A Novel Multi-objective Squirrel Search Algorithm: MOSSA

Xinyuan Wang, Fanhao Zhang, Zhuoran Liu(✉), Changsheng Zhang, Qidong Zhao, and Bin Zhang

Northeastern University, Shenyang 110819, People's Republic of China
paper820@sohu.com

Abstract. This paper suggests a non-dominated sorting genetic algorithm II (NSGA-II) as a multi-objective framework to construct a multi-objective optimization algorithm and uses the squirrel search algorithm (SSA) as the core evolution strategy. And a multi-objective improved squirrel search algorithm (MOSSA) is proposed. MOSSA establishes an external archive of the population to maintain the elitists in the population. The probability density is applied to limit the size of the merged population to maintain population diversity, based on roulette wheel selection. Also, this paper designs a fitness mapping evaluation according to the individual fitness value of each object. Compared with the original SSA, the generational gap is introduced to make the seasonal condition suitable for multi-objective optimization, which could keep the solution from the local convergence. This paper simulates MOSSA and other algorithms on multi-objective test functions to analyze the convergence and diversity of PF. It is concluded that MOSSA has a good performance in solving multi-objective problems.

Keywords: Multi-objective optimization · Non-dominated sorting · Squirrel search algorithm · Mapping fitness evaluation · Roulette wheel selection

1 Introduction

SSA is a novel natural heuristic optimization paradigm called squirrel search algorithm [1, 2] to simulate the flying squirrel seek for food on trees. It was proposed by Jain M, Singh V, Rani A, and others after studying the foraging behavior of southern flying squirrels in 2019. The algorithm was tested with several classic and modern unconstrained benchmark functions. Compared with other optimization algorithms reported in the literature, the SSA algorithm has significant convergence. Besides, for advanced highly complex CEC 2014 benchmark functions [3], SSA has the same good convergence. SSA provides quite competitive results on both numerical optimization and real-time problems.

However, many optimization problems encountered in reality are multi-objective problems (MOPs), such as communication engineering [3], transportation problems [5, 6], power systems [7] and other many fields [8]. Many complicated problems could be simulated to mathematical questions and solved efficiently [9, 10]. All the scenarios can be

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 370, pp. 180–195, 2021.
https://doi.org/10.1007/978-3-030-72795-6_15

simulated with computer methods to be multi-objective optimization problems mathematically [11, 12]. Usually, the sub-goals of multi-objective optimization problems are contradictory to each other. Improvements in one sub-goal may lead to degraded performance in another or other several sub-goals. In other words, it is impossible to make all the multiple sub-objectives reach the optimal value at the same time. You can only coordinate and compromise between them to optimize each sub-goal as much as possible. The essential difference between it and the single-objective optimization problem is that its solution is not unique, that is, there exists a set of optimal solution sets composed of many Pareto optimal solutions. Each element in the set is called a Pareto optimal solution or a non-suboptimal optimal solution.

There is no unique global optimal solution for multi-objective optimization problems. Too many non-inferior solutions cannot be directly applied, so it is necessary to find a final solution when solving. There are three main methods to find the final solution at this stage [11, 12]:

1. Decomposition method: Convert to a single-objective problem decomposition method based on the relative importance between objectives given by the decision-maker in advance.
2. Interaction method: The final solution is gradually obtained through the interaction between the analyst and the decision-maker.
3. Generating method: Find a large number of non-inferior solutions and then obtain the final solution according to the decision maker's intention.

Many experts and scholars have applied different algorithms to solve multi-objective optimization problems, such as multi-objective evolutionary algorithms [11], multi-objective particle swarm optimization (MOPSO) [13], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [14], a nondominated sorting genetic algorithm II (NSGA-II) [15] and many other algorithms.

Because the SSA algorithm has an efficient search capability, it is beneficial to obtain the optimal solution in the sense of multiple objectives. The algorithm searches for many non-inferior solutions by representing the total number of solution sets, that is, searching for many Pareto optimal solutions. At the same time, SSA is more versatile, suitable for processing various types of objective functions and constraints, and easy to combine with traditional optimization methods, thereby improving its limitations and solving problems more effectively. Therefore, the application of SSA to multi-objective optimization problems has excellent advantages.

Therefore, how to design a single-objective SSA for optimizing multi-objective problems has become a research hotspot. This paper proposes a multi-objective squirrel search algorithm (MOSSA). In the process of expanding a single target to multiple targets, the following challenges are encountered: In the single target optimization process, due to the characteristics of the single target, the global optimal solution and the suboptimal solution can be selected relatively easily. In MOSSA, there are multiple mutually restricted objects. Individuals cannot simply determine a learning sample by comparing a single target. Therefore, how to judge the fitness value of an individual is a key step of the MOSSA algorithm. Besides, since multi-objective optimization problems usually have a set of non-inferior solutions, how to choose the optimal solutions from a variety of non-inferior solutions becomes a huge challenge.

This paper applies the following methods from different perspectives to design an efficient and novel multi-objective algorithm, MOSSA. The contributions of this paper are outlined as follows:

- An external archive of the population is established to reserve the elites in the population. After merging two continuous populations, the probability density is applied to limit the population size and improve the extension of distribution.
- The original operation of calculating the crowded distance in NSGA-II may cause all the dense solutions to be filtered out at one time, and meanwhile some solutions which can be used to maintain diversity are accidentally deleted. Therefore, roulette wheel selection is introduced to make the dense grids have a higher probability of deletion, rather than necessarily deletion.
- Because multiple objects are restricted to each other in multi-objective optimization, individuals cannot be simply sorted by the fitness values. This paper uses the Pareto level and grid density to construct a mapping strategy to calculate the fitness values of individuals.
- The original solution to avoid the local convergence in SSA, seasonal condition, is no longer applicable in multi-objective optimization problems. The seasonal condition in MOSSA is improved by generational gap. It prevents MOSSA from the local convergence in multi-objective optimization problems and enhances the spread of PF.
- MOSSA is simulated on the Zitzler Deb Thiele series test functions [16] in order to compare with NSGA-II, MOPSO, MOEA/D. This paper analyzes the convergence, uniformity, and spread of all the algorithms through several indicators and PF. Experimental results reveals that MOSSA has excellent performance and is an efficient multi-objective optimization algorithm.

The rest of this paper is organized as follows. Section 2 shows the related work. And Sect. 3 presents the concept of the algorithm design. Experimental results analysis is concluded in Sect. 4. At last, Sect. 5 shows the conclusion of this paper.

2 Related Work

From the current research of multi-objective optimization by experts [17], the main task of solving multi-objective optimization problems is to find the Pareto optimal solution set. This solution set can weigh multiple objective functions and achieve 3 goals [18–21]. In general, some evaluation indexes can be used to reflect it.

1. Convergence of the solution set is used to evaluate the distance between the solution obtained by the algorithm and the real Pareto front is minimized. Generally, the obtained solution set is required to make the convergence as small as possible.
2. Uniformity of the solution set is used to evaluate the uniformity and evenness of the individual distribution. Generally, each solution in the obtained solution set should be distributed as uniformly as necessary.

3. Spread of the solution set is used to evaluate the level of the entire obtained solution set distributed in the target space. Generally, the solution set should be as wide as necessary to show the Pareto Front as completely as necessary.

These goals could be achieved through different algorithms. Many various methods are applied to solve multi-objective optimization problems more efficiently. Multi-objective evolutionary algorithm is a kind of global probabilistic optimization search method formed by simulating biological evolution mechanisms [11]. And a multi evolutionary algorithm based on decomposition (MOEA/D) was proposed [14], so multi-objective optimization problem is decomposed into multiple scalar optimization sub-problems. Multiple objective particle swarm optimization (MOPSO) was presented to use Pareto dominance to decide the next direction of swarm [13]. There are two main methods [22], namely, methods that are not based on Pareto optimization and methods that are based on Pareto optimization. On this basis, some scholars have proposed the concept of external archive [23]. The external archive sets save all the non-dominated individuals of the current generation so that the solution set maintains a good distribution. A multi-objective evolutionary algorithm with an external set is put more emphasis on the efficiency and effectiveness of the algorithm [11]. The more typical multi-objective evolutionary algorithms are NSGA-II [15], PESA2 [24], and SPEA2 [25]. NSGA-II took the nondominated sorting into a multi-objective optimization algorithm. The advantage of PESA2 is that its solution converges very well, and it is easier to approach the optimal surface, especially in the case of high-dimensional problems; but its disadvantage is that the selection operation can only select one individual at a time, which consumes much time and has a class The diversity is poor. The advantage of SPEA2 is that it can obtain a well-distributed solution set, especially for solving high-dimensional problems, but its clustering process takes a long time to maintain diversity and is not efficient.

At present, the demand for multi-objective optimization algorithms has become more extensive, not only in real life but in solving the processes of many algorithms, many multi-objective optimization problems are waiting to be solved. There are more multi-objective optimization algorithms that have also been successfully used in function optimization [26], neural network training [27], pattern classification [28], fuzzy system control [29], and other application areas.

3 Background

3.1 Squirrel Search Algorithm

When the squirrel begins to forage, the search process begins. During this time, they began to migrate and explore various forested areas. Squirrels form their own migration routes based on the fitness of their companions. As the weather changes, they adjust their foraging strategies to increase the likelihood of survival. This foraging strategy runs through the entire life of each squirrel (Fig. 1).

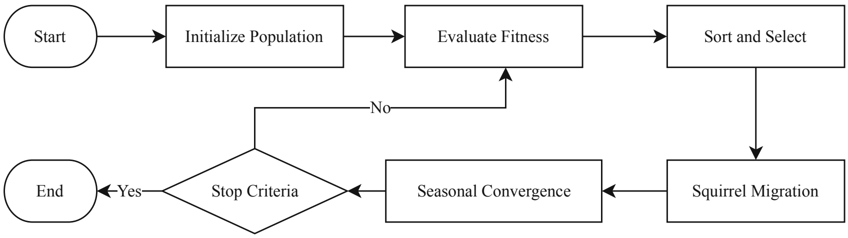

Fig. 1. A procedure of Squirrel Search Algorithm (SSA).

3.2 Basic Concept of NSGA-II

In the process of NSGA-II, fast nondominated sorting is a vital method to take advantage of sorting solutions with various Pareto dominance levels. Before the description of our design, some concepts of NSGA-II which are also applied in this paper are listed here.

Pareto dominance: if and only if $SF_{i,k} \leq SF_{j,k}$, and $SF_{i,k} < SF_{j,k}, k = 1, 2, \ldots, m$.

In this case, we say that S_i (i^{th} squirrel) dominates S_j or S_i being dominated by S_i, is written as $S_i \leq S_j$.

Non-dominated individuals: Individuals are non-dominated individuals in the population, if and only if there is no individual S_j, satisfying S_j dominates S_i.

Pareto Front (PF): The Pareto front is a hyperplane in the target space that is fitted by the optimal solution set of a theoretical optimization problem. In experimental research, the Pareto front is often used to represent the problem by a set of known non-dominated solutions. D_i a set of solutions that the solution S_i dominates.

4 Design of MOSSA

4.1 Basic Concept of MOSSA

To improve the performance of the multi-objective evolutionary algorithm NSGA-II in solving multi-objective optimization problems, this paper uses NSGA-II as a multi-objective optimization algorithm, combined with the efficient search strategy of the squirrel search algorithm (SSA), a pseudo-code of this multi-objective improved squirrel search algorithm (MOSSA) as shown in Algorithm 1.

First, based on the original SSA, a random initialization method suitable for multi-objective problems is designed. During the iteration process, an external archive of the population reserves the elites in the population. Then, MOSSA sorts non-dominantly the population and the generational gap is calculated for the seasonal condition. Multiple objectives are restricted to each other in multi-objective optimization, individuals cannot be simply sorted by the fitness values. In that case, MOSSA uses the Pareto level and grid density to construct a mapping function to calculate the fitness values of individuals. Squirrels then migrate according to the strategy in SSA. MOSSA uses novel seasonal condition which could avoid the local convergence. After ensuring that it meets the seasonal change conditions which means it is winter, the normal squirrels fly following the Levy flight. Two continuous populations should be merged into the archive as the current population, the probability density is applied to limit the population size and improve the distribution of PF. Finally, the feasible solution set is obtained.

Algorithm 1. Pseudocode for MOSSA

```
Input:
  Population number, test problem, parameter definition
Output
  Feasible solution set
Begin
  initialized population randomly;
  while stop criteria==false do
      archive current population as set R;
      non-dominated sort and calculate grid density (DG_i);
      obtain mapping fitness based on R and grid density (DG_i);
      squirrels migrate to form new population set T;
      calculate SC and determine season;
      if season==winter then
          normal squirrels Levy flight;
      else
          continue;
      end if
      archive current population as set T';
      merge two populations R and T';
  end while
  return feasible solution set;
End
```

4.2 Random Initialization

The position of all squirrels (SP) can be represented by the following matrix [1], where $SP_{i,j}$ represents the j^{th} dimension (d1 in total) of i^{th} squirrel. The objective function value of all squirrels (SF) can also be represented by the following matrix [2], where $SF_{i,k}$ represents the k^{th} objective function value (if dimension is d2 in total) of i^{th} squirrel.

$$SP_{i,j} = \begin{bmatrix} SP_{1,1} & SP_{1,2} & \cdots & \cdots & SP_{1,d1} \\ SP_{2,1} & SP_{2,2} & \cdots & \cdots & SP_{2,d1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ SP_{n,1} & SP_{n,2} & \cdots & \cdots & SP_{n,d1} \end{bmatrix}$$

Matrix [1]: decision space

$$SF_{i,k} = \begin{bmatrix} SF_{1,1} & SF_{1,2} & \cdots & \cdots & SF_{1,d2} \\ SF_{2,1} & SF_{2,2} & \cdots & \cdots & SF_{2,d2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ SF_{n,1} & SF_{n,2} & \cdots & \cdots & SF_{n,d2} \end{bmatrix}$$

Matrix [2]: target space

A uniform distribution is used to assign the initial position of each squirrel.

$$SP_i = SP_{min} + U(0,1) \times (SP_{max} - SP_{min}) \tag{1}$$

Where U (0, 1) is a uniformly distributed random number in the range [0, 1] and SP_{min}, SP_{max} are lower and upper limits of i^{th} squirrel in j^{th} dimension.

4.3 Non-dominated Sorting

The pseudo-code of the non-dominated sorting part of MOSSA which the concept of the fast-nondominated-sorting [15] in NSGA-II is adapted to is presented in Algorithm 2. The current unsorted population P is sorted by Pareto levels. After processing of this section, the pareto rank levels of solutions are obtained. Each candidate solution is detected whether it is dominated by other individuals in the population. The solutions which are not dominated by any others are marked as nondominated solutions with the least level.

Algorithm 2. Fast-nondominated-sorting

Input: Unsorted population P
Output: All pareto rank levels L_i (i^{th} level)
Begin
for each $p \in P$
  for each $q \in P$
  if $p \preccurlyeq q$ **then**
    $D_p = \{p\} \cup D_p$;
    $DS_p = DS_p + 1$; //DS_p means how many solutions dominated p
  else if $q \preccurlyeq p$ **then**
    $n_p = n_p + 1$; //n_p means the number of solutions dominating p
  end if
if $n_p = 0$ **then** //if no solution dominates p then
    $L_1 = \{p\} \cup L_1$;
 end if
  $i = 1$;
while $L_i \neq \emptyset$
  $TEM = \emptyset$;
  for each $p \in L_i$
    for each $q \in D_p$
      $n_p = n_p - 1$;
      if $n_p = 0$ **then** $TEM = \{p\} \cup TEM$
end while
$i = i + 1$;
$L_i = TEM$;
End

4.4 Grid Division

If a multi-objective problem has m objective functions, then it constitutes an m-dimensional target space. In order to make the population more diverse, we mesh the target space. That is, this target space is divided into $K1 \times K2 \times \ldots \times Ki \times \ldots \times Km$ grids, and the grid width of the i^{th} target(GW_i) of each grid is:

$$GWi = (SFmaxi - SFmini)/Ki \quad (2)$$

Where Ki is the number of grids which the i^{th} dimension objective function is divided by and SF^i_{max} and SF^i_{min} are the maximum and minimum value of objective function on

i^{th} dimension. Now we can calculate the density of each grid (DG_i) which means the number of individual in this certain grid and the grid coordinates of the i^{th} $target(GC_i)$.

$$GC_i = (SF_i - SF_{min})/GW_i \tag{3}$$

4.5 Fitness Mapping Function Design

After the target space is divided into several grids, the fitness value mapping function of each squirrel needs to be determined according to two indicators:

Grid density (DG_i): The number of squirrels corresponding to the grid which contains the i^{th} squirrel in the target space.

Dominant strength (DS_i): The number of other squirrels dominated by the i^{th} squirrel.

Therefore, the i^{th} squirrel's mapping fitness (MF_i) function can be defined as the k^{th} evolution process as:

$$MF_i^k = DG_i^k/(DS_i^k + 1) \tag{4}$$

Where DG_i^k is obtained by counting the number of particles with the same coordinate, DS_i^k represents the number of other particles dominated by the particle i in the k^{th} iteration, which can be obtained by the definition of domination. Adding 1 is to prevent the denominator from being zero. It can be seen that the more a particle controls other particles, the better the fitness, and the better the position of the particle, the healthier the particle is, the smaller the fitness obtained. The purpose of this definition is to obtain particles that are close to the real Pareto front end, with uniform distribution and good scalability.

The population is sorted according to the mapped fitness values, and the squirrels at the 3 optimal positions (SP_{best}) with the smallest fitness values are selected, and the squirrels at the 9 sub-optimal positions (SP_{sub}) with the smaller fitness values are selected. The default population in the algorithm is 100, so there are 88 ordinary solutions (SP_{other}) left.

4.6 Squirrel Migration

According to squirrel habits, we think that squirrels will begin to migrate when their natural enemies are not present. Suppose that the probability of the appearance of natural enemies is P, so that squirrels whose random numbers fall between [P, 1] in the interval [0,1] can migrate so that random numbers that fall between [0, P] should be randomly hidden.

In the first case, sub-best squirrels will move towards best squirrels.

$$SP_{sub,j}^{new} = SP_{sub,j} + const \times \left(SP_{best,j} - SP_{sub,j}\right) \tag{5}$$

In the second case, normal squirrels will move towards sub-best squirrels.

$$SP_{other,j}^{new} = SP_{other,j} + const \times \left(SP_{sub,j} - SP_{other,j}\right) \tag{6}$$

In the third case, some normal squirrels have already been sub-best squirrels, so they will move towards best squirrels.

$$SP_{other,j}^{new} = SP_{other,j} + const \times \left(SP_{best,j} - SP_{other,j}\right) \tag{7}$$

4.7 Seasonal Condition

In the single-objective squirrel algorithm, seasonal conditions are used to measure the degree of population aggregation during each iteration, but in the multi-objective squirrel search algorithm, we need to make some changes to apply to multi-objective optimization.

Generation $gap(G)$ is one of the classic convergence indicators. It is mainly used to describe the distance between the non-dominated solution obtained by the algorithm During two consecutive iterations. The smaller G is, the more likely the population is to fall into local convergence.

$$G = \frac{\sqrt{\sum_{i=1}^{n} gap_i}}{n} \tag{8}$$

Where n is the number of non-dominated solutions obtained by the algorithm, gap_i is the shortest Euclidean distance between the non-inferior solution and all the solutions in the new generation.

When the two populations reach seasonal constant (SC), which means winter is coming(G $\geq$ SC). On the contrary(G $<$ SC), it means that summer is coming.

$$SC = \frac{10E^{-6}}{(365)^{t/(t_m/2.5)}} \tag{9}$$

Where t means the current iteration number, t_m means maximum iteration number.

4.8 Levy Flight

Levy distribution can help the algorithm perform a global search in a better and more efficient way, and Levy flight helps the algorithm find new locations far from the current best location. Levy flight is a method of randomly changing the step size, where the step size is derived from the Levy distribution.

$$Levy = \frac{0.01 \times r_a \times \sigma}{|r_b|^{\frac{1}{\beta}}} \tag{10}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{11}$$

$$\Gamma(x) = (x-1)! \tag{12}$$

Where r_a and r_b are two normally distributed random numbers on the interval [0,1], $\beta = 1.5$. When the population reaches seasonal constant (SC), the ordinary individuals follow Levi's flight and update the squirrel's position.

$$SPi = SPmin + Levy \times (SPmax - SPmin) \tag{13}$$

4.9 Merge Population

After merging the two populations, because the size of new population exceeds the preset number of members of the population, corresponding screening is required. After non-dominated sorting and calculation of the grid density, the individual with the lowest Pareto rank is not what we required. The distribution of the obtained solutions is usually sparse and uneven. It is generally considered that those dense solutions are relatively poor in distribution, which need to be eliminated to ensure uniform solution distribution on the entire front. It is a problem to eliminate multiple individuals with high grid density at one time, which will probably cause one original dense grid empty. In that case, it makes the distribution of the Pareto solution worse. In response to this deficiency, this paper proposes the use of probability selection based on roulette wheel selection. Individuals with worst Pareto level are eliminated first. When the Pareto levels are the same, the relative density of the grids where these squirrels are located is regarded as the eliminating probability. First calculate the relative density (RD_j) of the j^{th} squirrel.

$$RD_i = DG_i / \sum_{i=1}^{m} DG_i \tag{14}$$

Where m means the number of squirrels which located on worst level. Squirrels of the same Pareto level divides a disc into m parts, in which the center angle of the j^{th} fan is $2\pi \cdot RD_j$. When making a selection, you can imagine turning the dial, and if the pointer falls into the j^{th} sector, delete the individual j. The implementation process is as follows: First generate a random number R in [0,1]. If $\sum_{j=1}^{j-1} RD_j \leq R < \sum_{j=1}^{j} RD_j$, then delete individual j. It can be seen that larger the central angle means more. More probably the individual will be eliminated to maintain population diversity.

5 Simulation Results Analysis

In this experimental simulation studies, this paper uses classic multi-objective optimization algorithms such as NSGA-II, MOPSO, and MOEA/D to compare with this algorithm. From the current research literature on multi-objective optimization problems, the evaluation indicators for multi-objective algorithms are mainly designed around convergence and diversity [11, 12, 18–21]. We use three indicators to evaluate the algorithm in this section, which are generational distance, spacing metric and diversity metric.

In the implementation process, the number of test instances was uniformly set to 100 and the number of iterations was 100 generations. Besides, non-dominated solutions obtained by these four algorithms are simulated on multi-objective optimization test functions (ZDT6, ZDTi, $i = 1 - 4$) [16], each experiment was executed 10 times, and the average values of evaluation parameters are represented to be compared.

5.1 Pareto Front Analysis

These 25 figures represent dominated solutions obtained using these four algorithms and also the ideal PF on a series of Zitzler-Deb-Thiele test functions in Fig. 2.

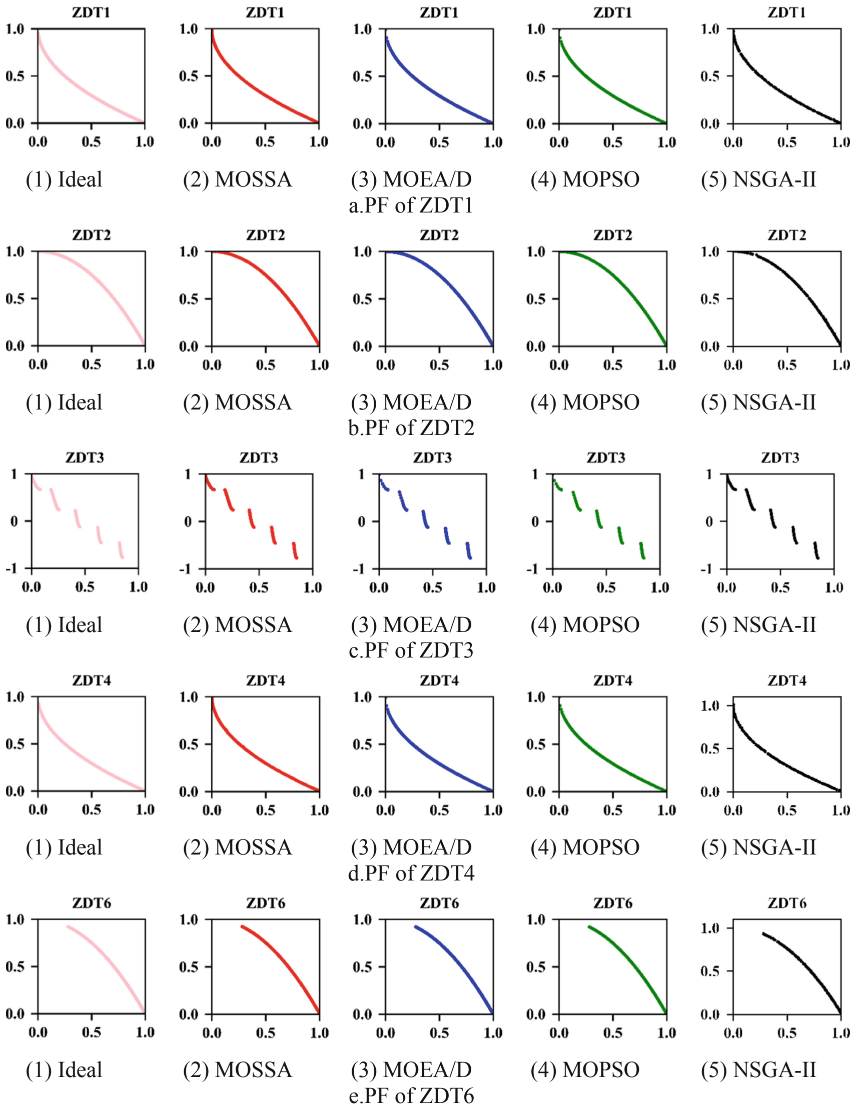

Fig. 2. PF of ZDT.

By comparing the ideal Pareto front and the experimentally obtained solution set, the solution set obtained by MOSSA almost coincides with the real Pareto front, which proves that MOSSA has good convergence. Because the differences in convergence from other algorithms is not obvious now, later in this paper we will also quantitatively compare the convergence of all algorithms.

It can be seen that the distribution of MOEA/D and MOPSO in extreme solutions is not as ideal as MOSSA, resulting in an uneven solution set. Although NSGA-II is relatively uniformly distributed, the distance between two consecutive solutions is larger than MOSSA. It can be seen that MOSSA is superior to the other three algorithms in terms of convergence, uniformity, and diversity on the ZDT test function.

Therefore, MOSSA solves the problem of the unsatisfactory distribution of MOEA/D and MOPSO at extreme solutions and solves the problem that NSGA-II has uniform and extensive solutions at the Pareto front, but the distance of continuous solutions is relatively longer. The uniformity and diversity of MOSSA in the non-dominated solution set are better than other algorithms.

5.2 Analysis with Generational Distance (GD)

Generational Distance (GD) is one of the classic convergence indicators [30]. It is mainly used to describe the distance between the non-dominated solution obtained by the algorithm and the real Pareto front end. The smaller the GD, the better the convergence.

Where n is the number of non-dominated solutions obtained by the algorithm, $dist_i$ is the shortest Euclidean distance between the non-inferior solution and all the solutions in the real Pareto front end.

$$GD = \frac{\sqrt{\sum_{i=1}^{n} dist_i}}{n} \tag{15}$$

For indicator GD, the smaller its value, the better the convergence of the algorithm. As shown in Table 1, every minimum value is bold for each column. It's observed from Table 1 that MOSSA is significantly better than the other three algorithms on ZDT1, ZDT4, and ZDT6 functions. However, for relatively poor performance on ZDT2, ZDT3 functions, we can also clearly see that MOSSA is the second best algorithm. The gap compared to the best algorithm is not obvious enough, so we can analyze from this indicator that the MOSSA algorithm has a good performance in the convergence of the multi-objective optimization algorithm.

Table 1. GD values of the solutions found by all the algorithms on all the test problems.

Algorithm	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MOSSA	**7.19E–04**	8.77E–4	0.004136	**0.001723**	**3.185E–4**
MOEA/D	9.62E–04	0.001494	**0.004013**	0.002115	8.30E–4
NSGAII	0.002306	0.00135	0.005052	0.006056	0.001584
MOPSO	9.07E–04	**7.85E–04**	0.004813	0.005592	7.24E–4

5.3 Analysis with Spacing Metric (SP)

Spacing Metric (SP) measures the standard deviation of the minimum distance from each solution to other solutions [31]. The smaller the Spacing value, the more uniform the solution set.

$$SP = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\bar{d} - d_i\right)^2}}{\bar{d}} \tag{16}$$

Where n is the number of non-dominated solutions obtained by the algorithm; di is the shortest Euclidean distance between the i^{th} solution and all solutions in the real Pareto front end. The smaller the SP, the better the distribution, and the better the diversity. When SP = 0, all the solutions in the Pareto solution set are uniformly distributed.

For indicator SP, the smaller value represents the better uniformity of the algorithm. As shown in Table 2, every minimum value is bold for each column. The data reveals that MOSSA has the smallest SP value on all of the ZDT series functions, which shows that MOSSA has a superior uniform performance.

Table 2. SP values of the solutions found by all the algorithms on all the test problems.

Algorithm	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MOSSA	**0.009245**	**0.001546**	**0.003498**	**0.002913**	**0.001289**
MOEA/D	0.016064	0.009078	0.03842	0.01566	0.008262
NSGAII	0.009456	0.013029	0.01517	0.01247	0.01048
MOPSO	0.015692	0.002360	0.01976	0.009101	0.001835

5.4 Analysis with Diversity Metric (DM)

Diversity Metric (DM) measures how extensive the solution set is [15].

$$DM = \frac{d_f + d_l + \sum_{i=1}^{n-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (n-1)\bar{d}} \tag{17}$$

Where d_f and d_l are the Euclidean distances between two extreme solutions of the true Pareto front and two boundary solutions of the non-dominated set obtained by experiments. And d_i is the Euclidean distance between two successive solutions in the non-dominated solution set. Assuming that the optimal non-dominated front has n solutions, then there are n − 1 consecutive distances, so $\bar{d}$ is the average of di ($\bar{d} = \frac{1}{n-1}\sum_{j=1}^{n-1} d_j$).

A uniform distribution makes all di approach $\bar{d}$, and when the distribution is extensive enough, $d_f = d_l = 0$ (there are extreme solutions in non-dominated sets). Therefore, for the most extensively and uniformly expanded non-dominated solution set, the numerator of DM will approach zero, making DM zero. For any other distribution, the value of the metric will be greater than zero. In other words, this evaluation parameter can measure both uniformity and breadth to achieve diversity. For two distributions with the same d_f and d_l values, DM has a higher value and a worse solution distribution in the extreme solution.

For the DM indicator, the smaller its value, the higher the diversity of the solution set generated by the multi-objective optimization algorithm. Every minimum value is bold for each column. At the same time, by analyzing data in Table 3, we can figure out that in all ZDT series functions MOSSA has excellent diversity. Although some gaps may not be obvious, the excellent diversity of MOSSA still can be observed.

Table 3. D values of the solutions found by all the algorithms on all the test problems.

Algorithm	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MOSSA	**0.6768**	**0.6841**	**0.7808**	**0.7072**	**0.6963**
MOEA/D	0.7524	0.8008	0.9317	0.7517	0.7495
NSGAII	0.7773	0.7951	0.7956	0.7877	0.8079
MOPSO	0.7529	0.7033	0.9549	0.7552	0.7084

The analysis of non-dominated solutions reveals that MOSSA can improve the extreme solutions and large gaps between consecutive solutions, resulting in a solution set which is uneven and diversity. Besides, three indicators which reflect the convergence, uniformity and diversity of multi-objective optimization algorithms also perform better in MOSSA then NSGA-II, MOPSO and MOEA/D.

6 Conclusion

This paper takes NSGA-II as the multi-objective framework and SSA as the main evolution strategy to construct a new improved multi-objective squirrel search algorithm (MOSSA). On their basis, this article has made several improvements. First, MOSSA established an external archive of the population to retain the elite individuals in the population. Moreover, after two consequent populations are merged, this paper suggests the grid density as an eliminated probability to limit population size, which guarantees the diversity of PF. Therefore, the concept of the grid density is introduced to further maintain the uniformity and the spread of the solution set. Based on SSA, the generational gap is introduced to improve the seasonal condition, which solves the problem that solutions may be trapped locally in multi-objective optimization. In addition, this paper designs a mapping function according to Pareto sorting level and grid density to calculate individual fitness values.

Finally, these algorithms MOSSA, MOEA/D, NSGA-II, and MOPSO are used to perform simulation experiments on a multi-objective test function set. This analysis presents the Pareto front and some indicators obtained by each algorithm on the test functions. Through quantitative analysis of GD, SP and DM indicators, it is observed that MOSSA performs well on the convergence of the solution set, the uniformity of solution distribution, and the spread of the distribution. It can be concluded that this novel multi-objective optimization algorithm, MOSSA provides a framework of SSA extended to multi-objective optimization problems and also has a satisfactory performance. The proposed algorithm, MOSSA, could solve MOPs in better performance, so that it could be applied to many fields in real life, such as the transportation, the finance, the engineering, and the biology. For example, vehicle routing programming problems, shopping tradeoff, network routing optimization problems, and protein-ligand docking problems.

References

1. Jain, M., Singh, V., Rani, A.: A novel nature-inspired algorithm for optimization: squirrel search algorithm. Swarm Evol. Comput. S2210650217305229 (2018)
2. Wang, Y., Du, T.: A multi-objective improved squirrel search algorithm based on decomposition with external population and adaptive weight vectors adjustment. Physica A: Stat. Mech. Appl. **542**, (2020)
3. Liang, J.J., Qu, B.Y., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 635 (2013)
4. Gunantara, N.. A review of multi-objective optimization: methods and its applications. Cogent Eng. **5**(1), 1502242 (2018)
5. Guo, Z., Liu, L., Yang, J.: A multi-objective memetic optimization approach for green transportation scheduling. In: 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), IEEE (2015)
6. Dai, M., Tang, D., Giret, A., Salido, M.A.: Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. Robot. Comput. Integrated Manuf. **59**, 143–157 (2019)
7. Zaro, F.R., Abido, M.A.: Multi-objective particle swarm optimization for optimal power flow in a deregulated environment of power systems. In: 2011 11th International Conference on Intelligent Systems Design and Applications, IEEE (2019)
8. Liu, Z., Jiang, D., Zhang, C., et al.: A novel fireworks algorithm for the protein-ligand docking on the AutoDock. Mob. Netw. Appl. 1–12, 53 (2019)
9. de Villiers, D.I., Couckuyt, I., Dhaene, T.: Multi-objective optimization of reflector antennas using kriging and probability of improvement. In: 2017 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting, pp. 985–986. IEEE, July 2017 (2007)
10. Delgarm, N., Sajadi, B., Kowsary, F., Delgarm, S.: Multi-objective optimization of the building energy performance: a simulation-based approach by means of particle swarm optimization (PSO). Appl. Energy **170**, 293–303 (2016)
11. von Lücken, C., Barán, B., Brizuela, C.: A survey on multi-objective evolutionary algorithms for many-objective problems. Comput. Optimization Appl. 1–50 (2014)
12. Cho, J.H., Wang, Y., Chen, R., et al.: A survey on modeling and optimizing multi-objective systems. IEEE Commun. Surv. Tutorials **19**(3), 1867–1901 (2017)
13. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimization for feature selection in classification: a multi-objective approach. Cybern. Trans. IEEE **43**(6), 1656–1671 (2013)
14. Zhang, Q., Li, H.: Moea/d: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2008)
15. Deb, K., Pratap, A., Agarwal, S., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
16. Mashwani, W.K., Salhi, A., Yeniay, O., et al.: Hybrid non-dominated sorting genetic algorithm with adaptive operators selection. Appl. Soft Comput. **56**, 1–18 (2017)
17. Li, K., Wang, R., Zhang, T., et al.: Evolutionary many-objective optimization: a comparative study of the state-of-the-art. IEEE Access **6**, 26194–26214 (2018)
18. Liu, Z., Zhang, C., Zhao, Q., et al.: Comparative study of evolutionary algorithms for protein-ligand docking problem on the AutoDock. International Conference on Simulation Tools and Techniques, pp. 598–607. Springer, Cham (2019)
19. Azzouz, R., Bechikh, S., Said, L.B.: Dynamic Multi-objective Optimization using Evolutionary Algorithms: A Survey. Recent Advances in Evolutionary Multi-objective Optimization, pp. 31–70. Springer, Cham (2017)

20. Bechikh, S., Elarbi, M., Said, L.B.: Many-objective Optimization using Evolutionary Algorithms: A Survey. Recent Advances in Evolutionary Multi-objective Optimization, pp. 105–137. Springer, Cham (2017)
21. Falcón-Cardona, J.G., Coello, C.A.C.: Indicator-based multi-objective evolutionary algorithms: a comprehensive survey. ACM Comput. Surv. (CSUR) **53**(2), 1–35 (2020)
22. Shamshirband, S., Shojafar, M., Hosseinabadi, A.A.R., Abraham, A.: A solution for multi-objective commodity vehicle routing problem by NSGA-II. International Conference on Hybrid Intelligent Systems. IEEE (2015)
23. Luo, G., Wen, X., Li, H., et al.: An effective multi-objective genetic algorithm based on immune principle and external archive for multi-objective integrated process planning and scheduling. The Int. J. Adv. Manuf. Technol. **91**(9–12), 3145–3158 (2017)
24. Gadhvi, B., Savsani, V., Patel, V.: Multi-objective optimization of vehicle passive suspension system using NSGA-II, SPEA2 and PESA-II. Procedia Technol. **2016**(23), 361–368 (2016)
25. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm. TIK-report, 103 (2001)
26. Wang, Y., Han, M.: Research on multi-objective multidisciplinary design optimization based on particle swarm optimization. In: 2017 Second International Conference on Reliability Systems Engineering (ICRSE). IEEE (2017)
27. Kaoutar, S., Mohamed, E.: Multi-criteria optimization of neural networks using multi-objective genetic algorithm. International Conference on Inteligent Systems & Computer Vision ISCV (2017)
28. Rosales-Perez, A., Garcia, S., Gonzalez, J.A., Coello, C.A.C., Herrera, F.: An evolutionary multiobjective model and instance selection for support vector machines with pareto-based ensembles. IEEE Trans. Evol. Comput. **21**(6), 863–877 (2017)
29. Juang, Chia-Feng., Jeng, Tian-Lu, Chang, Yu-Cheng: An interpretable fuzzy system learned through online rule generation and multiobjective ACO with a mobile robot control application. IEEE Trans. Cybern. **46**(12), 2706–2718 (2017)
30. Sheikholeslami, F., Navimipour, N.J.: Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance. Swarm and Evol. Comput. **35**, 53–64 (2017)
31. Tian, Y., Cheng, R., Zhang, X., et al.: Diversity assessment of multi-objective evolutionary algorithms: performance metric and benchmark problems [research frontier]. IEEE Comput. Intell. Magazine **14**(3), 61–74 (2019)