interested in three methods for the evaluation, namely genetic variable length, variable length particle swarm optimization and variable length black hole optimization with its two modes: position and fitness. The remainder of the article is organized as follows. In Section 2, we present the literature survey. Next, the methodology is presented in Section 3. Afterwards, the experimental results and analysis are presented in Section 4. Lastly, the conclusion and future works are presented in Section 5.

2. Literature Survey

The genetic algorithm is a type of heuristic algorithm inspired by the theory of evolution. It is used in optimization as a random searching algorithm with the capability of incorporating heuristic knowledge. Its capabilities come from the power of performing biological heuristic operators such as selection, mutation and crossover. Its concept is to build a chromosome that is a candidate solution to solve the problem and its degree of solving the problem is assessed based on the fitness functions. In the genetic process, a GA can generate a variety of individual genes and evolve the population. The methods of the genetic process include selection, mutation and crossover. To pick the superior and eliminate the inferior, the selection process mimics natural selection. The process of mutation and crossover allows for the creation of new individuals. The technical intricacies of the mutation and crossover processes are typically determined by the job at hand. For binary encoding, for example, a mutation operation can be designed to flip a single bit. In the work of [19], a variable length genetic algorithm (VLGA) for learning path recommendation was proposed. Because the sizes of the paternal chromosomes differ in VLGA, additional care must be taken while using the double-point crossover. They used double point crossover in conjunction with systems that prevent illegality in children's chromosomes. In the work of [20], a crossover operator prevents premature convergence by providing viable pathways with higher fitness values than their parents, allowing the algorithm to converge faster. The crossover supports variable length genetic optimization for robotic path planning. In the work of [21], bi-clustering algorithms to identify coherent and nontrivial bi-clusters were developed based on a variable length genetic optimization algorithm for low mean squared residue and high row variance. The algorithm uses three operators, namely selection, crossover and mutation with a designed fitness function based on the variable length strings. In the work of [22], variable length chromosome genetics was proposed for handling vehicle coordination multi-path problems. The goal of the algorithm is to organize vehicle arrival sequencing according to preset flow rates. The algorithm assumes non-symmetric traffic flow and it allows multiple paths instead of the fixed paths of intersection models. This enables any vehicle to go from any input point to any output branch in the intersection. In addition, the algorithm has designed its specific selection, crossover and mutation operators with the novel approach of carrying the crossover function between different-sized individuals. In the work of [23], the problem of unmanned aerial vehicle (UAV) deployment for the internet of things data collection platform has been handled. The goal was to optimize the energy consumption of the UAV based on minimizing the number and locations of stop points of the UAV. The optimization is regarded as a variable length optimization problem because the number of stops is unknown a priori. Consequently, the traditional fixed length crossover and mutation are changed. Each stop point's position is encoded into a person, and the total population thus symbolizes an entire deployment. Differential evolution is used to produce offspring throughout evolution. Then, based on the performance improvement, a strategy for adjusting the population size is devised. The number of stop points can be increased, decreased or kept constant using this technique. In the work of [3], a novel variable length particle swarm optimization for feature selection was proposed. It enables particles to have different lengths that improve the performance of the searching. In addition, the algorithm incorporates a solution order according to its performance. The order is based on the relevance of the features contained in the solution. In addition to evolutionary algorithms, researchers have developed variable length particle swarms [3]. We present an overview of metaheuristic searching algorithms that support variable length searching in Table 1.

Table 1. An overview of meta-heuristic articles with supporting variable length changing.

Author	Algorithm	Application	Operator
[19]	Variable length genetic	Learning path recommendation	Modified double-point crossover
[20]	Variable length genetic	Mobile Robot Path Planning	improved crossover operators
[21]	multiobjective genetic with variable length chromosome	Biclustering	Selection, crossover and mutation
[22]	genetic algorithm with variable length chromosomes	vehicle coordination multipath problem in intersections	selection, crossover and mutation operators with supporting variable length chromosome
[23]	Variable length genetic algorithm	UAV deployment for IoT data collection	Modified crossover and mutation
[3]	Variable length particle swarm optimization	High-Dimensional Classification	enabling particles to have different and shorter lengths
[24]	Variable length particle swarm optimization	Feature Selection on High-Dimensional Classification	Length-changing mechanism
[12]	Variable length particle swarm optimization	spaces with a variable number of dimensions	Modified mobility equation to change the length of the variable
[25]	adaptive variable length particle swarm optimization	optimization problem with the objective of minimizing the number of small base stations (SBSs) while satisfying both coverage and capacity constraints	Modified mobility equation

Overall, we find that many researchers have proposed variable length variants of genetic optimization and swarm optimization to solve various types of problems in various applications. Studying their performance and comparing them in solving the problem of feature selection is still an open research gap. Hence, we aim in this article at providing a framework for comparing variable length meta-heuristic searching in the problem of feature selection.

3. Methodology

This section presents the developed methodology for accomplishing the goal of the article. First, it presents variable length particle swarm optimization. Second, it presents variable length genetic optimization. Third, it presents variable length black-hole optimization. Fourth, it presents variable length black hole optimization.

3.1. Genetic Optimization

Genetic algorithms are based on biological principles. They take cues from Darwin's theory of evolution. Natural selection, according to Darwin's theory, selects the fittest individuals who then generate children. These individuals' characteristics are passed down the generations. If the parents are fit, their children will be fitter and have a better chance of surviving. This is something that genetic algorithms can learn from. They can be used to solve challenges related to optimization and search. Candidate solutions are evolved in genetic algorithms to produce better ones. The goal is to discover the best solution among a set of solutions that make up a search space. This is analogous to identifying the fittest person in a group. Genetic algorithms begin with a population of randomly generated solutions in the search space. Each solution has a chromosome, which stores information on the solution's properties. Changes to these chromosomes are possible. Selection, crossover and mutation are three bio-inspired operators that can be used on a chromosome in a standard genetic algorithm. Selecting a portion of the population as candidates for producing offspring and generating more solutions is referred to as selection. The fittest people are usually chosen. A fitness function can be used to calculate a solution's fitness, which indicates how good the solution is. Crossover is the process of combining the chromosomes of two parents to create a new chromosome for the offspring. The qualities of both parents' chromosomes are passed on to the offspring. Genetic algorithms are straightforward but powerful. They have been used to solve a variety of research challenges, including vehicle routing [26], power allocation [27], deep learning hyperparameter optimizations [28] and more. The pseudocode of genetic optimization is given in Algorithm 1.

Algorithm 1: Pseudocode of genetic optimization:

```
Input;
Objective Functions;
Number of Iterations;
Number of Population;
Output;
Optimal Solution;
Start;
    Generate initial population;
    Evaluate initial population;
    For each iteration until maximum iterations;
    Select elites using probabilistic model provided from population evaluation;
    Generate offspring using crossover and mutation and add them to pool of solutions;
    Evaluation pool of solutions;
    Select next generation from pool of solutions using environmental selection;
    End;
End.
```

3.2. Particle Swarm Optimization

Particle swarm optimization is another meta-heuristic algorithm used for random searching. Its concept is inspired by a swarm or flock collective behavior. For each individual in the swarm, the mobility model is responsible for moving it according to two components: best local and best global. The best local component is a velocity vector between the individual current position and the best local position. Similarly, the best global component is a velocity component between the individual current position and the best global position. The equation of moving solutions or particles is given as

$$v_{i,t} = wv_{i,t-1} + c_1 r_1 \left(x_{i,t} - x_{bl_i,t}\right) + c_1 r_1 \left(x_{i,t} - x_{bg,t}\right) \tag{1}$$

$$x_{i,t} = x_{i,t-1} + v_{i,t} \tag{2}$$

where:

w denotes the inertia;
c_1, c_2 denotes constants;
r_1, r_2 denotes random numbers between 0 and 1;
$x_{bl_i,t}$ denotes best local of solution i at moment t;
$x_{bg,t}$ denotes best global of solution i at moment t.

The pseudocode of particle swarm optimization is given in Algorithm 2.

Algorithm 2: Pseudocode of particle swarm optimization:

```
Input;
Objective Functions;
Number of iterations;
Size of swarm;
Output;
Start;
    Generate initial swarm;
    Evaluate initial swarm;
    For each iteration until maximum iteration;
    Select best global;
    For each solution;
    Find best local and move solution;
    End;
    Evaluation pool of solutions;
    End;
End.
```

3.3. Variable Length Variants

This section provides the methodology developed for the Comparative Evaluation of Meta-Heuristic Searching for Variable Length Searching. The methodology consists of presenting a variable length variant of each genetic optimization and particle swarm optimization. Afterwards, we provide benchmarking functions with variable length nature used for comparison. Lastly, we present the evaluation metrics used for our analysis.

The number of variables in variable length optimization problems is not always fixed. Traditional optimization methods can be used by assuming a limited number of variables because they were created for fixed-length design structures. Even so, a short length will result in an inferior solution. The problem-solving space, on the other hand, will vary depending on the value of the determinant variable, the design vector length. To put it another way, the unique search space makes the algorithm execution process more unique for proper space research. On the other hand, control values to accommodate these changes must be considered. In this section, we present three variants of variable length searching for meta-heuristic optimization.

3.3.1. Variable Length Particle Swarm Optimization

In this variable of variable length particle swarm optimization, each particle will have a different length L. The algorithm is based on a special variant of PSO named comprehensive learning CLPSO, with some modifications. First, in the original CLPSO, any particle can be used as an exemplar for a dimension of any particle. However, since in the variable length variants particles have different lengths, the selected particle for a certain dimension must have the same length as the corresponding dimension. Hence, the algorithm presents an exemplar selection mechanism.

The probability of choosing exemplars for each dimension of a particle (Pc) in the original CLPSO is set depending on its identity or index in the population and remains constant throughout the evolutionary process. As seen in Algorithm 3, particles with a lower index have a lower Pc than those with a higher index. As a result, according to CLPSO's use of Pc for exemplar selection, small-index particles are more likely to follow their own pbest. However, particles with higher fitness should learn from particles with lower fitness in order to find a better position or solution. The probability model of exemplar assignment is given by Equation (8).

$$Pc_i = 0.05 + 0.045\frac{e^{\frac{10(rank(i)-1)}{S-1}}}{e^{10}-1} \tag{3}$$

where:

S denotes the population size;
$rank(i)$ denotes the rank of particle i.

Algorithm 3: The pseudocode of exemplar assignment:

```
Input;
Particle $i$;
Output;
Exemplar for each dimension of particle $i$;
Start;
$L \leftarrow$ the length of particle $i$;
  For each $d$ = 1 until $L$;
    $Rnd \leftarrow$ generate random number from uniform distribution;
    $Pc_i \leftarrow Pc$ of particle $i$;
    If ($Rnd \geq Pc_i$);
      Exemplar[d] $\leftarrow i$;
    Else;
      $p_1 \leftarrow$ randomly selected particle that is different from $i$ and has length longer than $d$;
      $p_2 \leftarrow$ randomly selected particle that is different from $i$ and has length longer than $d$;
      Exemplar $\leftarrow$ the best among $p_1$ and $p_2$;
      End;
  End;
  Return Exemplar;
End.
```

In addition, the algorithm instead of setting a different length of each particle divides the solution space into smaller sub-spaces based on Equations (9) and (10).

$$DivSize = \frac{PopSize}{NbrDiv} \tag{4}$$

$$ParLen_v = MaxLen\frac{V}{NbrDiv} \tag{5}$$

where:

$DivSize$ denotes the number of particles in each division;
$PopSize$ denotes the population size;
$NbrDiv$ denotes the number of divisions;
$MaxLen$ denotes the maximum length or the dimensionality of the problem.

We observe that the particles in the same division will have the same length.

To arrange the feature ranking, the algorithm sorts the features in descending order according to their relevance. The literature contains various measures for this purpose such as symmetric uncertainty, which is a normalized version of information gain. In addition, the algorithm enables the length-changing mechanism to guide the algorithm toward a more optimal or promising area in the space.

3.3.2. Variable Length Genetic Optimization

The variable length of the genetic optimization is adapted from the work of [29]. The length of a metameric variable length genome can change, but it can only contain completely defined metavariables. Recombination and mutation operators can be used to add or remove metavariables from the genome. As a result, the typical genetic algorithm operators are ineffective. We use the cut-and-splice recombination, which is similar to two-point crossover with the exception that the crossover points in the two parents do not have to match. Therefore, the number of meta-variables in each child may be different than the number of meta-variables in either parent. For mutation, design-variable mutation and metavariable insertion or deletion are the two types of mutation. The overall number of design variables in the genome is inversely proportional to the rate of design variable mutation. Only one design variable is altered on average with each operator call. When utilizing the hidden-metavariable representation, the mutation rate is not affected by unexpressed metavariables, and the 'flag' variable is not affected by design variable mutation. A random number generator determines the magnitude of the mutation. A random number from a normal distribution with a standard deviation equal to 5% of the domain length of the design variable being altered determines the size of the mutation. A randomly generated metavariable will be inserted at a random place in the genome by the metavariable insertion mutation. The metavariable deletion mutation eliminates a metavariable from the genome at random. The insertion operation can only activate an unexpressed metavariable in the hidden-metavariable representation, in which case the design variables will be changed with new random values. The 'flag' variable of an expressed metavariable will be set to 'off' by the deletion action. The fixed-length GA does not use the insertion and deletion procedures.

3.3.3. Variable Length Black-Hole Optimization

Variable length black hole optimization (VLBHO) (add citation) is presented in Algorithm 4. The algorithm's inputs are as follows: Max iteration, numOfStars and rangeOfDimension all refer to the maximum number of algorithm iterations that will be carried out. RangeOfDimension refers to the range of dimensions connected to the search in the solution space. The algorithm's result is bho, a representation of a black hole object with the world's best gBest and other data.

The algorithm begins by using Max iteration, numOfStars and rangeOfDimension to initialize the black hole object (BHO). An original black hole object (bho) is returned. In this initialization () process, the initial population is created. Next, the algorithm iterates until Max iteration and it loops over the stars one by one to do the following: First, the function updatePosition () is used to update the star's location (); second, it uses updateFitness to update the star's fitness (); third, it refreshes the best using the global best 4. It uses UpdateEnergy () to update the energy. The method then runs an inner loop through each dimension and each exemplar in that dimension to assess the exemplar's energy, prohibiting it from serving as an example if the energy is below bho.Emin. The algorithm locates its follower stars and assigns each of them a new exemplar based on the star's dimension in order to deactivate its function as an example. On the other hand, when stagnation occurs or no progress is achieved for a predetermined amount of time, the algorithm is in charge of replacing the black hole.

When the method was seen in action, it was discovered that it adds two concepts: a black hole, which stands in for the absolute best, and an exemplar, which stands in for a solution with the same dimension as its predecessor. When the energy of the example falls below a particular threshold, it loses its function, whereas stagnation causes the black hole to lose its function.

Algorithm 4: The General algorithm of variable length black hole optimization:

```
Input;
Max_iteration;
numOfStars;
rangeOfDimension;
Emin;
Output;
bho object that includes: gBest and other information;
Start;
bho = initBHO(Max_iteration,numOfStars,rangeOfDimnesion);
for each iteration of Max_iteration;
2.1-for each star of bho.Stars;
    a.  bho =updatePosition(bho,star);
    b.  bho = updateFitnessANDpersonalBest(bho,star,itr);
    c.  bho.gBest = best(bho.stars);
    d.  bho = UpdateEnergy(bho, star, itr);
    e.  for each dimension in rangeOfDimension;
        e.1 for each exemplar;
        i   e.1.1. if bho.Stars(star).Energy(dimension) < bho.Emin;
                a   set_stars = get the stars that use this star as their
                    exemplar in this dimension;
                b   for each star in set_stars;
                c   bho.Stars(star).Exemplar(dimension) =
                    ExemplarAssignment(bho,star,dimension);
                d   end;
        ii  e.1.2.end;
    f.  e.2end;
    g.  end;
    h.  if gBest not improved for a time period;
    i.  bho = lengthChanging(bho);
    j.  end;
2.2-end;
End;
End.
```

4. Experimental Results and Analysis

The evaluation was conducted on MATLAB 2020b. For evaluation, we implemented four types of variable length meta-heuristic searching algorithms, namely VLBHO-Fitness, VLBHO-Position [30], variable length particle swarm optimization (VLPSO) and genetic variable length (GAVL). The evaluation was conducted on four functions, namely Rosenbrock, Rastrigin, Rastrigin and sphere.

The configuration is presented in Table 2.

Table 2. Configuration of the developed method and the benchmarks.

Parameter	VLBHO-Fitness	VLBHO-Position	VLPSO	Gavl
Population size	40	40	40	40
Iterations	50	50	50	50
Min-length	1	1	1	1
Max-length	10	10	10	10
Number of divisions	10	10	10	-
W	-	-	0.5	-
C	-	-	0.5	-
Alpha	-	-	7	-
Beta	4	4	4	-
Emax	10	10	-	-
Emin	10^{-3}	10^{-3}	-	-
EH	2	2	-	-
Time_window_length	5	5	-	-
T	5	5	-	-
elitism_rate	-	-	-	0.1
mutation_rate	-	-	-	0.2

The evaluation was performed based on four mathematical functions, namely Rosenbrock, Rastrigin, sphere and Griewank [31]. The fitness value was generated for each of the algorithms after running them for optimizing the functions. As observed in Figure 2, all algorithms accomplished the same performance for Rosenbrock.

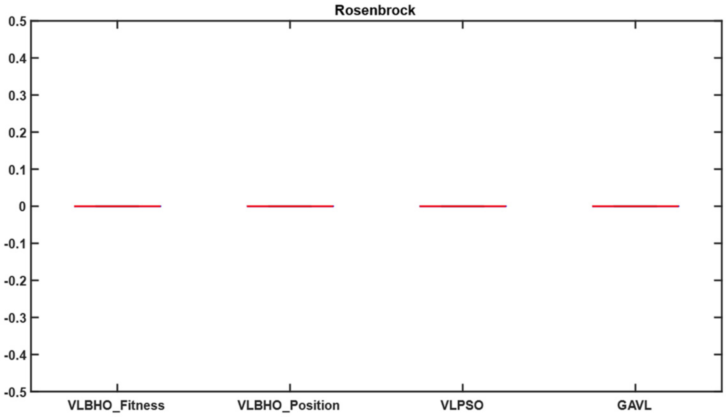

Figure 2. The fitness values after convergence for VLBHO fitness, VLBHO position, VLPSO for Rosenbrock.

For Rastrigin, the fitness values are provided in Figure 3. We find that GAVL was the best because it accomplished the lowest fitness value compared with the other benchmarking algorithms.

The fitness values for algorithms with respect to sphere are presented in Figure 4. As shown, VLBHO accomplished the best fitness value compared with the other algorithms followed by GVAL and then VLBHO. Similarly, visualizing the fitness value of Griewank in Figure 5, we find that VLBHO fitness provided the best performance compared with the benchmarks.

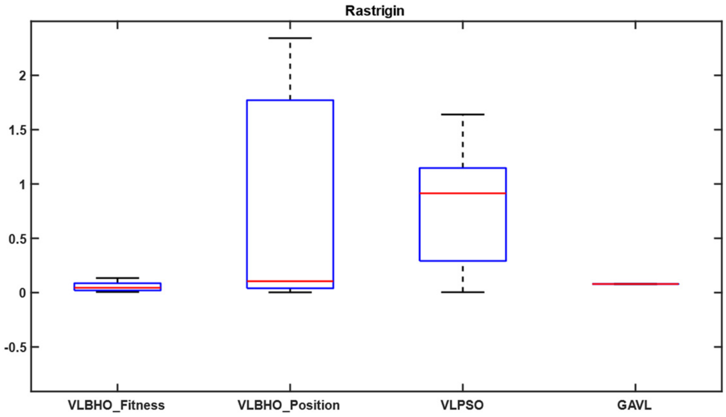

Figure 3. The fitness values after convergence for VLBHO fitness, VLBHO position, VLPSO for Rastrigin.

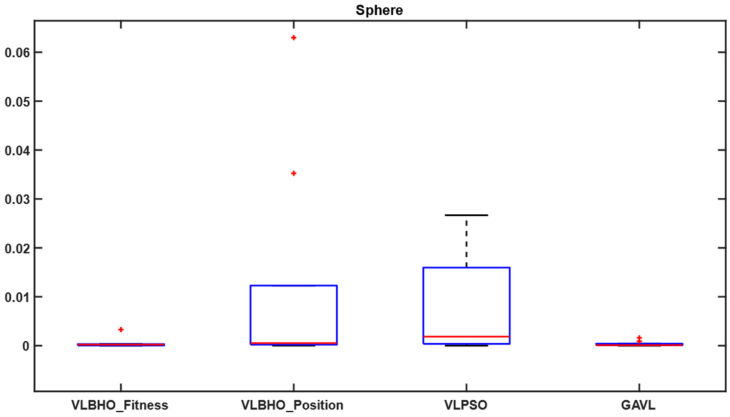

Figure 4. The fitness values after convergence for VLBHO fitness, VLBHO position, VLPSO for sphere.

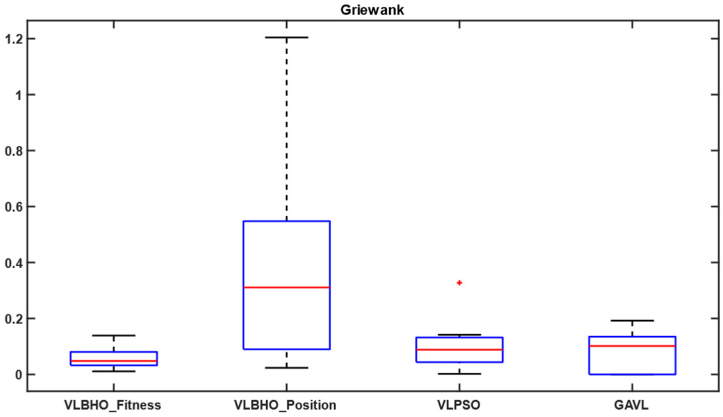

Figure 5. The fitness values after convergence for VLBHO fitness, VLBHO position, VLPSO for Griewank.

Analyzing the presented results, it can be stated that VLBHO fitness was superior to Griewank, sphere, Rastrigin and the equivalent in Rosenbrock.

5. Conclusions

This article studied the recent developments of meta-heuristic methods for serving the application of optimizing variable length space. The study considered three methods for the evaluation, namely genetic variable length, variable length particle swarm optimization and variable length black hole optimization with its two modes: position and fitness. The evaluation showed the overall superiority of VLBHO over the other algorithms in terms of accomplishing lower fitness values when optimizing mathematical functions of the variable length type. This research opens the door to adopting and adapting VLBHO for application in various areas of optimization research when the decision space does not fix length such as wireless sensor network deployment, data gathering and variable length feature selection.

Author Contributions: Conceptualization, T.O.Q.S., N.F. and N.S.A.M.T.; Formal analysis, T.O.Q.S., N.F. and N.S.A.M.T.; Visualization, T.O.Q.S. and N.F.; Writing—review & editing, N.S.A.M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data has been presented in main text.

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

References

1. Sahmoud, S.; Topcuoglu, H.R. A general framework based on dynamic multi-objective evolutionary algorithms for handling feature drifts on data streams. *Futur. Gener. Comput. Syst.* **2020**, *102*, 42–52. [CrossRef]
2. Solorio-Fernández, S.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F. A review of unsupervised feature selection methods. *Artif. Intell. Rev.* **2020**, *53*, 907–948. [CrossRef]
3. Tran, B.; Xue, B.; Zhang, M. Variable-Length Particle Swarm Optimization for Feature Selection on High-Dimensional Classification. *IEEE Trans. Evol. Comput.* **2018**, *23*, 473–487. [CrossRef]
4. Zebari, R.; AbdulAzeez, A.; Zeebaree, D.; Zebari, D.; Saeed, J. A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *J. Appl. Sci. Technol. Trends* **2020**, *1*, 56–70. [CrossRef]
5. Mafarja, M.; Mirjalili, S. Whale optimization approaches for wrapper feature selection. *Appl. Soft Comput.* **2018**, *62*, 441–453. [CrossRef]
6. Zhang, J.; Xiong, Y.; Min, S. A new hybrid filter/wrapper algorithm for feature selection in classification. *Anal. Chim. Acta* **2019**, *1080*, 43–54. [CrossRef]
7. Liu, H.; Zhou, M.; Liu, Q. An embedded feature selection method for imbalanced data classification. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 703–715. [CrossRef]
8. Qiao, W.; Yang, Z. Solving Large-Scale Function Optimization Problem by Using a New Metaheuristic Algorithm Based on Quantum Dolphin Swarm Algorithm. *IEEE Access* **2019**, *7*, 138972–138989. [CrossRef]
9. Hitomi, N.; Selva, D. Constellation optimization using an evolutionary algorithm with a variable-length chromosome. In Proceedings of the 2018 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2018; pp. 1–12.
10. Xiao, X.; Yan, M.; Basodi, S.; Ji, C.; Pan, Y. Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. *arXiv* **2020**, arXiv:arXiv:12703.
11. Wang, B.; Sun, Y.; Xue, B.; Zhang, M. A hybrid differential evolution approach to designing deep convolutional neural networks for image classification. In *Australasian Joint Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 237–250.
12. Kadlec, P.; Šeděnka, V. Particle swarm optimization for problems with variable number of dimensions. *Eng. Optim.* **2018**, *50*, 382–399. [CrossRef]
13. Kunakote, T.; Sabangban, N.; Kumar, S.; Tejani, G.G.; Panagant, N.; Pholdee, N.; Bureerat, S.; Yildiz, A.R. Comparative Performance of Twelve Metaheuristics for Wind Farm Layout Optimisation. *Arch. Comput. Methods Eng.* **2022**, *29*, 717–730. [CrossRef]
14. Jubair, A.M.; Hassan, R.; Aman, A.H.M.; Sallehudin, H. Social class particle swarm optimization for variable-length Wireless Sensor Network Deployment. *Appl. Soft Comput.* **2021**, *113*, 107926. [CrossRef]
15. Jalili, S.; Khani, R.; Maheri, A.; Hosseinzadeh, Y. Performance assessment of meta-heuristics for composite layup optimisation. *Neural Comput. Appl.* **2022**, *34*, 2031–2054. [CrossRef]
16. Al-Helali, B.; Chen, Q.; Xue, B.; Zhang, M. Genetic programming-based selection of imputation methods in symbolic regression with missing values. In *Australasian Joint Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 163–175.

17. Ryerkerk, M.; Averill, R.; Deb, K.; Goodman, E. A survey of evolutionary algorithms using metameric representations. *Genet. Program. Evolvable Mach.* **2019**, *20*, 441–478. [CrossRef]
18. Ryerkerk, M.; Averill, R.; Deb, K.; Goodman, E. A novel selection mechanism for evolutionary algorithms with metameric variable-length representations. *Soft Comput.* **2020**, *24*, 16439–16452. [CrossRef]
19. Dwivedi, P.; Kant, V.; Bharadwaj, K.K. Learning path recommendation based on modified variable length genetic algorithm. *Educ. Inf. Technol.* **2018**, *23*, 819–836. [CrossRef]
20. Lamini, C.; Benhlima, S.; Elbekri, A. Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. *Procedia Comput. Sci.* **2018**, *127*, 180–189. [CrossRef]
21. Maulik, U.; Mukhopadhyay, A.; Bandyopadhyay, S. Finding multiple coherent biclusters in microarray data using variable string length multiobjective genetic algorithm. *IEEE Trans. Inf. Technol. Biomed.* **2009**, *13*, 969–975. [CrossRef]
22. Cruz-Piris, L.; Marsa-Maestre, I.; Lopez-Carmona, M.A. A Variable-Length Chromosome Genetic Algorithm to Solve a Road Traffic Coordination Multipath Problem. *IEEE Access* **2019**, *7*, 111968–111981. [CrossRef]
23. Huang, P.-Q.; Wang, Y.; Wang, K.; Yang, K. Differential Evolution with a Variable Population Size for Deployment Optimization in a UAV-Assisted IoT Data Collection System. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, *4*, 324–335. [CrossRef]
24. Mohammadi, A.; Zahiri, S.H.; Razavi, S.M.; Suganthan, P.N. Design and modeling of adaptive IIR filtering systems using a weighted sum—Variable length particle swarm optimization. *Appl. Soft Comput.* **2021**, *109*, 107529. [CrossRef]
25. Wang, Y.; Zhu, X. A Novel Network Planning Algorithm of Three-Dimensional Dense Networks Based on Adaptive Variable-Length Particle Swarm Optimization. *IEEE Access* **2019**, *7*, 45940–45950. [CrossRef]
26. Dantzig, G.B.; Ramser, J.H. The Truck Dispatching Problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
27. Takshi, H.; Dogan, G.; Arslan, H. Joint Optimization of Device to Device Resource and Power Allocation Based on Genetic Algorithm. *IEEE Access* **2018**, *6*, 21173–21183. [CrossRef]
28. Han, J.-H.; Choi, D.-J.; Park, S.-U.; Hong, S.-K. Hyperparameter Optimization Using a Genetic Algorithm Considering Verification Time in a Convolutional Neural Network. *J. Electr. Eng. Technol.* **2020**, *15*, 721–726. [CrossRef]
29. Ryerkerk, M.L.; Averill, R.C.; Deb, K.; Goodman, E.D. Solving metameric variable-length optimization problems using genetic algorithms. *Genet. Program. Evolvable Mach.* **2016**, *18*, 247–277. [CrossRef]
30. Qadir, T.O.; Fuad, N.; Taujuddin, N.S.A.M. Variable Length Black Hole for Optimization and Feature Selection. *IEEE Access* **2022**, *10*, 63855–63866. [CrossRef]
31. Li, Q.Q.; He, Z.C.; Li, E. The feedback artificial tree (FAT) algorithm. *Soft Comput.* **2020**, *24*, 17. [CrossRef]

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