

# A Multi-Surrogate Assisted Salp Swarm Feature Selection Algorithm with Multi-Population Adaptive Generation Strategy for Classification

Zikang Yu

Hongbin Dong<sup>✉</sup>

Tianyu Guo

*Department of Computer Science and Technology, Harbin Engineering University, China*

Bingxu Zhao

*University of Victoria*

YUZIKANG@HRBEU.EDU.CN

DONGHONGBIN@HRBEU.EDU.CN

NOCHILI@HRBEU.EDU.CN

ZBX\_HEU@HRBEU.EDU.CN

**Editors:** Berrin Yanıkoğlu and Wray Buntine

## Abstract

The salp swarm algorithm(SSA) has been successfully used to solve the feature selection problem due to its fast convergence and simple structure. However, existing SSA-based methods still suffer from the issue of low classification accuracy due to the problem of getting trapped in local optima. Therefore, this paper proposes a novel feature selection method for classification based on SSA, which can continuously generate new sub-populations to improve the search environment of the main population. Specifically, a flip-prohibition(F-P) operator is first proposed to help the main population, which may currently fall into a local optimum, find a new and more promising region. A multi-surrogate technique is suggested to evaluate the region to determine the position of sub-populations, which can reduce the high computational cost. In addition, a population initialization method is developed according to the importance of features and the dimensionality of the dataset. Finally, a communication mechanism is presented to enable different sub-populations to learn from each other. By comparing the proposed method with other 6 feature selection methods on 16 datasets, we demonstrate that the proposed method has better classification ability and can select a smaller feature subset in most cases.

**Keywords:** feature selection, salp swarm algorithm, multi-population, classification, multi-surrogate.

## 1. Introduction

Feature selection can reduce redundant and unrelated features in the dataset, improve classification performance while successfully cutting down on computation, and effectively avert "dimensional disasters" Li et al. (2023). Because of its effectiveness, it has been widely used in different fields, such as network anomaly detection El Sayed et al. (2022), sentiment analysis Hosseinalipour and Ghanbarzadeh (2023), cancer classification Shaban (2023), etc.

Feature selection methods can be generally classified into 2 categories: filter-based and wrapper-based methods Dong et al. (2020). The primary distinction between them is that wrapper-based methods incorporate a learning/classification algorithm into the feature subset evaluation process Khaire and Dhanalakshmi (2022). The wrapper-based methods may

have better performance, but it has higher computational cost and poorer generalization. Filter-based models evaluate features without using any learning algorithms. Thus, it is usually fast. When selecting a feature subset, filters cannot reflect the relevance of each dimension, thus degrading the final classification performance [Sun et al. \(2023\)](#). Many researchers are now also proposing hybrid methods that advantages of both [Qu et al. \(2023\)](#). Despite the advancements in feature selection methods, the problem of dimensionality still poses a challenge due to the  $2^n$  possible solutions for a problem of dimension  $n$ .

In recent years, researchers have found that metaheuristic algorithms have shown positive effects in solving feature selection problems. The salp swarm algorithm (SSA) [Mirjalili et al. \(2017\)](#) is one of them. Due to its simple structure, lower computational cost, and easy integration with other algorithms, many SSA-based methods have been proposed. For example, [Qaraad et al. \(2022\)](#) developed an SSA-based algorithm for feature selection that uses local escape operators and quadratic interpolation to balance exploitation and exploration. [Faris et al. \(2020\)](#) proposed an SSA with a dynamic number of leaders and followers. However, the existing SSA-based feature selection methods still have deficiencies. First, the single-chain model of the basic SSA lacks diversity and tends to fall into local optima when encountering problems with a large number of local optima. Second, the existing SSA-based methods do not consider exploiting the correlation between features to improve the search environment of the population during the evolution of salps. Therefore, SSA still has great potential for improvement in solving feature selection problems.

In the process of finding the best solution, we often have to evaluate the fitness of a large number of solutions, which is expensive. To reduce the number of expensive true fitness assessments in the feature selection process, researchers use cheap surrogate models that approximately reflect the performance of solutions [Chen et al. \(2021\)](#). The surrogate models generally used are radial basis function neural network (RBFNN) [Liu et al. \(2022\)](#), Kriging [Dong et al. \(2021\)](#) and random forest [Sun et al. \(2019\)](#), etc. In this paper, surrogate techniques will be developed to predict the position of new populations.

Multi-population refers to dividing of a complete population into several independent sub-populations. Each sub-population may use a different search strategy or initialization method to ensure diversity. For example, [Kılıç et al. \(2021\)](#) proposed a new multi-population particle swarm optimization (MPPSO) algorithm for feature selection. It includes two populations that use random initialization and initialization based on relief sorting, respectively, to increase the diversity of the solution space. However, when the optimal solution in the population falls into the local optima, other individuals will also fall into it under its leadership. To address this issue, this paper proposes a multi-surrogate assisted salp swarm feature selection algorithm with a multi-population adaptive generation strategy (MSA-MPSSA). Unlike conventional multi-population methods, all salps in MSA-MPSSA are initially in a main population. When the main population falls into local optima, some individuals use the F-P operator and multi-surrogate to find new promising areas and form new independent sub-populations. This not only ensures diversity but also improves the search region of the population to avoid local optima. The following are the main contributions of this paper:

(1) A novel SSA-based multi-population adaptive generation strategy is proposed. When the main population falls into local optima, some salps are split to generate a new independent sub-population. This not only ensures diversity but also opens up a new independent

population to avoid local optima. To determine the position of the sub-population, an F-P operator is presented, which uses feature correlation and existing experience to generate a promising candidate pool for selection. To our knowledge, no studies have generated new independent populations during the search process.

(2) A multi-surrogate model is developed to estimate the fitness of the solutions in the candidate pool, so that each new sub-population can determine the initial position with a small time cost.

(3) A population initialization method based on the importance of features and the dimensionality of dataset is designed, which can improve the quality of the initial population.

(4) A communication mechanism is designed to help the populations learn from each other and improve local optima problems. It works for different population sizes.

The rest of this paper is organized as follows. In Section 2, the feature selection problem, SSA, and SSA-based feature selection methods are introduced. The proposed MSA-MPSSA is detailed in Section 3. In section 4, the proposed method is compared with 6 feature selection methods on 16 datasets. Finally, the full paper is concluded in Section 5.

## 2. Related Works

### 2.1. Feature Selection Problem

Feature selection is an important data preprocessing technique in machine learning and pattern recognition, which can significantly reduce the cost of model training and improve the performance of learning algorithms. The feature selection problem involves selecting  $d$  features from a dataset with  $D$ -dimensional features and  $M$  instances ( $d < D$ ), using a particular strategy. The objective is to optimize the performance metric  $f(X)$ , it can be described by Eq. (1).  $x_m$  takes the values 1 (the  $m$ th feature is selected) or 0 (not selected).

$$\begin{aligned} \min f(x) \\ \text{s.t. } X = (x_1, x_2, \dots, x_D) \\ x_m \in \{0, 1\}, m \in \{1, 2, \dots, D\} \end{aligned} \quad (1)$$

### 2.2. Salp Swarm Algorithm

As one of the swarm intelligence, SSA was proposed by Mirjalili et al. (2017) in 2017 to solve single-objective optimization problems. The SSA algorithm mimics the swarming behavior of salps forming a salp chain in the deep sea. The first salp in a salp chain is called the leader and the rest of the chain are called followers. Each salp is represented by a position vector of solutions in SSA. In the  $D$ -dimensional search space, the position vector is encoded as an  $D$ -dimensional vector.  $X^*$  is used to record the position of the food i.e. the best solution. The position of the leader is updated by the following equation:

$$X_j^1 = \begin{cases} X_j^* + c_1 ((ub_j - lb_j) c_2 + lb_j) & c_3 \geq 0.5 \\ X_j^* - c_1 ((ub_j - lb_j) c_2 + lb_j) & c_3 < 0.5 \end{cases} \quad (2)$$

The  $j$ th dimension position of the first salp (i.e. the leader) in the chain is denoted as  $X_j^1$ .  $ub$  and  $lb$  are the upper and lower bounds of the search space.  $c_2$  and  $c_3$  are random

numbers in  $[0,1]$ .  $c_1$  is a convergence factor used to balance exploration and exploitation, as shown in Eq. (3), where  $t$  and  $Tmax$  denote the current number of iterations and the maximum number of iterations, respectively. The position of the  $i$ th salp (i.e., follower) in the  $j$ th dimension is represented by  $X_j^i$  as in Eq. (4). It can be observed that its position is updated based on the guidance of the previous salp in the chain.

$$c_1 = 2e^{-\left(\frac{4t}{Tmax}\right)^2} \quad (3)$$

$$X_j^i = \frac{1}{2} \left( X_j^i + X_j^{i-1} \right), i \geq 2 \quad (4)$$

In this paper, each salp represents a solution (feature subset), and each dimension ranges within  $[0,1]$ . Set a threshold value of 0.5, when the position value of a dimension is greater than 0.5, the feature corresponding to that dimension is considered to be selected. Otherwise, it is not selected.

### 2.3. SSA-based Feature Selection Methods

Recently, various SSA-based methods have been widely applied to feature selection problems. Tubishat et al. [Tubishat et al. \(2021\)](#) proposed dynamic SSA with two main improvements. The first improvement is a new update formula for follower salp that increases the diversity of solutions. The second improvement is the introduction of a local search algorithm (LSA) to explore the optimal solution and improve the exploitation ability of the algorithm. However, the LSA significantly increases computational cost due to the high number of true fitness evaluations required.

[Saafan and El-Gendy \(2021\)](#) combined SSA with an improved WOA to enhance the exploitation and exploration abilities of the algorithm. [Zivkovic et al. \(2022\)](#) proposed a replacement mechanism to enhance the exploration ability of the algorithm by replacing the worst solution in early iterations. To explore the effect of parameter  $c_1$  on the performance of SSA, [Aljarah et al. \(2018\)](#) divided the salp chain into several sub-chains, and the  $c_1$  update methods of different chains are different. [Liu et al. \(2021\)](#) proposed an SSA-based framework with chaos assistance. To improve the exploitation ability of SSA, logic mapping is applied to generate new solutions for the population. However, these methods still suffer from the local optima, especially on high-dimensional datasets.

## 3. Proposed Method

### 3.1. MIC-based Population Initialization

The maximum information coefficient (MIC) is a statistical indicator that can be used to quantify the strength of the relationship between two variables [Wang et al. \(2023\)](#). We can use MIC to determine which features are more relevant to the class label, then we think they are more helpful for classification and need to be selected.

$$MIC(D) = \max_{p,q < B(n)} \frac{MI^*(D, p, q)}{\log(\min\{p, q\})} \quad (5)$$

Given variables  $X$  and  $Y$ ,  $D$  is the set of ordered pairs of variables.  $MI^*(\cdot)$  represents the maximum mutual information in the partition  $p * q$ , where  $B(n) = n^{0.6}$  is the upper

limit of the grid size and  $n$  is the number of samples. The MIC is a normalized statistic that ranges from 0 to 1, with higher values indicating stronger correlations between variables and lower values indicating independence. This overcomes the drawback of MI, which is not normalized.

We calculate the MIC values between features and class labels, and divide them into 4 groups from high to low. The group in front corresponds to a higher MIC value, and has a greater probability of being generated in the upper half of the search space (that is, selected) during initialization. However, we also consider that some features may have little correlation with the class label individually but may be important when combined with other features. So we use a random initialization method for 50% of the salps.

In addition, for high-dimensional datasets ( $D > 1000$ ), there may be more irrelevant or weakly correlated features. Therefore, we will give a threshold  $a$ , and remove features with MIC values lower than  $a$  in the original feature set in advance. Considering that not all datasets have the same range of MIC, we set threshold  $a_i = \delta \cdot MIC\_MAX_i$  for dataset  $i$ , where  $MIC\_MAX_i$  represents the largest MIC value in datasets  $i$ . By a sensitivity analysis,  $\delta$  was taken as 0.3.

### 3.2. Multi-Population Adaptive Generation Strategy

In the conventional multi-population method, all populations will be initialized in the initial stage. However, when the optimal solution of the population is a local optimum, it is difficult to produce a better solution to replace it, which will lead other individuals to the wrong direction. To solve this problem, we propose a multi-population adaptive generation strategy with only one main population in the initial stage. When the optimal solution of the main population does not change in  $\lambda$  iterations, we consider that it falls into local optima, where  $\lambda$  is set to 5 as suggested Qu et al. (2023). At this point,  $M$  (sub-population size) salps farthest from the leader in the salp chain of the main population will form a new independent sub-population, with the first salp as the leader and others as followers. The remaining main population will continue to search the current region, retaining the possibility of the main population. This method not only ensures diversity but also improves the search region of the population to avoid local optima (the effect of this method can refer to Section 4.4). But there are two challenges with the initialization of a new sub-population: 1. How to find a new region? 2. How to find a promising region? Therefore, we design a flip-prohibition (F-P) operator to solve the above challenges as much as possible. The specific steps are as follows:

*Step 1. Find a new region:* Given a threshold  $b$  (e.g. 0.2), if the event ( $rand < b$ ) holds for feature  $i$ , then feature  $i$  will be flipped. The aim is to generate a new solution for each salp using existing experience.

*Step 2. Find a promising region:* The flip of *Step 1* is random, but in order to find promising regions, this paper presents a MIC-based method to prohibit flipping. When a feature  $j$  satisfies *Step 1* and will be reversed, a new event ( $rand < P_j^{ban}$ ) is generated. Flipping will be prohibited if it holds.  $P_j^{ban}$  represents the flip prohibition probability of feature  $j$ , which is defined as follows.

$$P_j^{ban} = \begin{cases} \sqrt{MIC_j} \cdot A + B & X_j^i \geq 0.5 \\ (A + B) - \sqrt{MIC_j} \cdot A & X_j^i < 0.5 \end{cases} \quad (6)$$

$MIC_j$  is the MIC value between the  $j$ th feature and the class label. This method takes into account whether feature  $i$  is already selected or not. If feature  $i$  is already selected, the higher the  $MIC_i$ , the higher the chance of prohibiting flipping (i.e. keeping it). Conversely, if feature  $i$  is not selected, the lower the  $MIC_i$ , the higher the chance of prohibiting flipping (i.e. remaining unselected). The arithmetic square root is used to calculate  $P^{ban}$  as it is simple to implement and has a fast-to-slow growth curve for numbers between [0,1]. According to Chen et al. (2021), features with middle correlation are more promising for exploration. Therefore, in this method, the medium correlation features are quite different from the low correlation features, and not much different from the high correlation features. The range is adjusted to  $(B, B+A)$  using  $A$  and  $B$  (e.g., 0.45 and 0.05) when calculating  $P^{ban}$ . All parameters will be analyzed and given in Section 4.

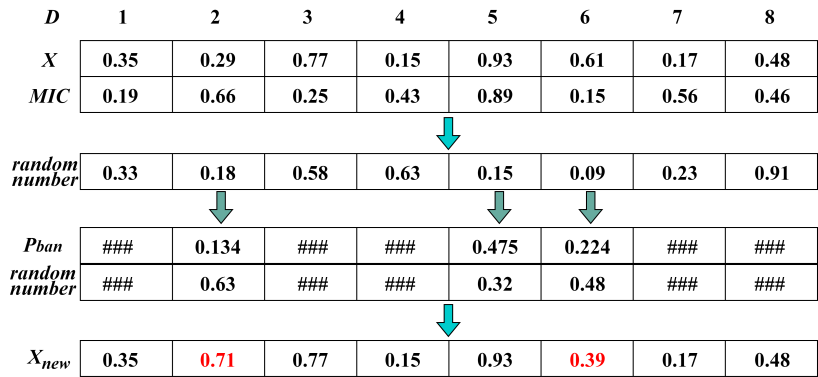


Figure 1: Illustration of F-P operation.

After performing the above two steps for each salp, we obtained a candidate pool. Figure. 1 illustrates the process of obtaining a new position by an F-P operation.  $X$  represents the position of a salp with 8 features. Only the features with serial numbers 2, 5, and 6 reached the events of step 1 and so will be flipped. Then we calculate the flip prohibition probability  $P^{ban}$  of these 3 features, if it is greater than the random number, the flip will be prohibited.  $X_{new}$  represents the new position obtained by  $X$  through the F-P operation. With the F-P operation, we utilize feature correlation in the flipping process and thus are able to find a new and promising set of solutions.

Finally, we use a surrogate model instead of true fitness function to predict the best  $M$  solutions as the positions of salps that constitute the new sub-population, which saves high computational costs. The surrogate model is constructed based on the position and true fitness of all salps from the previous iteration. We utilize two surrogates, RBFNN and k-nearest neighbor (KNN), which are used alternately each time a new sub-population is generated. Using a different surrogate each time takes into account the uncertainty of the surrogate. The ability of these two surrogates to predict the true fitness trends has been demonstrated in Liu et al. (2022) and Chen et al. (2021).

Figure. 2 illustrates the process of generating a new sub-population. Firstly, a candidate pool is obtained by performing an F-P operation on all salps. Then, the positions and fitness values of all salps are used to train the surrogate model. The trained surrogate model is used to predict the best  $M$  salps in the candidate pool. Finally, these  $M$  salps replace the

last  $M$  salps of the chain in the main population, resulting in a new sub-population for searching.

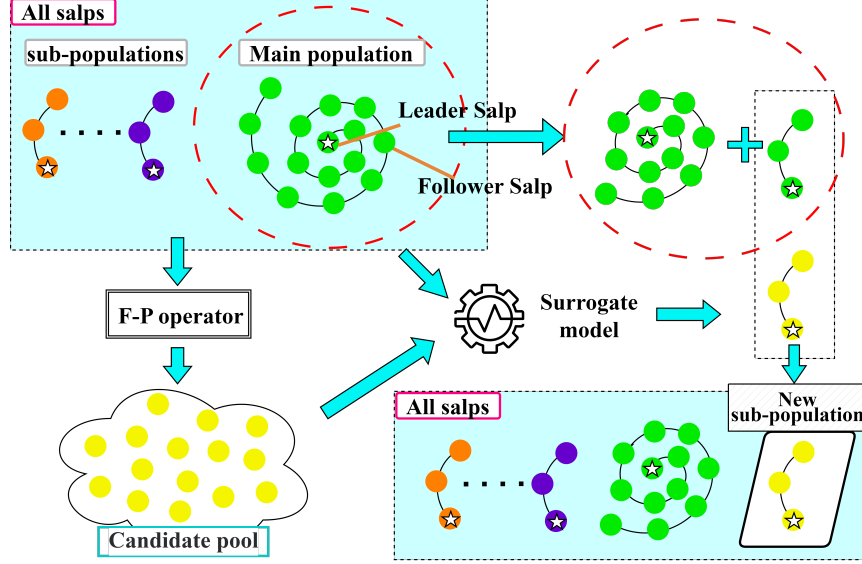


Figure 2: The process of generating a new sub-population.

Additionally, the salp follower update method is simpler and only considers the position of itself and the previous salp, which is easy to fall into local optima. So we used a fusion approach, combining a new physics-based metaheuristic algorithm proposed by Qais et al. (2020) with followers in SSA. The original Eq.(3) or Eq.(7) is chosen randomly.  $X_l$  represents the position of the  $l$ th iteration, and  $X_l^*$  represents the global optimum.  $r$  is a random number in  $[0, 1]$ ,  $T$  and  $C_1$  are random coefficients.

$$X_{l+1} = \begin{cases} X_l^* + e^{-T} [\cos(2\pi T) + \sin(2\pi T)] |(X_l - C_1 \cdot X_l^*)| & r \geq 0.5 \\ X_l^* + (X_l - C_1 \cdot X_l^*) e^{-T} & r < 0.5 \end{cases} \quad (7)$$

### 3.3. Communication Mechanism

During the iterative process, the main population is continuously improved by splitting it into new sub-populations. However, sub-populations may also suffer from the problem of falling into local optima. Cooperative co-evolutionary algorithms rely on the mutual cooperation of different populations to solve large-scale optimization problems Cai et al. (2021). Inspired by it, we designed a weight-based multi-population communication mechanism, which is applicable to different population sizes. Calling starts when the sub-population size reaches 3 or more. In this communication mechanism, each sub-population randomly selects two other sub-populations to communicate with, that is, replaces its worst solution with the weighted average of the best solutions from these two sub-populations. The weight is determined by the fitness, as indicated by the following formula.

$$X_m^{worst} = \left(1 - \frac{f_{r_1}^*}{f_{r_1}^* + f_{r_2}^*}\right) \cdot X_{r_1}^* + \left(1 - \frac{f_{r_2}^*}{f_{r_2}^* + f_{r_1}^*}\right) \cdot X_{r_2}^* \quad (8)$$

---

**Algorithm 1** MSA-MPSSA

---

**Input:** population size:  $N$ , maximum iteration:  $T_{max}$ , the number of individuals contained in each sub-population:  $M$ , the current number of independent populations:  $m$

**Output:** best salp position  $X^*$

```

1: Initialize the positions of all salps (Section 3.1)
2:  $m = 1$  // main population
3: while  $t < T_{max}$  do
4:   Calculate real fitness of each salp using Eq. (9)
5:   Update the best salp  $X^*$  of each population
6:   Update  $c_1$  by Eq. (3)
7:   if  $m < \frac{N}{M}$  &&  $X_{master}^*$  did not change in 5 iterations then
8:     Splitting the main population to generate a new independent sub-population (Section 3.2)
9:      $m = m + 1$  // At most  $\frac{N}{M}-1$  sub-populations
10:  end if
11:  for each salp in all populations do
12:    if  $X_i$  is leader then
13:      Update  $X_i$  using Eq. (2)
14:    else
15:      Update  $X_i$  using Eq. (4) or Eq. (7)
16:    end if
17:  end for
18:  if  $m \geq 4$  then
19:    for each sub-population do
20:      if best salp did not change in 5 iterations then
21:        Call communication mechanism (Section 3.3)
22:      end if
23:    end for
24:  end if
25:   $t = t + 1$ 
26: end while
return the best  $X^*$  among all populations

```

---

$X_{r_1}^*$  is the best salp of the first random sub-population and  $f_{r_1}^*$  is its fitness value. It can be seen that the smaller the fitness value, the greater the weight (according to the Eq. (9), the smaller the fitness, the better the solution).  $X_m^{worst}$  represents the worst solution of the  $m$ th sub-population, which will be replaced. When  $X_{r_1}^*$  and  $X_{r_2}^*$  are close to the global optimal solution, replacing the worst solution of the population can improve the poor population. Additionally, when the random solution is far from the global optimum, the obtained new individuals are expected to go to a new search region, which can test whether the optimal solution of the population is a local optimum.

The pseudocode of the proposed MSA-MPSSA is given in Algorithm 1.



## 4. Experimental Research

### 4.1. Experiment Settings

To evaluate the quality of a solution, two essential metrics are considered: minimizing the classification error rate and maximizing the dimension reduction rate. Thus, the fitness is calculated by Eq.(8), where  $\alpha$  is the weight factor,  $S$  is the number of selected features, and  $D$  is the total number of features. To avoid data imbalance, we used balanced accuracy(BA) [Bhowan et al. \(2011\)](#), where  $IC_i$  represents the proportion of correct recognition in class  $i$ . The weight of each class is  $\frac{1}{c}$  and  $\alpha$  is set to 0.9 [Qu et al. \(2023\)](#); [Chen et al. \(2021\)](#).

$$Fitness = \alpha \cdot (1 - BA) + (1 - \alpha) \cdot \frac{S}{D} \quad (9)$$

$$BA = \frac{1}{c} \cdot \sum_{i=1}^c IC_i \quad (10)$$

To validate the performance of the proposed method, we adopted 16 datasets from UCI machine learning repository [Dua and Graff \(2017\)](#) and [Qu et al. \(2023\)](#). Table 1 shows the details of the datasets. For the classifier, we used the most classical KNN, and employed 5-fold cross-validation. The classification balanced accuracy (CBA), the dimension reduction rate (DRR), and the running time will be recorded. To avoid chance, each dataset was independently run 30 times. Moreover, we conduct a statistical test to compare the performance of our method with competing methods. In this paper, we used the Wilcoxon rank sum test with a significance level of 0.05. In the subsequent results, we used the symbols " $\approx$ ", " $-$ " and " $+$ " to signify similar, weaker, and stronger performance than competing methods, respectively.

Table 1: Experimental datasets

No.	Dataset	Number of features	Number of instances	Number of classes
1	HeartEW	13	270	2
2	SpectEW	22	267	2
3	Horse	27	368	2
4	flags	28	194	8
5	BreastEW	30	568	2
6	ionosphere	34	351	2
7	Sonar	60	208	2
8	PenglungEW	325	73	7
9	Yale	1024	165	15
10	warpPIE10P	2420	210	10
11	lymphoma2	4026	96	9
12	Breast3	4869	95	3
13	Nci	5244	61	8
14	Leukemia1	5327	72	3
15	9_Tumors	5726	60	9
16	Lung_Cancer	12600	203	5

## 4.2. Parameter Settings

In the experiments, besides the basic SSA, we compared four most advanced SSA-based feature selection methods, namely Dynamic SSA(DSSA) [Tubishat et al. \(2021\)](#), hybrid improved Whale Optimization SSA(IWOSSA) [Saafan and El-Gendy \(2021\)](#), SSA with replacement mechanism and sine cosine algorithm(SSARM-SCA) [Zivkovic et al. \(2022\)](#), and termite colony inspired multi-population SSA(TCSSA3) [Aljarah et al. \(2018\)](#), and compared with the multi-population PSO (MPPSO) [Kılıç et al. \(2021\)](#). Table 2 shows the parameter settings for all methods. Moreover, we set all methods with the same population size of 50 and iteration number of 100.

Table 2: Parameter setting of algorithm

Algorithms	Parameter settings
MSA-MPSSA	$c_2, c_3 \in (0, 1)$ , the MIC threshold $a=0.3$ , the flipping threshold $b=0.2$ , sub-population parameter $M=\frac{N}{10}$ , range constants for prohibiting flipping probability $A=0.45, B=0.05$ .
DSSA <a href="#">Tubishat et al. (2021)</a>	$c_2, c_3 \in (0, 1)$ , LSA_maximum_iterations=10
IWOSSA <a href="#">Saafan and El-Gendy (2021)</a>	$c_2, c_3 \in (0, 1)$ ; $r, p, a_3 \in (0, 1)$ , $b = 1$ .
SSARM-SCA <a href="#">Zivkovic et al. (2022)</a>	$c_2, c_3 \in (0, 1)$ , $rms = T_{max}/10$ , $wrs = N/5$ .
TCSSA3 <a href="#">Aljarah et al. (2018)</a>	$c_2, c_3 \in (0, 1)$ , number of leaders = $N/2$
MPPSO <a href="#">Kılıç et al. (2021)</a>	$c_1 = c_2 = 2$ , $W_{max} = 0.9, W_{min} = 0.4$
SSA <a href="#">Mirjalili et al. (2017)</a>	$c_2, c_3 \in (0, 1)$

Table 3: The average CBA of MSA-MPSSA at different M

No.	$M=\frac{N}{3}$	$M=\frac{N}{5}$	$M=\frac{N}{8}$	$M=\frac{N}{10}$	$M=\frac{N}{12}$	$M=\frac{N}{15}$
1	0.8825	0.8832	0.8863	<b>0.8873</b>	0.8867	0.8869
2	0.8390	0.8391	<b>0.8441</b>	0.841	0.8384	0.8391
3	0.8357	0.8429	0.8453	0.8449	<b>0.8475</b>	0.8428
4	0.6270	0.6322	0.6343	0.636	0.6422	<b>0.6442</b>
5	0.9620	0.9645	0.9633	<b>0.9649</b>	0.9636	0.9628
6	0.9363	0.9368	0.9372	<b>0.94</b>	0.9359	0.9359
7	0.9065	0.9052	0.9043	<b>0.9107</b>	0.9000	0.9087
8	0.9426	0.9473	0.9498	<b>0.9545</b>	0.9529	0.9501
9	0.7096	0.7131	0.7051	0.7169	<b>0.7252</b>	0.7200
10	0.9749	0.9791	0.9785	0.9774	<b>0.9794</b>	0.9772
11	0.9709	0.9691	0.9735	0.9721	0.9714	<b>0.9740</b>
12	0.7478	0.7461	0.7447	<b>0.7539</b>	0.7459	0.7445
13	0.8555	<b>0.8654</b>	0.8534	0.8602	0.8557	0.8522
14	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>
15	0.6680	0.6752	0.6876	0.6809	0.6831	<b>0.6881</b>
16	0.9796	0.9807	0.9819	<b>0.9825</b>	0.9814	0.9816
rank	6	5	4	1	2	3

There are several important parameters in this paper, namely the MIC threshold  $a$ , the number of salps  $M$  in the new sub-population, the flipping threshold  $b$  in the F-P operation and the range constants  $A, B$  for calculating  $P_{ban}$ . Based on Table 7 and Table 8 (appendix),

$a$  and  $b$  were set to 0.3 and 0.2, respectively. Additionally, we performed sensitivity analyses for  $A$  and  $B$ . Since the minimum value of  $P_{ban}$  should be greater than 0,  $B$  is set to a small value of 0.05 and experimented with the values of  $A(0.2\sim 0.6)$ . Finally,  $A$  takes 0.45.

For  $M$ , we conducted an experimental comparison using different values and recorded the results in Table 3, where  $N$  is the total number of salps. From the table, we observe that as  $M$  decreases, the classification performance first increases and then decreases. This is because the smaller  $M$  has more sub-populations that can improve the search region when the main population encounters a local optimum. However, when  $M$  is too small, there are too few salps per sub-population, so the search ability also decreases. Therefore, we set  $M$  to  $\frac{N}{10}$  in our experiments.

### 4.3. Comparison of MSA-MPSSA with Competing Methods

Table 4 shows the average CBA of all methods running 30 times on 16 datasets. The best-performing methods on the dataset are marked in bold. We can see that among these 7 methods, the MSA-MPSSA performs best on all datasets. Moreover, the results of the Wilcoxon test demonstrate that in most cases, the CBA of MSA-MPSSA is significantly better than the competing methods.

Table 4: The average CBA of the MPSSA and competing algorithms over 30 runs

No.	DSSA	SSARM -SCA	IWOSSA	TCSSA3	SSA	MPPSO	MSA -MPSSA
1	0.8823( $\approx$ )	0.8540(+)	0.8769(+)	0.8563(+)	0.8646(+)	0.8858( $\approx$ )	<b>0.8873</b>
2	0.8344(+)	0.8082(+)	0.8396( $\approx$ )	0.8160(+)	0.8243(+)	0.8358(+)	<b>0.8410</b>
3	0.7349(+)	0.7123(+)	0.8100(+)	0.7339(+)	0.7260(+)	0.7725(+)	<b>0.8449</b>
4	0.5315(+)	0.4534(+)	0.5786(+)	0.4883(+)	0.4849(+)	0.5641(+)	<b>0.6360</b>
5	0.9623(+)	0.9541(+)	0.9582(+)	0.9524(+)	0.9600(+)	0.9619(+)	<b>0.9649</b>
6	0.9025(+)	0.8716(+)	0.9270(+)	0.8845(+)	0.8808(+)	0.9200(+)	<b>0.9400</b>
7	0.8950(+)	0.8732(+)	0.8940(+)	0.8839(+)	0.8873(+)	0.9086( $\approx$ )	<b>0.9107</b>
8	0.9418(+)	0.9140(+)	0.9399(+)	0.9354(+)	0.9391(+)	0.9471(+)	<b>0.9545</b>
9	0.7031(+)	0.6865(+)	0.7011(+)	0.6980(+)	0.7017(+)	0.7074(+)	<b>0.7169</b>
10	0.9570(+)	0.9463(+)	0.9564(+)	0.9550(+)	0.9541(+)	0.9564(+)	<b>0.9774</b>
11	0.9484(+)	0.9207(+)	0.9436(+)	0.9452(+)	0.9475(+)	0.9487(+)	<b>0.9721</b>
12	0.7384(+)	0.7046(+)	0.7533( $\approx$ )	0.7326(+)	0.7337(+)	0.7364(+)	<b>0.7539</b>
13	0.8413(+)	0.8337(+)	0.8421(+)	0.8406(+)	0.8444(+)	0.8454(+)	<b>0.8602</b>
14	0.9736(+)	0.9558(+)	0.9863(+)	0.9734(+)	0.9764(+)	0.9888(+)	<b>1.0000</b>
15	0.6202(+)	0.5890(+)	0.6446(+)	0.6073(+)	0.6217(+)	0.6354(+)	<b>0.6809</b>
16	0.9579(+)	0.9375(+)	0.9689(+)	0.9548(+)	0.9565(+)	0.9660(+)	<b>0.9825</b>
Overall	15+,1 $\approx$	16+	14+,2 $\approx$	16+	16+	14+,2 $\approx$	

When compared with DSSA, IWOSSA, SSARM-SCA, and TCSSA3, MSA-MPSSA achieves significantly better results in 61 out of 64 comparisons. This shows that MSA-MPSSA not only outperforms the basic SSA algorithm but also has surprising effects in the SSA-based improvement methods. In high-dimensional datasets with more than 5000 features, MSA-MPSSA outperforms all competing methods by far. This benefits from the MIC-based initialization method, which not only filters a batch of irrelevant features in

advance but also makes the population have high-quality initial positions. Compared with MPPSO, it still faces the problem of falling into local optima, despite using two populations to increase diversity. In contrast, the proposed F-P operator and multi-surrogate can generate new sub-populations that can improve the main population already trapped in local optima. The sub-populations can also learn from each other using the proposed communication mechanism. Furthermore, the classification accuracy of MSA-MPSSA on the high-dimensional dataset Leukemia1 reaches an amazing 100%. These results demonstrate the superiority of MSA-MPSSA in classification, especially on the high-dimensional dataset.

Table 5: The average DRR(%) over 30 runs

No.	DSSA	SSARM -SCA	IWOSSA	TCSSA3	SSA	MPPSO	MSA -MPSSA
1	65.90	62.56	70.00	66.92	63.08	74.36	<b>76.41</b>
2	58.94	51.21	70.00	60.45	57.88	62.27	<b>80.76</b>
3	63.21	59.88	<b>93.33</b>	61.23	59.51	74.57	91.98
4	58.45	52.38	<b>88.69</b>	62.98	60.83	75.36	84.40
5	69.78	69.78	90.33	60.44	64.11	78.33	<b>93.00</b>
6	71.76	66.96	<b>91.27</b>	61.76	63.82	79.71	90.69
7	53.72	57.50	75.28	63.39	56.89	69.83	<b>81.11</b>
8	50.90	49.55	72.59	62.58	51.11	61.14	<b>81.20</b>
9	50.60	45.73	69.72	61.42	50.13	60.94	<b>84.02</b>
10	50.20	53.38	<b>91.99</b>	61.84	49.97	66.61	91.42
11	50.03	46.90	77.12	62.04	49.97	60.92	<b>91.15</b>
12	49.85	50.58	<b>94.64</b>	62.09	50.03	57.85	84.25
13	50.23	33.60	78.39	61.92	50.03	58.06	<b>85.54</b>
14	49.85	49.62	88.66	61.91	50.29	63.51	<b>96.61</b>
15	50.27	35.84	70.58	61.79	49.90	54.94	<b>84.29</b>
16	50.06	51.36	82.05	61.98	50.06	63.42	<b>85.83</b>
rank	5	7	2	4	6	3	<b>1</b>

Table 5 records the average DRR of all methods running 30 times on 16 datasets. We can see that MSA-MPSSA performs best in 11 out of 16 datasets, ranking first. The second-ranked method is IWOSSA, and in its best-performing datasets, only Breast3 approximates the proposed method in terms of accuracy. For the other datasets Horse, flags, ionosphere and warpPIE10P, MSA-MPSSA is significantly better than IWOSSA in accuracy, which can indicate the MSA-MPSSA has achieved a good balance in dimensionality reduction and accuracy, and can better select the informative features. In addition, except for the dataset HeartEW, MSA-MPSSA achieves a dimensionality reduction rate of over 80% in all datasets, and six datasets are above 90%. These demonstrate the significant role of the proposed method in feature dimensionality reduction.

#### 4.4. Effect of Multi-Population Adaptive Generation Strategy and F-P Operator

One of the main ideas behind the proposed MSA-MPSSA is to generate new independent sub-populations when the main population falls into a local optimum, and use the F-P

operator and multi-surrogate to find promising regions for the sub-populations to search. To evaluate the effectiveness of the multi-population adaptive generation strategy and the F-P operator, we set up two methods named MSA-MPSSA-F and MSA-MPSSA-R. The former represents the generation of all sub-populations using the F-P operator and multi-surrogate at the first iteration. The latter refers to discarding the F-P operator when generating sub-populations, and randomly generating candidate pools.

We ran the three methods independently 30 times and recorded the average CBA and average DRR in Table 6. When compared with MSA-MPSSA-F, we observed that MSA-MPSSA performs better in CBA on all datasets, and DRR is better on almost all datasets. This indicates that the multi-population adaptive generation strategy is significantly better than the conventional multi-population approach and is able to continuously improve the population to avoid local optima. Furthermore, the proposed method outperforms MSA-MPSSA-R on most of the datasets, demonstrating that the F-P operator can help new sub-populations find more promising regions for exploration.

Table 6: Effects of multi-population adaptive generation strategy and F-P operator

No.	MSA-MPSSA-F		MSA-MPSSA-R		MSA-MPSSA	
	CBA	DDR	CBA	DDR	CBA	DDR
1	0.8847±0.004	<b>76.92</b>	0.8854±0.005	<b>76.92</b>	<b>0.8873±0.005</b>	76.41
2	0.8423±0.015	79.39	<b>0.8433±0.011</b>	80.00	0.8410±0.013	<b>80.76</b>
3	0.8394±0.010	91.73	0.8414±0.009	<b>91.98</b>	<b>0.8449±0.013</b>	<b>91.98</b>
4	0.6316±0.020	<b>86.43</b>	0.6271±0.027	85.95	<b>0.6360±0.025</b>	84.40
5	0.9628±0.003	92.56	0.9616±0.004	92.67	<b>0.9649±0.003</b>	<b>93.00</b>
6	0.9378±0.006	91.37	0.9372±0.006	<b>91.47</b>	<b>0.9400±0.006</b>	90.69
7	0.9068±0.010	79.50	<b>0.9108±0.010</b>	80.44	0.9107±0.013	<b>81.11</b>
8	0.9511±0.009	80.96	0.9477±0.011	<b>81.30</b>	<b>0.9545±0.012</b>	81.20
9	0.7081±0.014	82.06	0.7059±0.013	83.46	<b>0.7169±0.015</b>	<b>84.02</b>
10	0.9770±0.004	89.95	0.9768±0.004	91.34	<b>0.9774±0.004</b>	<b>91.42</b>
11	0.9712±0.009	89.70	0.9704±0.007	90.09	<b>0.9721±0.009</b>	<b>91.15</b>
12	0.7476±0.018	83.52	0.7503±0.019	83.31	<b>0.7539±0.013</b>	<b>84.25</b>
13	0.8555±0.015	82.28	0.8539±0.011	84.78	<b>0.8602±0.016</b>	<b>85.54</b>
14	0.9997±0.001	95.88	<b>1.0000±0.000</b>	96.56	<b>1.0000±0.000</b>	<b>96.61</b>
15	0.6805±0.023	83.03	<b>0.6911±0.021</b>	83.76	0.6809±0.026	<b>84.29</b>
16	0.9784±0.008	84.28	0.9808±0.007	85.78	<b>0.9825±0.007</b>	<b>85.83</b>

#### 4.5. Running Time

To further investigate the performance of the proposed MSA-MPSSA, we compared the running time and CBA of the 7 methods, and the results are shown in Figure 3. The dataset used covers the number of features from small to large.

As depicted in Figure 3, MSA-MPSSA consistently achieves the best CBA with a smaller running time, and reaches the optimal tradeoff between conflicting objectives. Although IWOSSA is the fastest method, it is slower than MSA-MPSSA on the dataset lymphoma2. This is due to the fact that MSA-MPSSA selects a smaller feature subset,

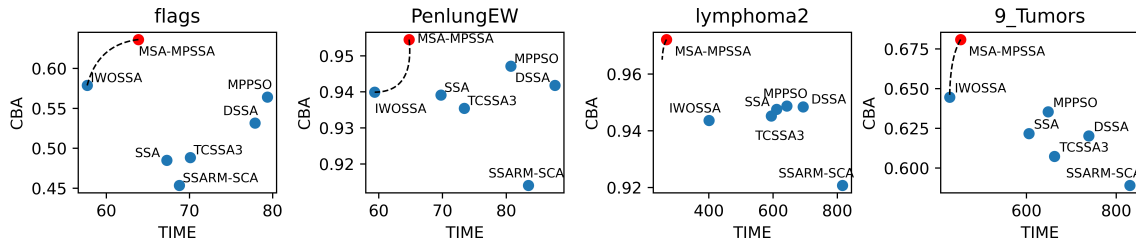


Figure 3: Comparison of classification balance accuracy and running time(s). The dotted line represents the optimal trade-off between the two conflicting objectives.

reducing the amount of data processed during the run. On the other hand, DSSA and SSARM-SCA are the slowest because their proposed local search algorithms add a lot of expensive real fitness evaluation.

## 5. Conclusion

In this paper, we propose a novel SSA-based feature selection method, namely MSA-MPSSA. Firstly, an initialization method based on feature importance and dataset size is suggested to improve the initial main population. A multi-population adaptive generation strategy is developed to continuously improve the main population trapped in local optima. To the best of our knowledge, no one has investigated generating new independent populations during the search process. To find promising search regions with a small time cost, F-P operation, and multi-surrogate are proposed. Finally, a communication mechanism is presented to help sub-populations to learn from each other.

We compare MSA-MPSSA with basic SSA and 5 state-of-the-art feature selection methods on 16 datasets. The results show that the MSA-MPSSA achieves best classification performance and is able to select a smaller feature subset in most cases. In addition, its time cost is usually smaller.

In the future, we will further study the generation timing and scale of new populations. Additionally, we will explore the applicability of the multi-population adaptive generation strategy to other metaheuristic methods. Our focus will be on improving the performance on high-dimensional datasets.

## Acknowledgments

This work received support from the National Science Foundation of China (No.61472095) and the Natural Science Foundation of Heilongjiang Province under Grant LH2020F023.

## References

Ibrahim Aljarah, Majdi Mafarja, Ali Asghar Heidari, Hossam Faris, Yong Zhang, and Seyedali Mirjalili. Asynchronous accelerating multi-leader salp chains for feature selection. *Applied Soft Computing*, 71:964–979, 2018.

- Urvesh Bhowan, Mark Johnston, and Mengjie Zhang. Developing new fitness functions in genetic programming for classification with unbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):406–421, 2011.
- Xing Cai, Huimin Zhao, Shifan Shang, Yongquan Zhou, Wu Deng, Huayue Chen, and Wuquan Deng. An improved quantum-inspired cooperative co-evolution algorithm with multi-strategy and its application. *Expert Systems with Applications*, 171:114629, 2021.
- Ke Chen, Bing Xue, Mengjie Zhang, and Fengyu Zhou. Correlation-guided updating strategy for feature selection in classification with surrogate-assisted particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 26(5):1015–1029, 2021.
- Hongbin Dong, Jing Sun, Xiaohang Sun, and Rui Ding. A many-objective feature selection for multi-label classification. *Knowledge-Based Systems*, 208:106456, 2020.
- Huachao Dong, Peng Wang, Weixi Chen, and Baowei Song. Sgop: Surrogate-assisted global optimization using a pareto-based sampling strategy. *Applied Soft Computing*, 106:107380, 2021.
- Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>, 7(1), 2017.
- Mahmoud Said El Sayed, Nhien-An Le-Khac, Marianne A Azer, and Anca D Jurcut. A flow-based anomaly detection approach with feature selection method against ddos attacks in sdns. *IEEE Transactions on Cognitive Communications and Networking*, 8(4):1862–1880, 2022.
- Hossam Faris, Ali Asghar Heidari, Al-Zoubi Ala’M, Majdi Mafarja, Ibrahim Aljarah, Mohammed Eshtay, and Seyedali Mirjalili. Time-varying hierarchical chains of salps with random weight networks for feature selection. *Expert Systems with Applications*, 140:112898, 2020.
- Ali Hosseinalipour and Reza Ghanbarzadeh. A novel metaheuristic optimisation approach for text sentiment analysis. *International journal of machine learning and cybernetics*, 14(3):889–909, 2023.
- Utkarsh Mahadeo Khaire and R Dhanalakshmi. Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences*, 34(4):1060–1073, 2022.
- Fatih Kılıç, Yasin Kaya, and Serdar Yildirim. A novel multi population based particle swarm optimization for feature selection. *Knowledge-Based Systems*, 219:106894, 2021.
- Lingjie Li, Manlin Xuan, Qiuzhen Lin, Min Jiang, Zhong Ming, and Kay Chen Tan. An evolutionary multitasking algorithm with multiple filtering for high-dimensional feature selection. *IEEE Transactions on Evolutionary Computation*, 2023.
- Shulei Liu, Handing Wang, Wei Peng, and Wen Yao. A surrogate-assisted evolutionary feature selection algorithm with parallel random grouping for high-dimensional classification. *IEEE Transactions on Evolutionary Computation*, 26(5):1087–1101, 2022.

- Yun Liu, Yanqing Shi, Hao Chen, Ali Asghar Heidari, Wenyong Gui, Mingjing Wang, Huiling Chen, and Chengye Li. Chaos-assisted multi-population salp swarm algorithms: Framework and case studies. *Expert Systems with Applications*, 168:114369, 2021.
- Seyedali Mirjalili, Amir H Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, 114:163–191, 2017.
- Mohammed H Qais, Hany M Hasanien, and Saad Alghuwainem. Transient search optimization: a new meta-heuristic optimization algorithm. *Applied Intelligence*, 50:3926–3941, 2020.
- Mohammed Qaraad, Souad Amjad, Nazar K Hussein, and Mostafa A Elhosseini. An innovative quadratic interpolation salp swarm-based local escape operator for large-scale global optimization problems and feature selection. *Neural Computing and Applications*, 34(20):17663–17721, 2022.
- Litao Qu, Weibin He, Jianfei Li, Hua Zhang, Cheng Yang, and Bo Xie. Explicit and size-adaptive pso-based feature selection for classification. *Swarm and Evolutionary Computation*, page 101249, 2023.
- Mahmoud M Saafan and Eman M El-Gendy. Iwossa: An improved whale optimization salp swarm algorithm for solving optimization problems. *Expert Systems with Applications*, 176:114901, 2021.
- Warda M Shaban. Insight into breast cancer detection: new hybrid feature selection method. *Neural Computing and Applications*, 35(9):6831–6853, 2023.
- Lin Sun, Shanshan Si, Jing Zhao, Jiucheng Xu, Yaojin Lin, and Zhiying Lv. Feature selection using binary monarch butterfly optimization. *Applied Intelligence*, 53(1):706–727, 2023.
- Yanan Sun, Handing Wang, Bing Xue, Yaochu Jin, Gary G Yen, and Mengjie Zhang. Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 24(2):350–364, 2019.
- Mohammad Tubishat, Salinah Ja’afar, Mohammed Alswaitti, Seyedali Mirjalili, Norisma Idris, Maizatul Akmar Ismail, and Mardian Shah Omar. Dynamic salp swarm algorithm for feature selection. *Expert Systems with Applications*, 164:113873, 2021.
- Peng Wang, Bing Xue, Jing Liang, and Mengjie Zhang. Feature selection using diversity-based multi-objective binary differential evolution. *Information Sciences*, 2023.
- Miodrag Zivkovic, Catalin Stoean, Amit Chhabra, Nebojsa Budimirovic, Aleksandar Petrovic, and Nebojsa Bacanin. Novel improved salp swarm algorithm: An application for feature selection. *Sensors*, 22(5):1711, 2022.