An improved multi-objective gravitational search algorithm

Yuhong Guo¹, Teng Luo and Hongqian Cao
School of Information and Control Engineering, Liaoning Petrochemical University, Fushun, 113001, China

¹Corresponding author’s e-mail: 1090832949@qq.com

Abstract. Multi-objective search algorithm is a common optimization tool to deal with complex multi-objective problems, such as Multiple Objectives Particle Swarm Optimization (MOPSO) and Non-dominated Sorting Genetic Algorithm-II (NSGA-II). Gravitational Search Algorithm (GSA) is a new heuristic evolutionary algorithm, which is based on the Newtonian gravity and the laws of motion to search optimal solutions. Some of agents have bigger mass, so other smaller mass agents are affected easily to fall into local optimization. In order to improve the search ability of the algorithm, this paper proposes an Improved Multi-Objective Gravitational Search Algorithm (IMOGSA). The proposed method uses the fast non-dominated sorting strategy and crowding distance of NSGA-II, and uses Sine Cosine Algorithm (SCA). Using strategy of NSGA-II is to reduce the complexity of the algorithm, in addition, using SCA is to improve the convergence and distribution of IMOGSA by improving the weight of acceleration. Finally, the proposed method has been compared with other well-known heuristic search methods by using some benchmark functions.

1. Introduction

Multi-objective optimization problem refers to deal with two or more optimization problems simultaneously, and their objective functions are contradictory. It is widely used in scientific research and practical engineering applications [1], such as workshop scheduling problem [2], resource allocation and environment [3]. The purpose of single objective optimization problem in calculation is to find out the global optimal solution, but multi-objective optimization problem is different. One solution is better for one of the objectives, however, it may be worse for the other targets. Therefore, a set of non-dominant solutions will be obtained in the calculation. At present, various heuristic optimization algorithms have been applied to multi-objective optimization. For example, Srinivas et al. [4] proposed Non dominated Sorting Genetic Algorithm-II (NSGA-II) and Coello et al. [5] proposed Multiple Objectives Particle Swarm Optimization (MOPSO). Both of them are classical multi-objective optimization algorithms.

Gravitational Search Algorithm (GSA) is a new heuristic search algorithm proposed by Rashedi et al. [6], and it is inspired by gravitational force between objects in universe. In GSA, the position of each object agent represents a feasible solution, and the mass of the object reflects that the position is better or worse. The larger one could have the better position and the smaller one has a worse position. Moreover, the motion mode of the object in solution space is determined by the gravitation of other objects. The larger mass of the object can get the smaller acceleration and the smaller velocity. In unit time, the larger one gets shorter distance under the same gravitation. On the contrary, the smaller mass can get larger velocity, acceleration, and the longer distance. Because GSA has the advantages of few parameters, better global search ability and fast convergence speed, GSA has been implemented in
many single objective optimization problems, and the effect is remarkable [7-11]. Therefore, Hassanzadeh et al. [12] combined GSA with MOPSO and proposed MOGSA. And Nobahari et al. [13] combined GSA with NSGA-II and proposed NSGSA. These algorithms are applied to solve multi-objective optimization problems. However, the research of the algorithm like MOGSA will easy to fall into local optimization because of the relative poor convergence [14]. Mirjalili [15] proposed Sine Cosine Algorithm (SCA) which is used to calculate the location of the current solution with multiple random and adaptive variables. And the algorithm can search different regions in the space to avoid local optimization effectively.

This paper proposes IMOGSA which uses the fast non-dominated sorting strategy and crowding distance of NSGA-II, and uses SCA. The proposed method uses the strategy of NSGA-II to reduce the complexity of the algorithm and uses SCA to improve the convergence and distribution by improving the weight of acceleration. Finally, using the benchmark function compares with other multi-objective optimization algorithms to illustrate the effectiveness of the algorithm.

2. Gravitational search algorithm

GSA is proposed by using the law of universal gravitation. Two objects in the universe can affect each other through gravitation. Individual can be regarded as a agent in the solution space, and each agent can be regarded as having mass. And the mass is an index to evaluate the agent better or worse. The position of the agent with large mass is regarded as a better solution, and these agents will affect other smaller agents. Then these smaller agents will move to the larger one. Moreover, every agents share their position information to others. After agents receive the information, they will find out better position, and move to this direction. In this way, agents can share information in the evolution, and the algorithm can search the better solution in the space.

Suppose that there are \( N \) particles in the space, then Eq.(1) is the position of the \( i \)th agent.

\[
X_i = (x^i_1, \ldots, x^i_d), \quad i = 1, 2, \ldots, N. \tag{1}
\]

The \( n \) is the dimension of the search space and \( x \) is the decision variable. \( x^i_j \) is the position of the \( i \)th agent in the \( j \)th dimension. The gravitational force between two particles can be expressed by Eq.(2).

\[
f^i_j(t) = G(t) \times \frac{M^i(t) \times M^j(t)}{R^i_j(t) + \varepsilon} \times (x^i_j(t) - x^j_i(t)) \tag{2}
\]

\( M^j(t) \) is the mass of \( j \)th particle and \( M^i(t) \) is the mass of \( i \)th particle. \( \varepsilon \) is an error, and it usually be expressed by a minimal constant. \( R^i_j(t) \) is the Euclidean distance between two particles in Eq.(3). \( G(t) \) is a gravitational constant in Eq.(4).

\[
R^i_j(t) = \| x^i_j(t), x^j_i(t) \| \tag{3}
\]

\[
G(t) = G_0 \times e^{-at/t} \tag{4}
\]

The particle mass is related to the fitness value. The larger one tends to the optimal solution closer. The larger mass particle can affect other smaller mass particles. The Eqs.(5) and (6) are aim to calculate the mass of a particle.

\[
m^i_j(t) = \frac{f^i_j(t) - f_{w_{best}}}{f_{best} - f_{w_{worst}}} \tag{5}
\]

\[
M^i_j(t) = \frac{m^i_j(t)}{\sum_{j=1}^{N} m^i_j(t)} \tag{6}
\]

\( f^i_j(t) \) is the fitness of agent \( i \) at time \( t \). \( f_{w_{worst}} \) is the worst value in particle fitness calculation. \( f_{best} \) is the best value in particle fitness calculation. Make an example to calculate minimum solution by Eqs. (7) and (8).

\[
f_{best} = \min f^i_j(t) \tag{7}
\]

\[
f_{w_{worst}} = \max f^i_j(t) \tag{8}
\]
Smaller particles are subject to acceleration of larger mass particles. The Eq. (9) is to calculate particle acceleration.

\[ a_i^k = \frac{f^k(t)}{M^k(t)} \]  

\( a_i^k \) is acceleration of particle \( i \), in \( k \) dimensional space. The acceleration of small mass particles are affected by larger particles. Eqs. (10) and (11) can update particle velocity and position in evolution.

\[ v_i^k(t+1) = rand_i \times v_i^k(t) + a_i^k(t) \]  

\[ x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \]  

3. Improved multi-objective gravitational search algorithm

3.1. Multi-objective optimization

Multi-objective optimization problems are usually described as Eqs. (12) and (13).

\[ \min f(x) = [f(x), i = 1, 2, \cdots, M] \]  

\[ \text{s.t. } g_j(x) \leq 0, j = 1, 2, \cdots, J \]  

\( f_i(x) \) is the fitness of the \( i \) particle objective function. \( x \) is the set of \( N \) alternative solutions in the space. The \( x \) is Eq. (14).

\[ x = \{x_1, \cdots, x_i, \cdots, x_N\} \]  

From Eqs. (12) and (13), \( M \) is objective function. \( g_j(\cdot) \) is a number of constraint condition.

In the multi-objective optimization problem, there is no unique global optimal solution. In the process of algorithm optimization, a group of non-dominated solutions can be obtained the dominant relationship between agents and they also be called Pareto optimal solution.

3.2. Fast non-dominated sorting and congestion

Fast non-dominated sorting is an improvement of NSGA sort strategy [4]. The strategy combines individual Pareto sequence value and individual crowding distance. Crowding distance is the distance of individuals around a given point in a population. In Figure 1, it shows the distance of individual crowding. From Figure 1, we can see that the crowding distance of individuals \( a, b \) and \( c \) is less than that of individuals I to VI. So the crowding distance of individuals \( a, b \) and \( c \) is smaller. The Pareto non-dominated sorting values of individuals I to VI are ranked as 1. The Pareto non-dominated sorting value of individuals from \( a \) to \( c \) is ranked as 2, and so on. In order to ensure the diversity of population distribution, considering the crowding distance in the range of individual, in Eq. (15) the Pareto ranking value of individual \( x \) is defined as the sum of the non-dominated sorting value of the individual dominated by the individual and the current non-dominated sorting value of the individual.

\[ r(x, g)_i = r(x, g) + \sum_{i=1}^{N} r(zx_i, g) \]  

\( r(x, g) \) express when iterate \( g \) times, individual \( x \) get the non-dominated sorting number. \( zx_i \) is the set of individuals dominated by individual \( x \) after \( g \) iterations. The Eq. (16) is crowding distance.

\[ i_d = \sum_{j=1}^{M} |f_j^{i+1} - f_j^i| \]  

\( i_d \) is congestion of \( i \). \( f_j^{i+1} \) is objective function \( j \) of \( i+1 \) point.
In this paper, Firstly, find all of individuals in the population, and store them in the current non-dominated solution set \( r(x,g) \). Search the individuals in the current non-dominated set and find the dominated individual set \( z_x \). The number of dominant individuals is stored in another \( H_1 \). The crowding distance of each individual is stored in the \( H_2 \). Before confirmed the individual level of the population, should rank the non-dominated solution. Compare the crowding distance at the same time to determine the correct of the individual non-inferior solution. Then, each individual selects the non-dominated solution with the ranking level of 1 as the better solution in the algorithm. Finally, update iteration and evolve.

### 3.3. Improve weight of acceleration

In the IMOGSA, suppose that there are two individuals \( a \) and \( b \), then the individual \( i \) in each solution space has two positions for two objective individuals. Firstly, calculate the fitness value \( f_i \) of each individual. Then using Eqs. (5) and (6) calculate the mass of each individual. Thirdly, acceleration is introduced in update evolution. Because of two targets, each individual will be affected in space. The universal gravitation is Eq.(17).

\[
F_a = G \times \frac{m_i \times m_k}{R^2}, \quad F_b = G \times \frac{m_i \times m_k}{R^2}, \quad a_i = \frac{F}{m_i} \quad (17)
\]

The acceleration of agent can affect another target of agents. And the acceleration is Eq.(18).

\[
a_i^h = G \times (f_a(x_i) - f_b(x_j)) \times (x_i - x_j) \times \frac{m_k}{\|x_i - x_j\|_2} \quad (18)
\]

The total acceleration of individual is Eq.(19) which is affected by two targets.

\[
a_i = \sum_{h=1}^{4} \omega_h a_i^h = \omega_1 a_i^1 + \omega_2 a_i^2 + \ldots + \omega_4 a_i^4, \quad i = 1, 2, \ldots, N \quad (19)
\]

According to Tariq et al. [15], \( \omega_h \) is a random number between \((0,1)\). But the randomly selected weights will not work well in the search. By adding SCA, it improved the weight of acceleration. It enhanced the individual motion mode of the algorithm and improved the convergence ability of the algorithm. Eqs.(20) and (21) are improved weight.

\[
a_i^{t+1} = \alpha_i^t + r_5 \sin(r_6) |r_5| \quad \|r_6\| < 0.5 \quad (20)
\]

\[
a_i^{t+1} = \alpha_i^t + r_5 \cos(r_6) |r_5| \quad \|r_6\| \geq 0.5 \quad (21)
\]

### 3.4. Improve algorithm process

The following is the algorithm steps of IMOGSA.

1. Initialize the population and generate \( N \) individuals randomly. Then calculate the objective function \( f_i(t) \) of each individual.

2. Calculate crowding distance and fast non-dominated sorting of individuals.

3. Using the improved acceleration, calculate the search direction of each individual \( a_i \).

4. Calculate new group \( x_i(t+1) \) by Eqs.(10) and (11).
(5) Compare new generation individuals with previous individuals and update the non-dominated solution set. Then delete the larger crowding individuals.

(6) Continue the iterative until the condition is met, then stop the calculation. But if is not meet, return to the third step.

Figure 2 is the flowchart diagram of IMOGSA.

![Flowchart Diagram of IMOGSA](image)

3.5. Numerical Computations

To evaluate a multi-objective algorithm, this paper use Generational Distance (GD) and Spacing (SP). And the following two criteria introduced in [16] are selected for further comparisons:

$$GD = \frac{1}{N} \left( \sum_{i=1}^{N} d_i^2 \right)^{1/2}$$

This criterion expresses the average distances of solutions from the true Pareto front of the problem under optimization. The lower this value becomes better and the known Pareto fits the real one.

$$SP = \frac{1}{N-1} \sum_{i=1}^{N} (\bar{d} - d_i)^{1/2}$$

This metric demonstrates how well the found solutions are uniformly distributed. And the lower this metric becomes the more solutions equitably spaced.

To evaluate IMOGSA, using the benchmark function [17] ZDT1 ZDT2 and ZDT4 to compared with the original classical multi-objective algorithm, MOPSO and NSGA-II. The benchmark function is as follows:

$$\min f = (f_1, f_2)$$

$$f_1 = \frac{1}{x_1^2 + x_2^2 + 1}$$

$$f_2 = x_1^2 + x_2^2 + 1$$

s.t. $x_1 \in [-3,3], x_2 \in [-5,5]$
Figure 3. Pareto optimal front of NSGA-II and IMOGSA.

Figure 4. Pareto optimal front of MOPSO and IMOGSA.

Table 1. The GD metric for different methods.

| Metric | Algorithm | GD   | NSGA-II | MOPSO |
|--------|-----------|------|---------|-------|
| Benchmark function[16] |    | 0.6803 | 1.6856 | 0.1694 |
| ZDT1   | IMOGSA    |      |         |       |
| ZDT2   | NSGA-II   |      |         |       |
| ZDT4   | MOPSO     |      |         |       |

Table 2. The SP metric for different methods.

| Metric | Algorithm | SP   | NSGA-II | MOPSO |
|--------|-----------|------|---------|-------|
| Benchmark function[16] |    | 3.4768 | 9.6098 | 1.0148 |
| ZDT1   | IMOGSA    |      |         |       |
| ZDT2   | NSGA-II   |      |         |       |
| ZDT4   | MOPSO     |      |         |       |

Figure 3 and Figure 4 are Pareto optimal front of IMOGSA compared with NSGA-II and MOPSO. As easily observed from Table 1 and Table 2, in some aspect, the result of IMOGSA is better than NSGA-II and MOPSO. The GD factor in the IMOGSA outperforms NSGA-II, and the SP factor has also been improved for the NSGA-II and MOPSO. Therefore, the distribution of IMOGSA is better than NSGA-II and MOPSO. However, when we compare IMOGSA with MOPSO in GD aspect, the efficiency of the proposed algorithm needs to be improved.

4. Conclusions

This paper introduces a new improved multi-objective gravitational search algorithm which uses fast non-dominated sorting strategy and SCA. It uses the strategy of NSGA-II to reduce the complexity of the algorithm and uses SCA to improve the convergence and uniform distribution by improving the weight of acceleration. In order to evaluate the proposed method, we have examined it on the standard benchmark functions and compared it with the classical multi-objective optimization algorithms. The results showed that the proposed method has some advantages in comparing NSGA-II and MOPSO. Moreover, when IMOGSA is compared with MOPSO in GD aspect, it also needs to be improved.

In the future work, we will use more benchmark functions to analysis the proposed algorithm. On the initial population, we will use chaotic map to improve convergence and distribution of IMOGSA.
References

[1] Lai X, Li C, Zhou J and Zhang N 2019 Multi-objective optimization of the closure law of guide vanes for pumped storage units Renewable Energy 139 302-312

[2] Gao K, Cao Z, Zhang L, Chen Z, Han Y and Pan Q 2019 A Review on Swarm Intelligence and Evolutionary Algorithms for Solving Flexible Job Shop Scheduling Problems IEEE/CAA Journal of Automatica Sinica 6 1-13

[3] Sakthivel V P, Suman M and Sathya P D 2020 Combined economic and emission power dispatch problems through multi-objective squirrel search algorithm Applied Soft Computing 100

[4] Deb K, Pratap A and Agarwal S 2002 A fast and elitist multi-objective genetic algorithm: NSGA-II IEEE Trans on Evolutionary Computation 6(2) 182-197

[5] Coello C A C, Pulido G T and Lechuga M S 2004 Handling multiple objectives with particle swarm optimization IEEE Trans on Evolutionary Computation 8(3) 256-279

[6] Rashedi E, Nezamabadi-Pour H and Saryazdi S 2009 GSA: A gravitational search algorithm Information Sciences 179 (13) 2232-2248

[7] Rashedi E, Rashedi E and Nezamabadi-pour H 2018 A comprehensive survey on gravitational search algorithm Swarm and Evolutionary Computation 41 141-158

[8] Mirjalili S and Gandomi A H 2017 Chaotic gravitational constants for the gravitational search algorithm Applied Soft Computing 53 407-419

[9] Han X, Quan L, Xiong X, Almeter M, Xiang J and Lan Y 2017 A novel data clustering algorithm based on modified gravitational search algorithm Engineering Applications of Artificial Intelligence 61 1-7

[10] Sun G, Ma P, Ren J, Zhang A and Jia X 2018 A stability constrained adaptive alpha for gravitational search algorithm Knowledge-Based Systems 139 200-213

[11] Li L L, Lin G Q, Tseng M L, Tan K and Lim M K 2018 A maximum power point tracking method for PV system with improved gravitational search algorithm Applied Soft Computing 65 333-348

[12] Hassanzadeh H R, Rouhani M 2010 A multi-objective gravitational search algorithm 2010 2nd Int Conf on Computational Intelligence, Communication Systems and Networks(CICSyN) 32 7-12

[13] Nobahari H, Nikusokhan M and Siarry P 2012 A multi-objective gravitational search algorithm based on non-dominated sorting Int J of Swarm Intelligence Research(IJSIR) 3(3) 32-49

[14] Tariq I, AlSattar H A, Zaidan A A, Zaidan B B, Abu Bakar M R, Mohammed R T, Albahri O S, Alsalem M A and Albahri A S 2018 MOGSABAT: a metaheuristic hybrid algorithm for solving multi-objective optimisation problems Neural Computing and Applications 32(5)

[15] Mirjalili S 2016 SCA: A Sine Cosine Algorithm for solving optimization problems Knowledge-Based Systems 96 120-133

[16] Tang L and Wang X 2013 A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems IEEE Trans on Evolutionary Computation 17(1) 20-45

[17] Sun J J, Lv J J, Qian W Y and Sun Y 2013 A new algorithm for the multi-objective optimization problems based on the law of gravity Journal of Bohai University(Natural Science Edition) 34(04) 348-351