



Sun, Genyun and Ma, Ping and Ren, Jinchang and Zhang, Aizhu and Jia, Xiuping (2018) A stability constrained adaptive alpha for gravitational search algorithm. Knowledge Based Systems, 139. pp. 200-213. ISSN 0950-7051 , <http://dx.doi.org/10.1016/j.knosys.2017.10.018>

This version is available at <https://strathprints.strath.ac.uk/62088/>

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<https://strathprints.strath.ac.uk/>) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: strathprints@strath.ac.uk

A stability constrained adaptive alpha for gravitational search algorithm

Genyun Sun^{1,2*}, Ping Ma^{1,2}, Jinchang Ren^{4*}, Aizhu Zhang^{1,2}, Xiuping Jia^{1,3},

¹*School of Geosciences, China University of Petroleum (East China), Qingdao, 266580, China*

²*Laboratory for Marine Mineral Resources Qingdao National Laboratory for Marine Science and Technology, Qingdao, 266071, China*

³*School of Engineering and Information Technology, The University of New South Wales at Canberra, Canberra, ACT2600, Australia*

⁴*Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, United Kingdom*

Abstract:

Gravitational search algorithm (GSA), a recent meta-heuristic algorithm inspired by Newton's law of gravity and mass interactions, shows good performance in various optimization problems. In GSA, the gravitational constant attenuation factor alpha (α) plays a vital role in convergence and the balance between exploration and exploitation. However, in GSA and most of its variants, all agents share the same α value without considering their evolutionary states, which has inevitably caused the premature convergence and imbalance of exploration and exploitation. In order to alleviate these drawbacks, in this paper, we propose a new variant of GSA, namely stability constrained adaptive alpha for GSA (SCAA). In SCAA, each agent's evolutionary state is estimated, which is then combined with the variation of the agent's position and fitness feedback to adaptively adjust the value of α . Moreover, to preserve agents' stable trajectories and improve convergence precision, a boundary constraint is derived from the stability conditions of GSA to restrict the value of α in each iteration. The performance of SCAA has been evaluated by comparing with the original GSA and four alpha adjusting algorithms on 13

20 conventional functions and 15 complex CEC2015 functions. The experimental results have demonstrated that
21 SCAA has significantly better searching performance than its peers do.

22 **Keywords:** Meta-heuristic algorithm, Gravitational Search Algorithm, Adaptive parameter, Stability Conditions,
23 Exploration and exploitation

24 **1. Introduction**

25 With the growing complexity in many real-world optimization problems, adaptable and flexible meta-heuristic
26 algorithms are of increasing popularity due to their efficient performances [5,34]. In recent years, a variety of
27 meta-heuristic algorithms have been proposed, including Genetic Algorithm (GA) [3], Particle Swarm Optimization
28 (PSO) [34], Differential Evolution (DE) [45], Artificial Bee Colony (ABC) [19] and Gravitational search algorithm
29 (GSA) [35], etc. Among these algorithms, GSA is one of the latest population-based stochastic algorithm that
30 originates from the Newton's law of gravity and motion [35]. GSA considers every agent as a celestial body
31 attracting each other with a gravitational force that is directly proportional to the product of their masses and
32 inversely proportional to the squared distance between them. Agents search for the optimum by their interactive
33 movements. Since it was developed, GSA has gained popularity due to its several attractive features, such as simple
34 structure, easy implementation and well understanding [9,18]. However, there are still some drawbacks in GSA,
35 especially the premature convergence and imbalance of exploration and exploitation [9,22,55,56].

36 Recently, numerous improvements have been proposed to overcome these drawbacks. One active research trend
37 is to hybridize GSA with other meta-heuristics algorithms, such as DE [25], PSO [4,17,31,32], GA [39,47], ABC
38 [12] and Simulated Annealing (SA) [24]. For example, Li et al. [25] incorporated both the concepts of DE and GSA

39 and proposed a hybrid DE-GSA approach, in which agents were updated not only by DE operators but also by GSA
40 mechanisms. Mirjalili et al. introduced the social thinking of PSO into GSA to accelerate convergence in the last
41 iterations and improve the search ability [4,17,32]. In [39,47], GSA was hybridized with GA to escape from local
42 optima when applied to cope with multi-level image thresholding and neural network training issues, respectively.

43 Another research trend is to introduce new learning strategies into GSA. To tackle the prematurity problem of
44 GSA, Sun et al. [48] presented a locally information topology by taking individual heterogeneity into account and
45 Doraghinejad et al. [7] embedded the Black Hole theory into the original GSA. Sarafrazi et al. [37] defined a new
46 operator named as “disruption” to increase the exploration and exploitation ability of GSA. For overcoming the
47 limitation of lack of historical memory in GSA, the information of agents’ best solution obtained so far was
48 introduced in [18]. Xiao et al. [53] modified GSA by introducing the chaotic local search operator to avoid the local
49 optima trapping problem. Besides, Soleimanpour-moghadam et al. [42] proposed a Quantum based GSA for
50 increasing the population diversity. For striking a good balance between exploration and exploitation, Khajezadeh
51 et al [20] developed a modified GSA (IGSA) by introducing a controlled trajectory for velocity update that limited
52 the velocity within a certain interval value.

53 In addition, there is another strong research trend towards designing new parameter adjusting strategies of GSA
54 to improve its performance. In GSA, the gravitational constant G^t determines the convergence speed and the
55 balance of exploration and exploitation. In order to improve the search ability of GSA, several linearly decreasing
56 functions of G^t were used in [13,14] to extend the solution search space. Li et al. [23] proposed a piecewise
57 function based GSA (PFGSA) for providing more rational gravitational constant to control the convergence. Vijaya
58 Kumar et al. [52] developed a fuzzy adaptive GSA (FAGSA), where fuzzy rules were used to determine the optimal

59 selection of gravitational constant. More specially, by adjusting the attenuation factor alpha (α), G^t is
60 correspondingly changed and leads to the alteration of agents' movement directions and steps [2,36]. Thus, the
61 parameter α plays an important role in the searching ability of GSA. However, a constant parameter α was used
62 in the original GSA in the whole evolutionary process, which may severely affect the optimization performance. To
63 address this limitation, a number of alpha adjusting strategies have been proposed. In [43] and [11], a fuzzy strategy
64 was used to adjust the α value on the basis of the iteration number for the sake of promoting the balance of
65 exploration and exploitation and discouraging the premature convergence. In [23] and [56], a hyperbolic function
66 was introduced to replace the fixed value of α , which requested α to be changed with iteration to tackle the
67 premature problem. Besides, Saeidi-Khabisi et al. [36] proposed an adaptive alpha determination strategy by using
68 a fuzzy logic controller. In this method, some feedback information including the current iteration value, population
69 diversity, population progress and the α value in the previous iteration were utilized to adjust α dynamically,
70 which aimed at accelerating the convergence rate and preventing prematurity. More comprehensive and detailed
71 overview of the GSA variants can be found in [33,40].

72 Nevertheless, the aforementioned alpha adjusting methods have mitigated but not solved the premature
73 convergence. One key issue is that most of them adopt the same α value for all agents in each iteration without
74 considering their evolutionary states. Moreover, very limited focus has been put on the stability of GSA, though it
75 actually promotes the convergence speed and precision. After having elaborated investigation of the parameter α
76 and the stability conditions, a new adaptive alpha adjusting strategy, stability constrained adaptive alpha for GSA
77 (SCAA), is introduced in this paper to enhance the performance of GSA. The novel contributions of the proposed
78 SCAA are highlighted in two aspects as follows:

79 (1) An adaptive alpha adjusting strategy: In SCAA, the evolutionary state of each agent is first estimated.
80 According to the estimated state, the variation of the agent's position and fitness are used as feedback to adaptively
81 adjust its α value. Consequently, the novel alpha adjusting method can accelerate the convergence speed and
82 alleviate the premature problem.

83 (2) Stability-based boundary constraint for parameter α : For further improving the convergence speed and
84 precision, a boundary constraint on the basis of stability conditions is presented to restrict the α value in each
85 iteration. Experimental results show that this α boundary constraint ensures the stable convergence.

86 The remainder of this paper is organized as follows. Section 2 provides some preliminaries of GSA. The detail
87 of the proposed method is discussed in Section 3. In Section 4, experimental results and stability analysis are given
88 to evaluate the proposed algorithm. Finally, some concluding remarks are drawn in Section 5.

89 **2. Gravitational search algorithm**

90 GSA is a population-based meta-heuristic algorithm motivated by the laws of gravity and mass interactions [35].
91 In GSA, every agent $X_i = [x_{i1}, \dots, x_{id}, \dots, x_{iD}]$ ($i = 1, 2, \dots, NP$) attracts each other by a medium called gravitational
92 force in a D -dimensional search space. The gravitational force is directly proportional to their masses and inversely
93 proportional to their squared distance [35,36,43]. Accordingly, agents tend to move towards other agents with
94 heavier masses, which are corresponding to good solutions in the search space [31,32]. The mass of the i -th agent in
95 the iteration t , M_i^t , is calculated as follows:

$$mass_i^t = \frac{fit_i^t - worst^t}{best^t - worst^t} \quad (1)$$

$$M_i^t = \frac{mass_i^t}{\sum_{j=1}^{NP} mass_j^t} \quad (2)$$

96 where fit_i^t is the fitness value of the i -th agent in the iteration t . For the minimum problem,

$$97 \quad best^t = \min_{j \in \{1, \dots, NP\}} fit_j^t, worst^t = \max_{j \in \{1, \dots, NP\}} fit_j^t.$$

98 During all epochs, the gravitational force exerted on the i -th agent from the j -th agent at a specific time t is
99 defined by Eq. (3).

$$F_{id,jd}^t = G^t \frac{M_i^t \times M_j^t}{\|X_i^t, X_j^t\|_2 + \varepsilon} (x_{id}^t - x_{jd}^t) \quad (3)$$

100 where M_j^t and M_i^t are the gravitational mass related to the i -th agent and j -th agent, respectively. G^t is the
101 gravitational constant in the iteration t , $\|X_i^t, X_j^t\|_2$ is the Euclidian distance between the i -th agent and j -th agent,
102 ε is a small positive constant.

103 In the d -th dimension of the problem space, the total force that acts on the agent i is calculated by:

$$F_{id}^t = \sum_{j \in K_{best}, j \neq i}^{NP} rand_j F_{id,jd}^t \quad (4)$$

104 where $rand_j$ is a random number between the interval $[0,1]$, which is used to provide the random movement step
105 for agents to empower their diverse behaviors. K_{best} is an archive to store K superior agents (with bigger masses and
106 better fitness values) after fitness sorting in each iteration, whose size is initialized as NP and linearly decreased
107 with time down to one. Thus, by the law of motion, the acceleration of the agent i in the d -th dimension in the
108 iteration t , a_{id}^t , is calculated by Eq. (5).

$$a_{id}^t = F_{id}^t / M_i^t \quad (5)$$

109 The gravitational constant G^t is defined as follows:

$$G^t = G_0 \times e^{\left(-\alpha \frac{t}{t_{\max}}\right)} \quad (6)$$

110 where α is the gravitational constant attenuation factor and t_{\max} is the maximum number of iterations. In the
111 original GSA, G_0 is set to 100 and α is set to 20. In this way, the gravitational constant G^t is initialized to G_0
112 at the beginning and decreases exponentially into zero with lapse of time.

113 The velocity and position of the agent i are updated by

$$v_{id}^{t+1} = rand_i \times v_{id}^t + a_{id}^t \quad (7)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (8)$$

114 where $rand_i$ is a uniform random variable in the interval $[0, 1]$ and it can give a randomized characteristic to the
115 search. In this paper, for clearly describing and calculating the stability conditions in Section 3.2, a user-specified
116 inertia weight w is introduced to determine how easily the previous velocity can be changed. Thus, Eq. (7) is
117 rewritten as follows:

$$v_{id}^{t+1} = w \times v_{id}^t + a_{id}^t \quad (9)$$

118 3. The proposed SCAA algorithm

119 3.1. Adaptive alpha adjusting strategy

120 In the original GSA, each agent moves toward the center composed by those elite agents stored in K_{best} [32]. If
121 the center locates at a promising region, the agent's fitness value becomes increasingly better. As shown in Fig. 1

122 (a), the center c_1^t is close to the global optimum and the agent M_1^t has been self-improved in several sequential
 123 steps in the current iteration t . In this case, the impact of elite masses should be enhanced to strengthen the movement
 124 tendency of M_1^t towards the center c_1^t for accelerating convergence. On the contrary, if the elite masses are
 125 trapped into local optima, especially in the latter stage when the size of K_{best} is decreased to a smaller value, their
 126 center is more likely far away from the global optimum. As a result, the agent may experience false convergence
 127 and its fitness can be worse and worse [32,48] as shown in Fig. 1 (b). In this regard, the impact of elite masses
 128 should be weakened to reduce the movement tendency of M_1^t towards the center c_1^t for avoiding prematurity.

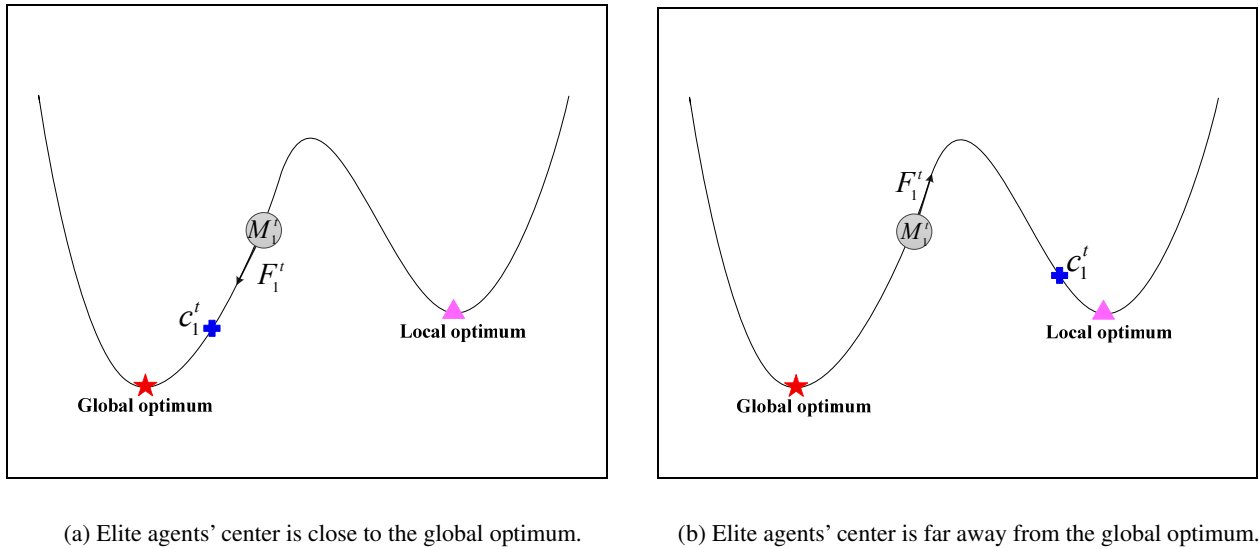


Fig. 1. Schematic diagram of the agent's movement.

129 As analyzed above, agents may experience different evolutionary states during a course of simulation, which
 130 can be defined in two cases: (1) the agent has improved self-solution in several sequential steps and (2) the agent
 131 has failed to improve self-solution for several sequential steps. The tendency of an agent moving towards elite
 132 masses is supposed to be dynamically changed corresponding to its states for a better convergence. According to Eq.
 133 (6), the gravitational constant G^t is the modulus of force that controls the impact of elite masses. Moreover, by

134 adjusting the attenuation factor alpha (α), G^t is also changed accordingly. Specifically, a smaller α value results
 135 in a greater G^t that promotes the agent to move faster toward the center of K_{best} , while a larger α value leads to a
 136 lower G^t that prevents the agent from reaching the center [9,32]. Therefore, in this paper, the parameter α is
 137 adaptively adjusted according to the agent's current state.

138 In order to estimate agents' evolutionary states, two counters, ns and nf , are used as the indicators. For a
 139 given agent i , the ns_i^t and nf_i^t are both set to zero initially. Then, as described in Eqs. (10) and (11), if X_i^t can
 140 improve self-solution in the new iteration, ns_i^t is incremented by 1 and nf_i^t is set to 0. Otherwise, the counter
 141 nf_i^t is incremented by 1 whilst ns_i^t is set to 0. Obviously, the values of ns_i^t and nf_i^t indicate the different
 142 evolutionary states of the agent i .

$$ns_i^t = \begin{cases} ns_i^{t-1} + 1, & \text{if } fit_i^t < fit_i^{t-1} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

$$nf_i^t = \begin{cases} nf_i^{t-1} + 1, & \text{if } fit_i^t \geq fit_i^{t-1} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

143 We set a limit value lp to judge whether or not to conduct the adjustment of α . For the i -th agent in the
 144 iteration t , if ns_i^t exceeds lp , X_i^t is recognized as in the first case [1,49] and its α value should be decreased to
 145 enhance the convergence to elite masses. More specifically, a less variation in fitness or position of agent X_i^t
 146 denotes its slow movement. Thus, its α value should be decreased greatly to reach a higher convergence speed.
 147 When the position or fitness of agent X_i^t change greatly, it means the agent moves faster or locates a more
 148 promising region, thus the α value needs be slightly decreased to relatively refine its neighboring areas. On the
 149 other hand, if nf_i^t exceeds lp , X_i^t is regarded as in the second case [1,49]. The parameter α is supposed to be
 150 increased to reduce the attraction of elite masses, i.e. its α value changes just in the opposite way as the first case

151 does. Based on the discussions above, in this paper, the variation of fitness and position of an agent are employed as
 152 feedback to adaptively adjust the parameter α according to the evolutionary state, which is described in Eq. (12).

$$\alpha_i^t = \begin{cases} \alpha_i^{t-1} - rand \times \exp\left(-\frac{\|\mathbf{X}_i^t, \mathbf{X}_i^{t-1}\|_2}{\max_{j \in \{1, \dots, N\}} \|\mathbf{X}_j^t, \mathbf{X}_j^{t-1}\|_2} + \varepsilon\right) \times \exp\left(\frac{fit_i^t - fit_i^{t-1}}{\max_{j \in \{1, \dots, N\}} (fit_j^{t-1} - fit_j^t) + \varepsilon}\right), & \text{if } ns_i^t \geq lp \\ \alpha_i^{t-1} + rand \times \left(1 - \exp\left(-\frac{\|\mathbf{X}_i^t, \mathbf{X}_i^{t-1}\|_2}{\max_{j \in \{1, \dots, N\}} \|\mathbf{X}_j^t, \mathbf{X}_j^{t-1}\|_2} + \varepsilon\right)\right) \times \left(1 - \exp\left(\frac{fit_i^{t-1} - fit_i^t}{\max_{j \in \{1, \dots, N\}} (fit_j^t - fit_j^{t-1}) + \varepsilon}\right)\right), & \text{if } nf_i^t \geq lp \\ \alpha_i^{t-1}, & \text{otherwise} \end{cases} \quad (12)$$

153 where α_i^t is the alpha value of the i -th agent in the t -th iteration, $rand$ is a random value in $[0,1]$ that can enhance
 154 the diversity of α_i^t [21]. From Eq. (12), it is obvious that α_i^t can be dynamically changed in coincidence with its
 155 evolutionary state.

156 3.2. Stability-based boundary constraint

157 In GSA, the stability conditions play an important role in promoting the convergence speed and precision. In
 158 this section, after having elaborated investigation of the stability conditions, we present a boundary constraint for
 159 parameter α to enhance the stable convergence of GSA.

160 As proven in [18], for the i -th agent in the t -th iteration, gravitational constant G^t has to satisfy Eq. (13) to
 161 ensure the stability of its movement trajectory.

$$0 < G^t < \frac{4(1+w) \times (R_{i,j} + \varepsilon)}{\sum_{j \in B_i} M_{p_j}^t + \sum_{j \in W_i} M_{p_j}^t} \quad (13)$$

162 where w satisfies the stability condition: $0 \leq w < 1$, B_i is a set of agents which own better fitness than the agent i ,

163 W_i is a set of agents whose fitness values are no better than the agent i . $M_{p_j}^t$ is the personal best fitness history

164 found so far by the agent j , which is calculated as follows:

$$p_{jd}^{t+1} = \begin{cases} p_{jd}^t, & \text{if } \hat{fit}_p^{t+1} > fit_{p_j}^t \\ x_{jd}^{t+1}, & \text{if } \hat{fit}_p^{t+1} \leq fit_{p_j}^t \end{cases} \quad (14)$$

$$mass_j^t = \frac{fit_{p_j}^t - worst^t}{best^t - worst^t} \quad (15)$$

$$M_{p_j}^t = \frac{mass_j^t}{\sum_{k=1}^N mass_k^t} \quad (16)$$

165 According to the analysis in [2], parameter α has a drastic influence on G^t . Therefore, on the basis of the

166 gravitational constant equation in Eq. (6), we can rewrite Eq. (13) as follows:

$$\begin{aligned} 0 < G_0 \times e^{\left(-\frac{\alpha-t}{t_{\max}}\right)} &< \frac{4(1+w) \times (R_{i,j} + \varepsilon)}{\sum_{j \in B_i} M_{p_j}^t + \sum_{j \in W_i} M_{p_j}^t} \\ \Rightarrow 0 < e^{\left(-\frac{\alpha-t}{t_{\max}}\right)} &< \frac{4(1+w) \times (R_{i,j} + \varepsilon)}{G_0 \times \left(\sum_{j \in B_i} M_{p_j}^t + \sum_{j \in W_i} M_{p_j}^t \right)} \\ \Rightarrow \frac{t_{\max}}{t} \ln \left(\frac{G_0 \times \left(\sum_{j \in B_i} M_{p_j}^t + \sum_{j \in W_i} M_{p_j}^t \right)}{4(1+w) \times (R_{i,j} + \varepsilon)} \right) &< \alpha < \text{Inf} \end{aligned} \quad (17)$$

167 where w is set to a certain value in the range $[0,1)$. For simplicity, we define

168 $\alpha_{\min}^t = \frac{t_{\max}}{t} \ln \left(G_0 \times \left(\sum_{j \in B_i} M_{p_j}^t + \sum_{j \in W_i} M_{p_j}^t \right) / 4(1+w) \times (R_{i,j} + \varepsilon) \right)$. Eq. (17) offers the lower boundary that α should be

169 satisfied. Thus, for the agent i in current iteration t , if its alpha value is lower than α_{\min}^t , α_i^t will be restricted to its

170 boundary:

$$\alpha_i^t = \alpha_{\min}^t, \quad \text{if } \alpha_i^t < \alpha_{\min}^t \quad (18)$$

171 Eq. (18) formulates the lower boundary constraint for parameter α . However, there is lack of an upper
 172 boundary constraint for α . In practice, too large value of α can cause the search to the stagnation and impair the
 173 exploration capability. Therefore, it is unreasonable to set the upper boundary value to the infinite great as shown in
 174 Eq. (17). To resolve this problem, we set a parameter α_{\max} to control the upper boundary of α . Note α_{\max} is
 175 fixed to a certain value. In such a way, the boundary constraint equation of α , Eq. (17), can now be rewritten as:

$$\alpha_{\min}^t \leq \alpha_i^t \leq \alpha_{\max} \quad (19)$$

176 If α_i^t is larger than α_{\max} , it will be conditioned to its upper boundary as follows:

$$\alpha_i^t = \alpha_{\max}, \quad \text{if } \alpha_i^t > \alpha_{\max} \quad (20)$$

177 The sensibility tests on α_{\max} in Section 4.4 verify its effectiveness on the performance of GSA. In a word,
 178 the parameter α_i^t should satisfy the boundary conditions as show in Eq. (19). If the α_i^t value exceeds its
 179 boundary, α_i^t will be forced to gather on its corresponding boundary as described in Eqs. (18) and (20). Thus,
 180 the stability-based boundary constraint can ensure the stable convergence of the swarm.

181 Based on the above introduction of the SCAA algorithm, its complete pseudo-code is summarized in
 182 **Algorithm 1** as follows.

Algorithm 1 Pseudo-code of SCAA

- 1 Generate uniformly distributed population randomly and initialize the velocity associated with each agent;
- 2 Calculate the fitness value of each agent and generate the K_{best} agents of the population;
- 3 Set $FES = NP$, $t=1$, $ns = 0$, $nf = 0$;

```

4  While  $FES \leq FES_{\max}$  do
5  /*Adaptive alpha adjusting strategy*/
6  For  $i = 1$  to  $NP$  do
7      If  $fit_i^t < fit_i^{t-1}$  then
8          Set  $ns_i^t = ns_i^{t-1} + 1$ ,  $nf_i^t = 0$ ;
9          If  $ns_i^t \geq lp$  then
10             Set  $ns_i^t = 0$ , decrease  $\alpha_i^t$  according to Eq. (12);
11             Else
12                  $\alpha_i^t = \alpha_i^{t-1}$ ;
13             End If
14         Else
15             Set  $nf_i^t = nf_i^{t-1} + 1$ ,  $ns_i^t = 0$ ;
16             If  $nf_i^t \geq lp$  then
17                 Set  $nf_i^t = 0$ , increase  $\alpha_i^t$  according to Eq. (12);
18             Else
19                  $\alpha_i^t = \alpha_i^{t-1}$ ;
20             End If
21         End If
22  /*Stability-based boundary constraint*/
23      Calculate the lower boundary of alpha  $\alpha_{\min}^t$  using Eq. (17);
24      Restrict  $\alpha_i^t$  to its boundary according to Eqs. (18) and (20);
25      Calculate the acceleration  $\alpha_i^t$  by Eqs. (6), (3), (4) and (5);
26      Update the position  $X_i^t$  according to Eqs. (7) and (8);
27      Calculate the fitness values  $fit_i^t$ ;
28       $FES ++$ ;
29  End For
30   $t ++$ ;
31 End While
32 Output the best solution achieved so far.

```

183 3.3. Search behaviors of SCAA

184 In this section, the search behaviors of SCAA are investigated so as to validate the effectiveness of the proposed
 185 alpha adjusting strategy. We herein take a time-varying 30- D Sphere function as an example.

$$f(x-r) = \sum_{j=1}^D (x_j - r)^2, \quad x_j \in [-100, 100] \quad (21)$$

186 where r is initialized to -10 and shifted to 10 in the 200th iteration with the total number of iterations set as 2000.
 187 That is, the theoretical minimum of f shifts from $(-10)^D$ to $(10)^D$ during the evolutionary process. SCAA and GSA
 188 are employed with the same initialized population, which include 50 agents to solve this minimization problem. For
 189 a better observation, only the first agent's alpha value α_1 and its first dimension trajectory x_{11} are recorded in Fig.
 190 2 (a) and Fig. 2 (b), respectively. The convergence curve of SCAA and GSA are depicted in Fig. 2 (c).

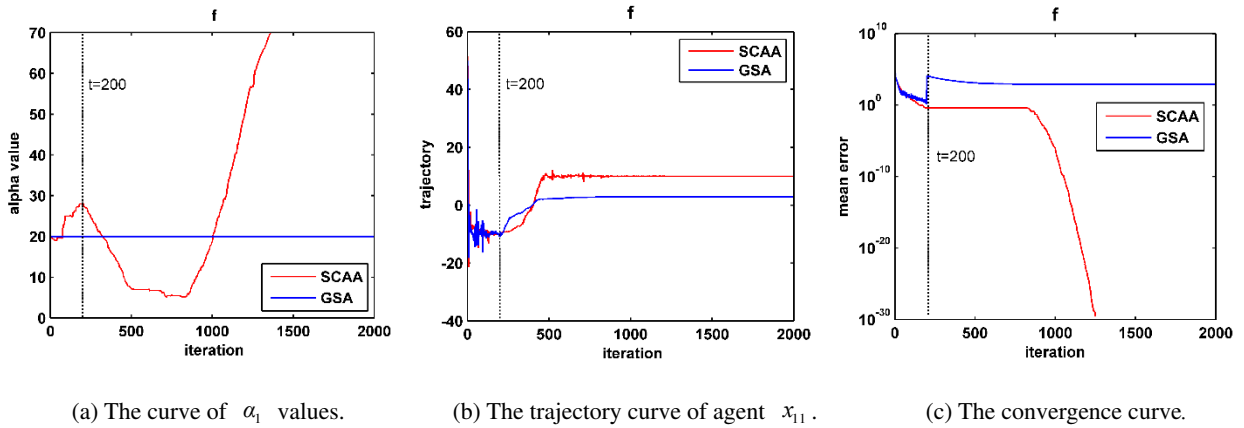


Fig. 2. Search behaviors of SCAA on 30- D Sphere function.

191 During the evolutionary process, the center of elite masses is dynamically changed with agents' movements. In
 192 the early iterations, the elite agents may experience poor areas, hence the center is more likely located at the local

193 optima. In this case, as shown in Fig. 2 (a), the parameter α_1 is continuously increased to a higher value before the
194 200th iteration to weaken the impact of elite masses. This adjustment prevents the agent from being trapped into
195 local optima as revealed in Fig. 2 (b). The x_{11} gradually reaches the global minimum (-10). Thereafter, in the 200th
196 iteration, the value of r is shifted to 10. At this moment, those agents that are closer to the new global minimum
197 would be the elite masses in K_{best} . To this end, the impact of elite masses should be enhanced to accelerate the
198 convergence. From the Fig. 2 (b) and Fig. 2 (c), it can be seen that parameter α_1 is rapidly decreased into a smaller
199 value whilst the trajectory of x_{11} deviates from -10 and fast reaches the new global optimum. In the latter
200 iterations, when agents cluster together and converge to the globally optimal area, less improvement of agents'
201 fitness are obtained. Therefore, parameter α_1 is rapidly increased under the stability-based boundary constraint to
202 improve the convergence precision. As depicted in Fig. 2 (c), SCAA finally achieves the globally optimal results.
203 With regard to the original GSA, even though the swarm can converge to the global optimum in the early iterations,
204 the agents are trapped into local optima after the global minimum is changed. This behavior may mainly result from
205 the usage of a constant α , which lacks the dynamic momentum in coincidence with the changed search
206 environment. Conversely, in SCAA, the α value of each agent is adaptively adjusted according to its evolutionary
207 state, which motivates the agent to detect the promising direction and avoid the premature convergence.

208 **4. Experimental verification and comparison**

209 *4.1. Experimental setup*

210 To fully evaluate the performance of SCAA, a thorough comparison with the original GSA [35] and four recent
211 GSA variants with well-established alpha adjusting strategies, including MGSA- α [23], FS α (Increase) [43],
212 FS α (Decrement) [43] and FuzzyGSA [36], is tested in this study. Note that two alpha adjusting methods in [43],
213 FS α (Increase) and FS α (Decrement), are both introduced for investigating the effect of different alpha change trends
214 on the searching performance and the stability of GSA. The tested functions include 28 scalable benchmark functions,
215 where F1-F13 are conventional problems in [54] and F14-F28 are derived from the CEC2015 functions [26]. Detailed
216 description of these benchmark functions can be found in [26,54]. In this paper, the evaluations are performed under
217 30 dimensions (30- D) and the accuracy level δ is set to 0.001 for all benchmark functions.

218 During the experiments, the parameter configurations of all involved algorithms are set according to their
219 recommended settings [48]. Note that for the fair comparison, MGSA- α only uses the α adjusting strategy and
220 abandons the mutation operator in MGSA. As for SCAA, the initial value of parameter alpha α_0 is set to 20 for all
221 agents on the basis of the recommendations in [35]. The inertia weight is set to $w = 1 - 1/t_{\max}$ according to the range
222 in stability conditions as suggested in [18]. The limit value lp and α_{\max} are respectively set to 2 and 70 according to
223 the sensitive analysis conducted in Section 4.5. The detailed parameter settings of all involved algorithms are
224 summarized in Table 1.

225 In the experiments, common parameters are the total number of trials, maximum number of function evaluations
226 (FES_{\max}) and the population size (NP). All algorithms were independently run 51 times to reduce random discrepancy
227 [26]. The FES_{\max} for terminating the algorithms is specified as $10000 \times D$ for each function as suggested in [26].
228 The population size NP for solving the 30- D problems is set to 50 based on the recommendation in [35,36,43].
229 Moreover, since all involved algorithms have the same fitness evaluations $FES = NP$ in each iteration, the maximum

230 number of iterations t_{\max} is set to $t_{\max} = FES_{\max}/NP$ in this paper. All the algorithms are implemented using Matlab
 231 2012b and executed on a computer with Intel Pentium 4 CPU (2.40 GHz) and 4 GB of memory.

Table 1

Parameter settings of the involved GSA variants.

Algorithms	Year	Parameter settings
GSA [35]	2009	$G_0 = 100$, $\alpha = 20$
FuzzyGSA [36]	2012	$ED \in [0,1]$, $CM \in [0,1]$, $t \in [0, t_{\max}]$, $\alpha \in [29,31]$
FS α (Increase) [43]	2013	$t \in [0, 100\%] \times t_{\max}$, $\alpha \in [0, 150]$
FS α (Decrement) [43]	2013	$t \in [0, 100\%] \times t_{\max}$, $\alpha \in [0, 150]$
MGSA- α [22]	2016	$G_0 = 100$, $\gamma = 0.2$, $\eta = 10$, $\lambda = 25$
SCAA	–	$G_0 = 100$, $\alpha_0 = 20$, $w = 1 - 1/t_{\max}$, $lp = 2$, $\alpha_{\max} = 70$

232 4.2. Performance metrics

233 In this study, the searching accuracy, searching reliability and searching efficiency of different algorithms are
 234 evaluated in terms of the mean error (*Mean*), success rate (*SR*), success performance (*SP*) and execution time
 235 (*runtime* in seconds), respectively [46]. *Mean* is the average error between the best output results and the global
 236 optimum of the optimization problem [46]. *SR* represents the percentage of successful runs where the algorithm
 237 achieves good solutions under the predefined accuracy level δ [46]. *SP* denotes the number of *FES* required by an
 238 algorithm to achieve the acceptable solutions under δ [46]. The performance metric *runtime* is the execution time of
 239 an algorithm to obtain the acceptable solutions [27]. Meanwhile, we record the standard deviation (*SD*) of the
 240 optimization error and rank the algorithms from the smallest *Mean* to the highest. The average ranks and the overall
 241 ranks obtained by algorithms are also recorded.

242 For the rigorous comparison between SCAA and its competitors, a non-parametric statistical test is used in this
243 study. Unlike parametric tests, non-parametric tests are employed to analyze the performance of stochastic algorithms
244 based on computational intelligence despite the assumptions of data types used are violated [10]. Specifically, the
245 Wilcoxon signed ranks test [6] with a confidence level of 5% is utilized to perform pairwise comparison between
246 SCAA and its peers in this paper.

247 4.3. Comparison with other alpha adjusting GSA variants

248 Following the experimental setup and parameter settings in Section 4.1, comprehensive experiments are employed
249 to evaluate the overall searching behaviors of GSA, MGSA- α , FuzzyGSA, FS α (Increase), FS α (Decrement) and
250 SCAA. For performance assessment, six metrics are used which include the *Mean*, *SD* and Wilcoxon signed ranks
251 test (*p-value*, *h-value*, *z-value* and *signedrank*), as summarized in Table 2 and Table 3, along with *SR*, *SP* and *runtime*
252 as reported in Table 4. Besides, we rank the six competing algorithms according to their *Mean* values. In Table 2 and
253 Table 3, the symbol '*h*' describes the non-parameter test results, where '+' means SCAA is significantly better than
254 the compared algorithms, '-' indicates it significantly performs worse and '=' stands for comparable performance
255 between the algorithms. Moreover, we summarize the results of Wilcoxon test results at the bottom of Table 2 and
256 Table 3, respectively. The best result in each row is highlighted in bold in the Tables 2-4. Note that Fig. 3 depicts some
257 convergence curves of the competing algorithms.

Table 2

Optimization errors among six algorithms on 13 conventional problems at 30-*D*.

	metrics	GSA	MGSA- α	FuzzyGSA	FS α (Increase)	FS α (Decrement)	SCAA
F1	<i>Mean (Mean_rank)</i>	1.188E-17(6)	4.481E-34(3)	7.291E-27(5)	1.322E-38(2)	6.514E-27(4)	9.162E-58(1)
	<i>SD</i>	3.398E-18	7.093E-34	2.081E-27	4.230E-38	7.117E-28	2.283E-57
	<i>p-value (h-value)</i>	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	
F2	<i>Mean (Mean_rank)</i>	1.727E-08(6)	3.038E-16(3)	4.257E-13(5)	1.247E-18(2)	3.855E-13(4)	4.558E-20(1)
	<i>SD</i>	2.829E-09	2.295E-16	5.629E-14	1.094E-18	1.52E-15	9.218E-20
	<i>p-value (h-value)</i>	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	9.662E-09 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	-5.7366 (51)	-6.2146 (0)	
F3	<i>Mean (Mean_rank)</i>	1.500E-02(2)	2.264E+00(3)	1.272E+01(4)	5.556E+01(5)	1.749E+02(6)	7.700E-03(1)
	<i>SD</i>	3.010E-02	2.207E+00	7.251E+00	2.557E+01	6.488E+01	5.600E-03
	<i>p-value (h-value)</i>	1.197E-01 (=)	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-1.5560 (497)	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	
F4	<i>Mean (Mean_rank)</i>	1.823E-09(6)	5.247E-16(2)	5.486E-14(5)	1.329E-18(1)	3.385E-14(3)	5.041E-14(4)
	<i>SD</i>	2.171E-10	2.276E-16	9.236E-15	6.262E-19	2.305E-15	2.008E-13
	<i>p-value (h-value)</i>	5.145E-10 (+)	9.104E-01 (=)	1.309E-05 (+)	1.768E-09 (-)	1.309E-05 (-)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	0.1125 (675)	-4.3587 (198)	6.0178 (1305)	-4.3587 (198)	
F5	<i>Mean (Mean_rank)</i>	1.918E+01(2)	2.214E+01(3)	2.387E+01(4)	2.901E+01(5)	3.223E+01(6)	1.321E+01(1)
	<i>SD</i>	2.264E-01	1.769E-01	1.743E-01	2.290E+01	3.235E+01	4.512E-01
	<i>p-value (h-value)</i>	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	
F6	<i>Mean (Mean_rank)</i>	0.00E+00(1)	0.00E+00(1)	0.00E+00(1)	0.00E+00(1)	0.00E+00(1)	0.00E+00(1)
	<i>SD</i>	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	<i>p-value (h-value)</i>	1.000E+00(=)	1.000E+00(=)	1.000E+00(=)	1.000E+00(=)	1.000E+00(=)	
	<i>z-value (signedrank)</i>	— (0)	— (0)	— (0)	— (0)	— (0)	
F7	<i>Mean (Mean_rank)</i>	9.24E-02(6)	8.18E-02(5)	1.070E-02(2)	1.240E-02(3)	1.460E-02(4)	1.050E-02(1)
	<i>SD</i>	3.560E-02	2.890E-02	3.200E-03	5.000E-03	5.200E-03	6.200E-03
	<i>p-value (h-value)</i>	5.145E-10 (+)	5.145E-10 (+)	8.293E-01 (=)	1.153E-01 (=)	2.759E-04 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-6.2146 (0)	-0.2156 (640)	-1.5747 (495)	-3.6369 (275)	

	<i>Mean (Mean_rank)</i>	1.149E+04(5)	1.149E+04(5)	9.729E+03(2)	9.953E+03(4)	9.926E+03(3)	8.811E+03(1)
F8	<i>SD</i>	1.643E+02	1.674E+02	4.365E+02	4.408E+02	2.908E+02	6.786E+02
	<i>p-value (h-value)</i>	5.145E-10 (+)	5.145E-10 (+)	2.939E-07 (+)	1.124E-07 (+)	2.872E-08 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-6.2146 (0)	-5.1273 (116)	-5.3054 (97)	-5.5491 (71)	
	<i>Mean (Mean_rank)</i>	1.252E+01(1)	1.270E+01(2)	1.319E+01(4)	1.309E+01(3)	1.461E+01(6)	1.447E+01(5)
F9	<i>SD</i>	2.822E+00	3.125E+00	3.939E+00	2.879E+00	4.293E+00	3.096E+00
	<i>p-value (h-value)</i>	3.583E-01 (=)	3.390E-01 (=)	8.586E-01 (=)	5.611E-01 (=)	7.490E-02 (=)	
	<i>z-value (signedrank)</i>	0.9186 (761)	0.9561 (765)	-0.1781 (644)	0.5812 (725)	-1.7810 (473)	
	<i>Mean (Mean_rank)</i>	2.653E-09(6)	1.837E-14(3)	1.833E-13(5)	4.998E-15(2)	6.303E-14(4)	4.441E-15(1)
F10	<i>SD</i>	3.382E-10	7.548E-15	3.428E-13	1.305E-15	4.679E-15	6.486E-16
	<i>p-value (h-value)</i>	5.145E-10 (+)	6.769E-11 (+)	5.079E-10 (+)	4.700E-03 (+)	3.697E-10 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-6.5257 (0)	-6.1266 (0)	-2.8284 (0)	-6.2663 (0)	
	<i>Mean (Mean_rank)</i>	1.500E-03(4)	7.236E-04(3)	6.500E-03(5)	5.309E-04(2)	3.218E-01(6)	4.931E-04(1)
F11	<i>SD</i>	4.600E-03	3.600E-03	1.490E-02	2.900E-03	4.722E-01	1.900E-03
	<i>p-value (h-value)</i>	1.212E-01 (=)	4.652E-01 (=)	9.333E-04 (+)	1.000E+00(=)	7.614E-09 (+)	
	<i>z-value (signedrank)</i>	-1.5498 (7)	-0.7303 (3)	-3.3099 (4)	0 (3)	-5.7768 (0)	
	<i>Mean (Mean_rank)</i>	2.000E-03(3)	1.636E-32(2)	5.400E-03(5)	3.200E-03(4)	1.020E-02(6)	1.573E-32(1)
F12	<i>SD</i>	1.450E-02	4.366E-34	2.200E-02	1.620E-02	3.740E-02	2.900E-35
	<i>p-value (h-value)</i>	5.145E-10 (+)	1.896E-09 (+)	5.145E-10 (+)	6.195E-01 (=)	5.141E-10 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-6.0064 (3)	-6.2146 (0)	0.4966 (77)	-6.2147 (0)	
	<i>Mean (Mean_rank)</i>	1.249E-18(5)	2.121E-32(3)	8.077E-28(4)	1.359E-32(2)	2.154E-04(6)	1.346E-32(1)
F13	<i>SD</i>	3.394E-19	7.685E-33	2.653E-28	9.347E-34	1.500E-03	2.808E-48
	<i>p-value (h-value)</i>	5.145E-10 (+)	2.301E-08 (+)	5.145E-10 (+)	1.902E-01 (=)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-5.5877 (17)	-6.2146 (0)	1.3099 (33)	-6.2146 (0)	
	<i>Average Mean_rank</i>	4.0769	2.9231	3.9231	2.7692	4.5385	1.5325
	+ / = / -	9 / 4 / 0	9 / 4 / 0	10 / 4 / 0	6 / 6 / 1	10 / 2 / 1	

Table 3

Optimization errors among six algorithms on 15 CEC2015 functions at 30-D.

metrics	GSA	MGSA- α	FuzzyGSA	FS α (Increase)	FS α (Decrement)	SCAA	
F14	<i>Mean (Mean_rank)</i>	8.737E+05(2)	1.012E+06(3)	2.707E+06(5)	2.046E+06(4)	1.428E+07(6)	4.093E+05(1)
	<i>SD</i>	4.497E+05	3.907E+05	5.063E+06	1.297E+06	8.813E+06	2.405E+05
	<i>p-value (h-value)</i>	1.273E-08 (+)	9.141E-09 (+)	4.673E-09 (+)	5.145E-10 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-5.6897 (56)	-5.7459 (50)	-5.5884 (38)	-6.2146 (0)	-6.2146 (0)	
F15	<i>Mean (Mean_rank)</i>	4.729E+02(2)	7.445E+02(5)	7.169E+02(4)	7.634E+02(6)	4.701E+02(1)	5.328E+02(3)
	<i>SD</i>	4.771E+02	9.367E+02	9.480E+02	1.117E+03	6.262E+02	8.585E+02
	<i>p-value (h-value)</i>	5.486E-01 (=)	3.250E-01 (=)	2.489E-01 (=)	2.528E-01 (=)	7.858E-01 (=)	
	<i>z-value (signedrank)</i>	-0.5999 (599)	-0.9842 (558)	-1.1529 (540)	-1.1436 (541)	-0.2718 (634)	
F16	<i>Mean (Mean_rank)</i>	2.00E+01(1)	2.00E+01(1)	2.00E+01(1)	2.00E+01(1)	2.00E+01(1)	2.094E+01(2)
	<i>SD</i>	9.746E-05	6.594E-05	8.113E-05	1.109E-04	6.150E-05	5.840E-02
	<i>p-value (h-value)</i>	5.145E-10 (-)	5.145E-10 (-)	5.145E-10 (-)	5.145E-10 (-)	5.145E-10 (-)	
	<i>z-value (signedrank)</i>	6.2146 (1326)	6.2146 (1326)	6.2146 (1326)	6.2146 (1326)	6.2146 (1326)	
F17	<i>Mean (Mean_rank)</i>	2.073E+02(2)	1.963E+02(1)	2.194E+02(5)	2.084E+02(3)	2.363E+02(6)	2.173E+02(4)
	<i>SD</i>	2.201E+01	2.811E+01	1.924E+01	2.023E+01	2.261E+01	2.223E+01
	<i>p-value (h-value)</i>	2.830E-02 (+)	5.617E-04 (-)	4.148E-01 (=)	4.900E-03 (+)	1.215E-04 (+)	
	<i>z-value (signedrank)</i>	-2.1934 (429)	-3.4494 (295)	-0.8155 (576)	-2.8120 (363)	-3.8431 (253)	
F18	<i>Mean (Mean_rank)</i>	3.818E+03(4)	3.625E+03(1)	3.981E+03(5)	3.773E+03(2)	4.156E+03(6)	3.810E+03(3)
	<i>SD</i>	4.518E+02	4.456E+02	4.533E+02	5.325E+02	4.548E+02	4.548E+02
	<i>p-value (h-value)</i>	7.930E-01 (=)	2.010E-02 (-)	7.190E-02 (=)	9.030E-01 (=)	7.916E-04 (+)	
	<i>z-value (signedrank)</i>	-0.2625 (635)	-2.3246 (415)	-1.7997 (471)	-0.1219 (650)	-3.3557 (305)	
F19	<i>Mean (Mean_rank)</i>	1.302E+05(2)	3.553E+05(3)	6.856E+05(4)	9.472E+05(5)	1.704E+06(6)	5.587E+04(1)
	<i>SD</i>	6.396E+04	1.725E+05	2.962E+05	3.593E+05	6.204E+05	2.566E+04
	<i>p-value (h-value)</i>	1.420E-08 (+)	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-5.6709 (58)	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	-6.2146 (0)	
F20	<i>Mean (Mean_rank)</i>	1.324E+01(2)	1.524E+01(3)	2.236E+01(4)	2.441E+01(5)	6.397E+01(6)	9.779E+00(1)
	<i>SD</i>	4.061E+00	9.643E+00	1.949E+01	2.058E+01	2.388E+01	3.133E+00
	<i>p-value (h-value)</i>	6.522E-05 (+)	1.669E-06 (+)	6.142E-07 (+)	3.371E-08 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-3.9931 (237)	-4.7898 (152)	-4.9867 (131)	-5.5210 (74)	-6.2146 (0)	

F21	<i>Mean (Mean_rank)</i>	2.356E+04(2)	2.388E+04(3)	3.050E+04(4)	5.557E+04(5)	1.007E+05(6)	2.154E+04(1)
	<i>SD</i>	1.079E+04	7.509E+03	1.152E+04	3.344E+04	1.141E+05	8.329E+03
	<i>p-value (h-value)</i>	2.728E-01 (=)	6.760E-02 (=)	1.201E-06 (+)	5.797E-10 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-1.0967 (546)	-1.8278 (468)	-4.8555 (145)	-6.1959 (2)	-6.2146 (0)	
F22	<i>Mean (Mean_rank)</i>	1.511E+02(4)	1.265E+02(2)	1.513E+02(5)	1.262E+02(1)	2.025E+02(6)	1.358E+02(3)
	<i>SD</i>	1.218E+02	8.221E+01	1.218E+02	7.943E+01	1.627E+02	1.016E+02
	<i>p-value (h-value)</i>	1.700E-03 (+)	6.080E-02 (=)	7.446E-06 (+)	1.469E-04 (-)	1.425E-05 (+)	
	<i>z-value (signedrank)</i>	-3.1307 (329)	-1.8747 (463)	-4.4805 (185)	-3.7963 (258)	-4.3399 (200)	
F23	<i>Mean (Mean_rank)</i>	4.299E+05(2)	6.936E+05(3)	9.961E+05(4)	1.280E+06(5)	2.485E+06(6)	1.921E+05(1)
	<i>SD</i>	1.961E+05	2.310E+05	3.994E+05	6.108E+05	1.004E+06	5.998E+04
	<i>p-value (h-value)</i>	9.869E-10 (+)	5.145E-10 (+)	5.145E-10 (+)	5.462E-10 (-)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-6.115 (11)	-6.2146 (0)	-6.2146 (0)	-6.2052 (1)	-6.2146 (0)	
F24	<i>Mean (Mean_rank)</i>	3.244E+02(2)	3.343E+02(3)	3.480E+02(5)	3.473E+02(4)	3.849E+02(6)	3.226E+02(1)
	<i>SD</i>	9.771E+01	1.121E+02	1.557E+02	1.422E+02	2.021E+02	9.132E+01
	<i>p-value (h-value)</i>	6.080E-02 (=)	6.675E-04 (+)	1.272E-08 (+)	8.648E-09 (+)	8.648E-09 (+)	
	<i>z-value (signedrank)</i>	-1.8747 (463)	-3.4026 (300)	-5.6897 (56)	-5.7553 (49)	-5.7553 (49)	
F25	<i>Mean (Mean_rank)</i>	1.040E+02(3)	1.036E+02(2)	1.053E+02(5)	1.047E+02(4)	1.491E+02(6)	1.034E+02(1)
	<i>SD</i>	8.472E-01	8.215E-01	1.104E+00	9.304E-01	2.771E+01	7.031E-01
	<i>p-value (h-value)</i>	2.473E-04 (+)	2.302E-01 (=)	1.873E-09 (+)	2.070E-07 (+)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-3.6650 (272)	-1.1998 (535)	-6.0084 (22)	-5.1929 (109)	-6.2146 (0)	
F26	<i>Mean (Mean_rank)</i>	5.527E+03(6)	4.759E+03(5)	1.673E+03(3)	1.602E+03(2)	2.100E+03(4)	1.550E+03(1)
	<i>SD</i>	3.988E+03	3.987E+03	1.083E+03	1.571E+03	1.121E+03	1.296E+03
	<i>p-value (h-value)</i>	2.905E-06 (+)	1.425E-05 (+)	6.528E-01 (=)	6.528E-01 (=)	2.220E-02 (+)	
	<i>z-value (signedrank)</i>	-4.6774 (164)	-4.3399 (200)	-0.4499 (615)	0.4499 (711)	-2.2871 (419)	
F27	<i>Mean (Mean_rank)</i>	1.00E+02(1)	1.00E+02(1)	1.00E+02(1)	1.00E+02(1)	2.821E+04(2)	1.00E+02(1)
	<i>SD</i>	1.385E-07	8.716E-13	6.668E-10	0.00E+00	7.551E+03	3.565E-13
	<i>p-value (h-value)</i>	5.145E-10 (+)	6.380E-02 (=)	5.145E-10 (+)	1.573E-01 (=)	5.145E-10 (+)	
	<i>z-value (signedrank)</i>	-6.2146 (0)	-1.8537 (10)	-6.2146 (0)	-1.4142 (0)	-6.2146 (0)	
F28	<i>Mean (Mean_rank)</i>	1.00E+02(1)	1.00E+02(1)	1.00E+02(1)	1.00E+02(1)	1.282E+02(2)	1.00E+02(1)

<i>SD</i>	1.712E-10	4.295E-13	2.422E-10	1.435E-13	7.414E+00	1.435E-13
<i>p-value (h-value)</i>	5.139E-10 (+)	2.034E-07 (+)	5.127E-10 (+)	1.000E+00 (=)	5.145E-10 (+)	
<i>z-value (signedrank)</i>	-6.2148 (0)	-5.1962 (0)	-6.2152 (0)	— (0)	-6.2146 (0)	
<i>Average Mean_rank</i>	2.4667	2.5333	3.8000	3.0000	4.6667	1.7333
+ / = / -	10 / 4 / 0	7 / 5 / 3	10 / 4 / 2	7 / 5 / 3	13 / 1 / 1	

258 **Searching accuracy:**

259 From [Table 2](#) and [Table 3](#), we can conclude that SCAA performs the best searching accuracy as it
260 outperforms its peers with a large margin in most of the problems. SCAA exhibits 11 best *Mean* values out of
261 13 for conventional problems ([Table 2](#)) and obtains 10 best *Mean* values out of 15 for CEC2015 functions
262 ([Table 3](#)). Specifically, for the conventional problems, SCAA ranks the first on 5 multimodal problems (F8,
263 F10-F13). These functions have multiple local optima around the global optimum, which can mislead the
264 swarm into the non-optimal basin. While in SCAA, thanks to the proposed dynamic α adjustment strategy, it
265 has successfully enabled the population to track more promising regions and realize the balance between
266 exploration and exploitation on multimodal functions.

267 As for the CEC2015 functions, it is noticeable that all involved algorithms suffer from performance
268 degradation and none of them can find the acceptable solutions. Even so, SCAA achieves most number of best
269 *Mean* values compared with other algorithms. To be specific, the performance of SCAA is superior on the
270 unimodal functions (F14 and F15), hybrid functions (F19-F21) and composition functions (F22-F28). However,
271 SCAA performs mediocly on the multimodal functions (F16-F18) in which SCAA ranks 2th, 4th and 3th out of
272 6 algorithms on F16, F17 and F18, respectively. These functions possess a huge number of local optima [\[26\]](#)

273 that are more complex than the conventional multimodal functions. The mediocre performance of SCAA may come
274 from its α boundary constraint, which may block agents' jumping-out momentum to explore the whole search
275 space. Therefore, this limitation can be further explored in the future work.

276 According to the statistic results in the last row of [Table 2](#) and [Table 3](#), the involved algorithms obtain different
277 *rank* values on conventional problems and CEC2015 functions, which validate their distinguishing search
278 performance. Nevertheless, SCAA has the minimum value in *Average rank* and obtains the first place of results in
279 both tables, which confirm the superior searching accuracy of SCAA. In addition, SCAA performs relatively robust
280 towards the 28 benchmark functions as evidenced by its smallest *SD* values.

281 **Wilcoxon signed ranks test:**

282 From Tables 2 and 3, it is apparent that SCAA has significant better performance compared to its peers in most
283 cases. For the conventional problems, SCAA are significantly better than GSA, MGSA- α , FuzzyGSA,
284 FS α (Increase) and FS α (Decrement) on 9, 9, 10, 6 and 10 functions, respectively. FS α (Increase) and
285 FS α (Decrement) are statistically better than SCAA only on one function (F4), while GSA, MGSA- α and
286 FuzzyGSA cannot statistically outperform SCAA on any conventional functions. As for the CEC2015 functions,
287 when compared with GSA, MGSA- α , FuzzyGSA, FS α (Increase) and FS α (Decrement), SCAA shows
288 significantly better performance on 10, 7, 10, 7 and 13 functions and exhibits statistically worse performance just
289 on 1, 3, 2, 3 and 1 functions, respectively. These statistic results have validated the competitive advantages of the
290 proposed SCAA approach.

Table 4

Convergence speed and reliability comparison among six algorithms on conventional functions at 30-D.

	metrics	GSA	MGSA- α	FuzzyGSA	FS α (Increase)	FS α (Decrement)	SCAA
F1	<i>SP (SR%)</i>	8.03E+04 (100)	6.93E+04 (100)	5.64E+04 (100)	4.43E+04 (100)	2.47E+04 (100)	2.41E+04 (100)
	<i>runtime</i>	14.8011	12.7453	17.4103	10.8446	7.1359	6.6472
F2	<i>SP (SR%)</i>	1.52E+05 (100)	1.24E+05 (100)	1.03E+05 (100)	7.16E+04 (100)	5.14E+04 (100)	3.90E+04 (100)
	<i>runtime</i>	30.5721	24.3482	34.4027	18.1258	13.3601	12.3655
F3	<i>SP (SR%)</i>	8.65E+04 (33.3)	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)
	<i>runtime</i>	38.1149	Inf	Inf	Inf	Inf	Inf
F4	<i>SP (SR%)</i>	1.18E+05 (100)	9.79E+04 (100)	8.17E+04 (100)	5.88E+04 (100)	3.80E+04 (100)	7.96E+04 (100)
	<i>runtime</i>	23.2371	18.2716	25.4612	14.6428	9.6580	20.3386
F5	<i>SP (SR%)</i>	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)
	<i>runtime</i>	Inf	Inf	Inf	Inf	Inf	Inf
F6	<i>SP (SR%)</i>	2.65E+04 (100)	2.88E+04 (100)	1.79E+04 (100)	1.94E+04 (100)	1.56E+04 (100)	1.09E+04 (100)
	<i>runtime</i>	4.9064	5.0853	26.0255	4.7630	1.7719	3.0915
F7	<i>SP (SR%)</i>	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)
	<i>runtime</i>	Inf	Inf	Inf	Inf	Inf	Inf
F8	<i>SP (SR%)</i>	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)
	<i>runtime</i>	Inf	Inf	Inf	Inf	Inf	Inf
F9	<i>SP (SR%)</i>	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)	Inf (0)
	<i>runtime</i>	Inf	Inf	Inf	Inf	Inf	Inf
F10	<i>SP (SR%)</i>	1.28E+05 (100)	1.04E+05 (100)	8.65E+04 (100)	6.12E+04 (100)	4.11E+04 (100)	3.84E+04 (100)
	<i>runtime</i>	25.5664	20.7946	28.3058	15.4832	10.9001	9.8405
F11	<i>SP (SR%)</i>	5.77E+04 (100)	5.24E+04 (100)	4.04E+04 (100)	3.49E+04 (100)	Inf (0)	3.51E+04 (100)
	<i>runtime</i>	11.4403	10.2226	22.7925	9.0093	Inf	13.2134
F12	<i>SP (SR%)</i>	4.51E+04 (100)	4.19E+04 (100)	3.12E+04 (100)	2.89E+04 (100)	1.07E+04 (100)	2.94E+04 (100)
	<i>runtime</i>	13.2818	11.9072	16.0448	10.7113	3.8040	7.4324
F13	<i>SP (SR%)</i>	6.62E+04 (100)	5.65E+04 (100)	4.32E+04 (100)	3.83E+04 (100)	2.01E+04 (100)	1.38E+04 (100)
	<i>runtime</i>	14.8885	12.6030	15.3931	10.9785	5.8154	9.8506
Number of smallest <i>SP</i>		1	0	0	1	2	5

Average <i>runtime</i>	19.6454	14.4905	23.2295	11.8198	7.4923	10.3474
------------------------	---------	---------	---------	---------	---------------	---------

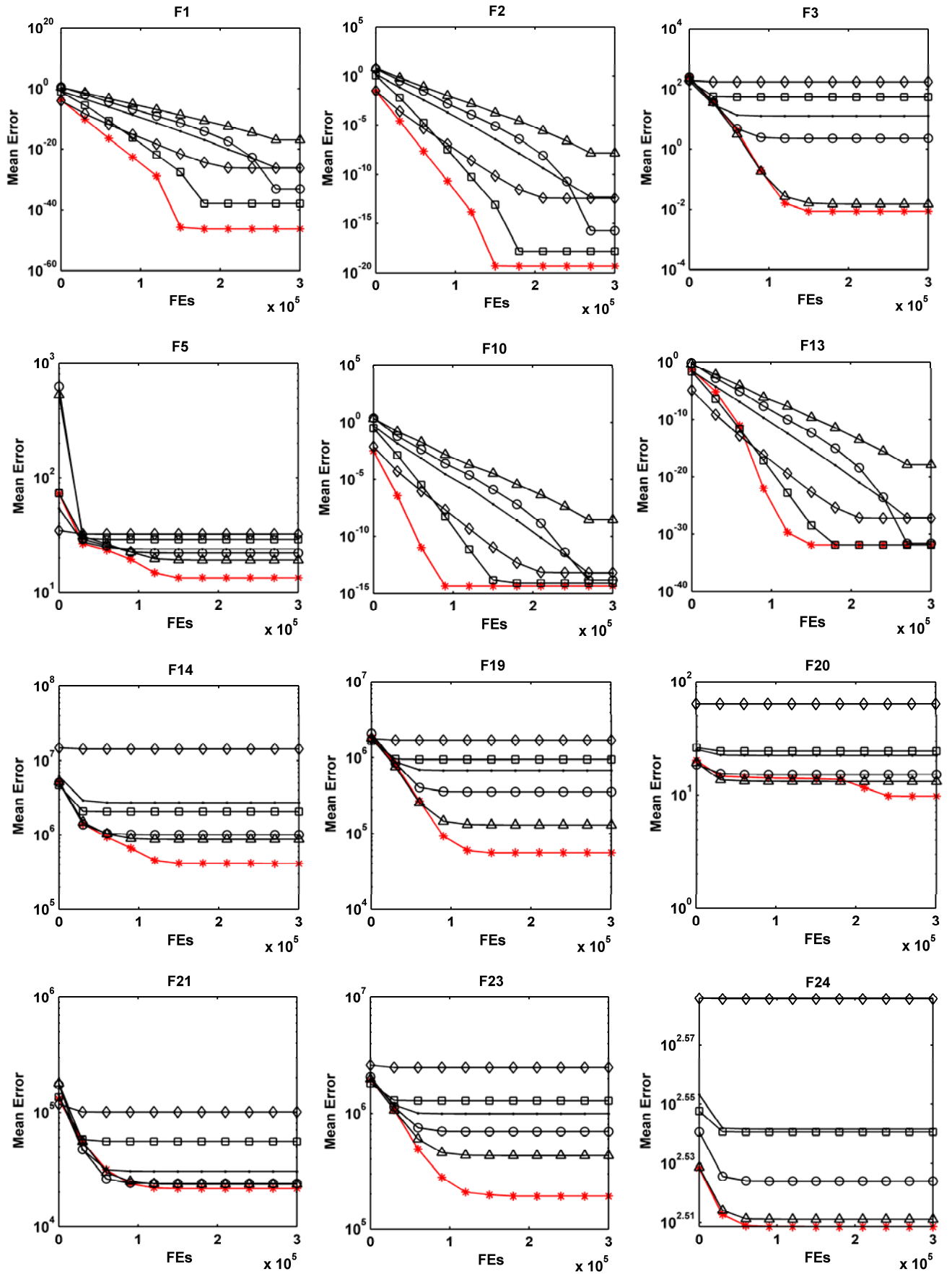
291 **Convergence comparison:**

292 The speed in obtaining an acceptable solution is also a salient yardstick for measuring the performance of an
 293 algorithm. For testing the searching efficiency of SCAA, the metric *SP*, *runtime* and convergence curves are also
 294 reported for comparison. It is worth mentioning that if one algorithm cannot solve the problem ($SR=0\%$), the *SP*
 295 value is defined as infinity (Inf). Because of the strict threshold settings, none of the involved algorithms can
 296 achieve the available *SP* values on all CEC2015 functions. Thus, only the results on 13 conventional functions are
 297 recorded. As shown in Table 4, SCAA achieves the smallest *SP* values on 5 conventional problems (F1, F2, F6, F10
 298 and F13), which is obviously better than that of its 5 peers. The fast convergence characteristic of SCAA mainly
 299 results from its adaptive adjustment of the parameter alpha.

300 As for the *runtime* results, SCAA has a tendency to spend slightly more computational time due to its repeating
 301 calculation of the ratio of mass and distance between any two agents in the stability conditions in each iteration.
 302 Nevertheless, from Table 4 we can conclude that SCAA spends the least computational time on unimodal functions
 303 F1 and F2. These results may benefit from the superior *SP* results of SCAA, which can decrease its *runtime* values
 304 to some extent. For more complicated functions, superiority on efficiency of SCAA is not as obvious as that on F1
 305 and F2. This is mainly because that the global optimum in these functions can be very far away from the local
 306 optima or can be surrounded by a considerable amount of local optima [4]. In this situation, for avoiding trapping
 307 into the local optima, the parameter alpha in SCAA is adjusted to decrease the convergence tendency of agents to
 308 the elite masses, which causes the increase of execution time. Even so, the average *runtime* of SCAA ranks the

309 second among the competing algorithms as listed at the bottom of [Table 4](#). In this experiment, FS α (Decrement)
310 ranks the first according to its average *runtime*. This is primarily because the decreasing trend of α from an initial
311 value 150 enables FS α (Decrement) to have a much larger α value during the whole iteration, which can improve
312 the convergence speed. However, the larger alpha value can lead to the smaller search steps of agents and easily
313 cause the premature convergence. This is observed by the poor searching performance of FS α (Decrement) as
314 shown in [Table 2](#) of the manuscript.

315 A closer look at the convergence curves of different algorithms in [Fig. 3](#) provides more insight into their
316 searching behavior. The figure includes the representative conventional problems (F1, F2, F3, F5, F10 and F13) and
317 CEC2015 functions (F14, F19, F20, F21, F23 and F24). Based on the graphical results in [Fig. 3](#), the outstanding
318 convergence characteristics of SCAA on different test functions are explained. Specifically, for the conventional
319 problems, the convergence curves of SCAA sharply drop at one point on functions F1, F2, F3, F5, F10 and F13 in
320 the early iterations and then reach better results in the latter iterations. These observations prove the capability of
321 SCAA to facilitate the balance between exploration and exploitation. With regard to the CEC2015 functions,
322 because of their complex characteristics, the convergence speeds of all involved algorithms are slightly inferior to
323 that in conventional problems. Nevertheless, the convergence curves of SCAA rapidly drop at one point on
324 functions F14, F19, F23 and F24 in the early iterations. On functions F20 and F21, the convergence speed of SCAA
325 is more slowly than its peers at the early stage, while its speed increases in the middle stage of optimization.
326 Moreover, SCAA achieves superior convergence accuracy among all algorithms, which may result from the ability
327 of α boundary constraint to improve convergence precision in the latter iterations. In general, SCAA has produced
328 improved searching efficiency compared with other algorithms.



—*— SCAA —△— GSA —○— MGSA- α —■— FuzzyGSA —□— FS α (Increase) —◇— FS α (Decrement)

Fig. 3. Convergence curves of different alpha adjusting GSA variants.

329 *4.4. Stability analysis of SCAA*

330 In order to validate the stability of SCAA, the trajectories of agents are analyzed. Experiments are conducted on
331 three conventional functions F2, F7 and F9, respectively. For a better observation, the number of iterations is set to
332 100 and the trajectory of the first dimension for the first agent, denoted as x_{11} in each algorithm, is recorded.
333 Experiments are carried out with 51 independent runs and the average trajectory curves are listed in Fig. 4. Note
334 that the stability stage occurs when there is little variation in an agent's position between iterations [8,18].

335 First, in order to analyze the effect of stability boundary constraint, we compare the performance of SCAA in
336 two cases: SCAA with stability constraints and SCAA without stability constraints (denoted as SCAA-Ns). From
337 the trajectory curves in Fig. 4 (a)-(c), it is clear that the agent's trajectory in red line (SCAA) are more stable than
338 that in blue line (SCAA-Ns), which validates that the stability constraints play an important role in guaranteeing the
339 stable convergence.

340 Second, we make the stability comparison between SCAA and other involved algorithms. The results are
341 plotted in Fig. 4 (d)-(f). It can be observed that SCAA performs more stable when compared with other algorithms.
342 To be specific, the agent trajectories of GSA, FS α (Increase) and MGSA- α are rather unstable, whereas SCAA
343 and FS α (Decrement) are more stable. Particularly, the agent trajectories of FS α (Decrement) is more stable than
344 that in SCAA. This may because that the decreasing trend of α from an initial value 150 in FS α (Decrement)

345 makes it have a much larger α value than its peers in most evolutionary steps. However, this method seems to
 346 possess a bad searching performance in Section 4.3 as it fails to maintain the population diversity.

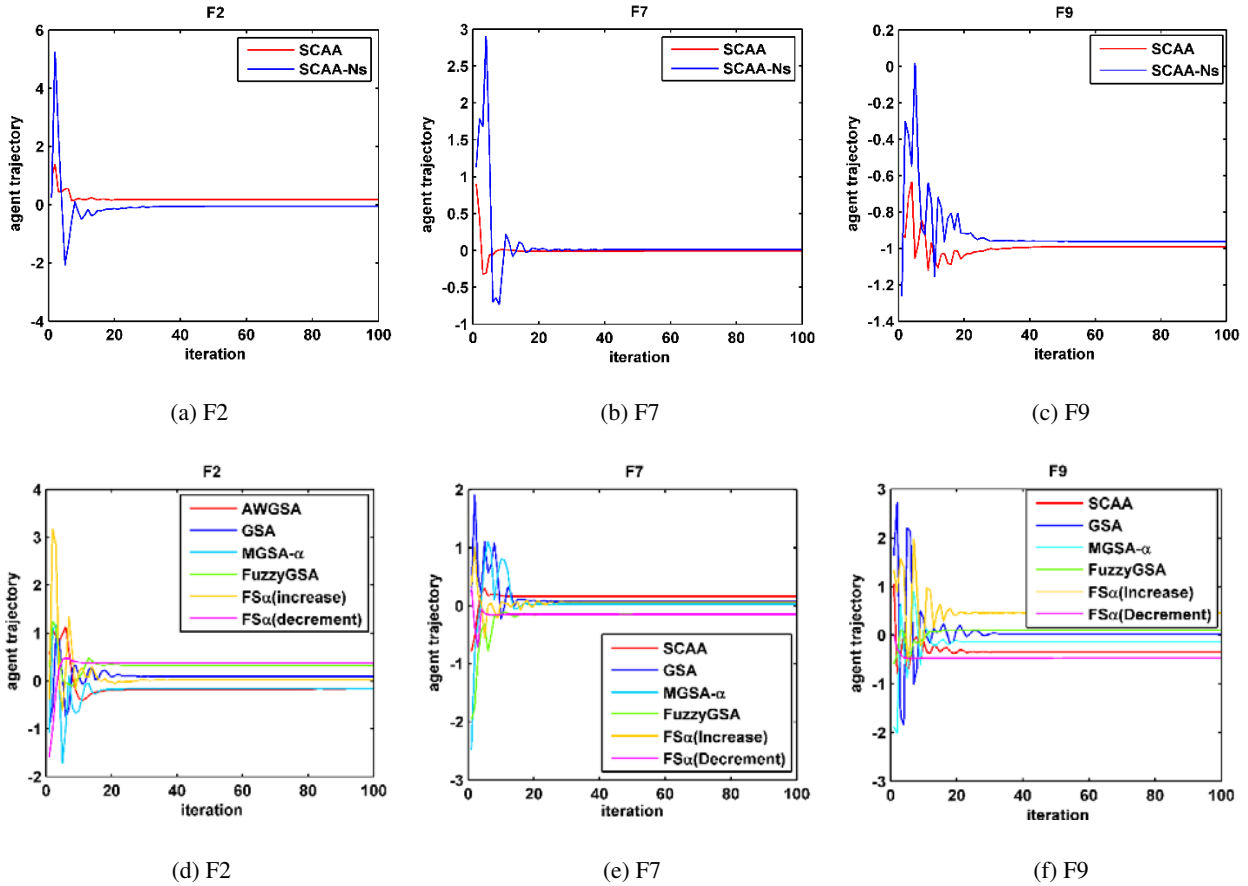


Fig. 4. Agent trajectories of different algorithms.

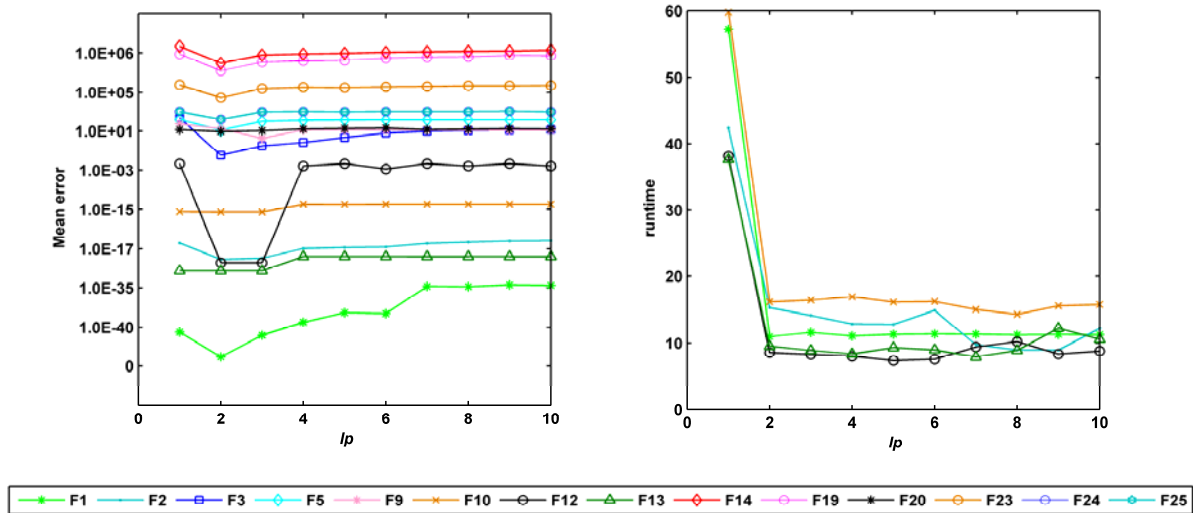
347 *4.5. Sensitivity analysis of key parameters*

348 The lp and α_{\max} are two key parameters that affect the searching performance of SCAA. To evaluate the
 349 impact of lp and α_{\max} on different kinds of functions, we conduct parameter sensitivity analysis on 30
 350 dimensional versions of the 14 representative functions including 4 unimodal conventional functions (F1, F2, F3,

351 F5), 4 multimodal conventional functions (F9, F10, F12, F13) and 6 rotated and shifted functions (F14, F19, F20,
 352 F23-F25) in this study.

353 We first carried out experiments with $lp = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ with all other parameters set as discussed
 354 in Section 4.1. Then we perform sensitivity analysis experiment of parameter α_{\max} with $\alpha_{\max} = [40, 50, 60, 70, 80,$
 355 $90, 100, 110, 120, 130]$, where the settings of other parameters are also kept the same as suggested in Section 4.1.
 356 For both of the two experiments, the average optimization error (*Mean*) and the average *runtime* of 51 independent
 357 runs obtained from each value of lp and α_{\max} are shown in Fig. 5 and Fig. 6, respectively. Note that due to the
 358 rigorous threshold setting of CEC2015 functions, only the *runtime* value of SCAA on conventional functions are
 359 reported.

360 (1) Sensitivity analysis of the parameter lp



(a) The *Mean* values of SCAA with different lp

(b) The *runtime* values of SCAA with different lp

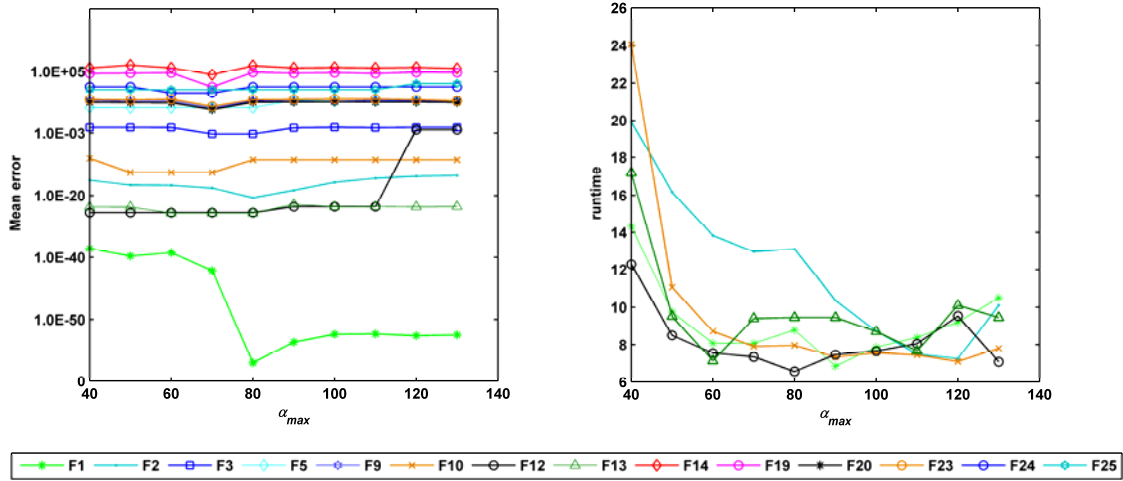
Fig. 5. Effects of lp on the performance of SCAA on conventional functions at 30-D. (The vertical axis of (a) is not to scale for easy presentation of the data).

361 In SCAA, the parameter lp determines the frequency of alpha changing. A suitable lp can activate the alpha
362 changing of stagnated agents without wasting too much computational cost. A too small lp value may cause agent's
363 oscillation and disturb the swarm convergence, while a too large value for lp will result in the waste of computation
364 on the local optima and lead to a premature convergence.

365 From Fig. 5, it is apparent that the search accuracy (Mean) and efficiency (runtime) of SCAA are affected by
366 the parameter lp . As shown in Fig. 5 (a), SCAA achieves the best *Mean* results when $lp=2$ on unimodal
367 conventional functions. This is mainly owing to the fact that these functions have only one minimum in the search
368 space. Thus, a small lp can make agents quickly adjust the search direction to detect the promising regions and then
369 rapidly enlarge the alpha value to emphasize exploitation around good solutions. With respect to the multimodal
370 conventional functions, the setting $lp=3$ is most suitable for SCAA as shown in Fig. 5 (a). This is probably because
371 that a small lp may lead to the oscillation of agents in local optima and limit the probabilities to find other
372 promising areas. Thus, on the multimodal functions, a relatively larger setting $lp=3$ is recommended in this paper.
373 For the rotated and shifted functions, SCAA performs the best searching accuracy when $lp=2$. These functions are
374 more complex than the conventional functions, where most of them are asymmetrical and have different properties
375 around different local optima [26]. In this situation, a small lp can sufficiently monitor the problem environment
376 and quickly change the search behavior of agents.

377 In terms of the *runtime* as shown in Fig. 5 (b), it is obvious that the execution time of SCAA is significantly
 378 reduced when lp is increased from 1 to 2 for most testing functions. When the lp increases from 2 to 10, the
 379 differences of *runtime* on each function is minor. Considering that the search accuracy of SCAA becomes worse
 380 when $lp \geq 6$ as shown in Fig. 5 (a), we recommend $lp=2, 3$ and 2 for the unimodal conventional functions,
 381 multimodal conventional functions, rotated and shifted functions respectively in this paper.

382 (2) Sensitivity analysis of the parameter α_{max}



(a) The Mean values of SCAA with different α_{max}

(b) The runtime values of SCAA with different α_{max}

Fig. 6. Effects of α_{max} on the performance of SCAA on conventional functions at 30-D. (The vertical axis of (a) is not to scale for easy presentation of the data).

383 From Fig. 6, it can be seen that the performance of SCAA is also affected by the parameter α_{max} . A smaller
 384 α_{max} may slow down the convergence of SCAA while a larger one may lead to the searching stagnation and
 385 prematurity problem. As plotted in Fig. 6 (a), the searching accuracy of SCAA is the best when $\alpha_{max}=80$ on
 386 unimodal conventional functions. This is mainly due to that a larger α_{max} can enhance the hill-climbing

387 performance on functions with only one minimum. When SCAA is applied to solve the multimodal functions and
388 the rotated and shifted functions, the setting $\alpha_{\max}=70$ becomes the best choice. These functions have lots of
389 minimum and are more complex than the unimodal functions. Thereby a smaller values of α_{\max} may ensure the
390 global search abilities of agents and enables them to escape from the local optima.

391 As for the runtime results, it is apparent that the execution time of SCAA is dramatically reduced in most cases
392 when α_{\max} is increased from 40 to 50. When $\alpha_{\max} \geq 50$, the differences of runtime on each function is minor.
393 These runtime distribution results support the analysis that a larger α_{\max} is beneficial for the convergence speed.
394 However, if the value of α_{\max} becomes too large, it will lead to high *Mean* errors as reported in [Fig. 6 \(a\)](#).
395 Considering both the search accuracy and efficiency of SCAA, we set α_{\max} to 80, 70 and 70 for unimodal
396 conventional functions, multimodal conventional functions, rotated and shifted functions, respectively in this
397 paper.

398 **5. Conclusion and further study**

399 In this paper, we proposed a stability constrained adaptive alpha for GSA (SCAA) to enhance the search
400 performance of the original GSA. In SCAA, the evolutionary state of each agent was first estimated. Then, the
401 variation of an agent's position and fitness is employed as feedback to guide the adjustment of α according to its
402 current state. This adaptive alpha adjusting strategy can enhance the convergence when an agent finds a promising
403 direction and relieves the premature problem when an agent moves to a local optimum. In addition, a boundary

404 constraint derived from the stability conditions was put forward to restrict agents' α values in each iteration,
405 which has guaranteed agents' stable trajectories and improved the precision of convergence.

406 To verify the performance of SCAA, 28 benchmark functions including conventional problems and CEC2015
407 functions were tested in this paper. Simulation results and comparisons have clearly showed the superiority of the
408 proposed SCAA over the original GSA and other alpha adjusting algorithms in terms of the searching accuracy,
409 searching reliability and searching efficiency. Besides, the stability analysis has demonstrated the effect of α
410 boundary constraint and validated the stability of SCAA.

411 One area in which SCAA falls short is the constant setting of the alpha upper boundary α_{\max} , which lacks the
412 time-varying characteristics as the lower boundary does. This reveals an area where SCAA could be further
413 improved in the future work. Another potential direction for improvement is to consider the adaptive information
414 systems, which have received increasing attention and are widely used for different applications, such as customer
415 churn prediction [29], cloud computing [16], case-based reasoning [30] and clustering of uncertain data [28]. A
416 number of adaptive metaheuristics techniques have been applied to control key parameters for improving the
417 optimization performance. For example, in [38,51], an improved version of the teaching-learning-based
418 optimization (TLBO) algorithm was proposed, in which an adaptive teaching factor was considered. Similarly, in
419 [41], a new variant of TLBO was presented by integrating a self-adaptive strategy for population sizing. Besides,
420 Tejani et al. [50] introduced adaptive benefit factors into the symbiotic organisms search (SOS) to enhance its
421 searching efficiency. Jia et al. [15] put forward an improved cuckoo search (ICS) algorithm by employing adaptive
422 technique in the step length of levy flight and discovering probability. R Sridhar proposed an adaptive genetic
423 algorithm to optimize the bin packing problem [44]. How to combine GSA with adaptive systems may have great

424 potential to further improve the optimization performance, where SCAA can also be applied to more real-world
425 optimization problems.

426 **Acknowledgements**

427 This work was supported by Chinese Natural Science Foundation Projects (No. 41471353), Fundamental
428 Research Funds for the Central Universities (No. 15CX06001A).

429 **References**

- 430 [1] H. Askari, S.-H. Zahiri, Decision function estimation using intelligent gravitational search algorithm, *Int. J. Mach.*
431 *Learn. Cybern.* 3 (2) (2012) 163-172, <http://dx.doi.org/10.1007/s13042-011-0052-x>.
- 432 [2] M.S.I.B. Aziz, S.W. Nawawi, S. Sudin, N. Abd Wahab, Exploitation selection of alpha parameter in Gravitational
433 Search Algorithm of PID controller for computational time analysis, in: *Control System, Computing and Engineering*
434 *(ICCSCE)*, 2014 IEEE International Conference on, IEEE, 2014, pp. 112-117,
435 <http://dx.doi.org/10.1109/ICCSCE.2014.7072699>.
- 436 [3] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic*
437 *algorithms*, Oxford university press, 1996, <http://dx.doi.org/10.1108/k.1998.27.8.979.4>.
- 438 [4] Z. Chen, X. Yuan, H. Tian, B. Ji, Improved gravitational search algorithm for parameter identification of water turbine
439 regulation system, *Energy Conversion & Management* 78 (30) (2014) 306-315,
440 <http://doi.org/10.1016/j.enconman.2013.10.060>.
- 441 [5] P. Civicioglu, Artificial cooperative search algorithm for numerical optimization problems, *Inf. Sci.* 229 (6) (2013)
442 58-76, <http://dx.doi.org/10.1016/j.ins.2012.11.013>.
- 443 [6] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a
444 methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* 1 (1) (2011)
445 3-18, <http://dx.doi.org/10.1016/j.swevo.2011.02.002>.
- 446 [7] M. Doraghinejad, H. Nezamabadi-Pour, Black hole: A new operator for gravitational search algorithm, *Int. J. Comput.*
447 *Intell. Syst.* 7 (5) (2014) 809-826, <http://dx.doi.org/10.1080/18756891.2014.966990>.
- 448 [8] F. Farivar, M.A. Shoorehdeli, Stability analysis of particle dynamics in gravitational search optimization algorithm,
449 *Inf. Sci.* 337 (2016) 25-43, <http://dx.doi.org/10.1016/j.ins.2015.12.017>.

- 450 [9] S. Gao, C. Vairappan, Y. Wang, Q. Cao, Z. Tang, Gravitational search algorithm combined with chaos for
451 unconstrained numerical optimization, *Appl. Math. Comput.* 231 (2014) 48-62,
452 <http://dx.doi.org/10.1016/j.amc.2013.12.175>.
- 453 [10] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the
454 evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization,
455 *J. Heuristics.* 15 (6) (2008) 617-644, <http://dx.doi.org/10.1007/s10732-008-9080-4>.
- 456 [11] B. González, F. Valdez, P. Melin, G. Prado-Arechiga, Fuzzy logic in the gravitational search algorithm for the
457 optimization of modular neural networks in pattern recognition, *Expert. Syst. Appl.* 42 (14) (2015) 5839-5847,
458 <http://dx.doi.org/10.1016/j.eswa.2015.03.034>.
- 459 [12] Z. Guo, A Hybrid Optimization Algorithm Based on Artificial Bee Colony and Gravitational Search Algorithm,
460 *International Journal of Digital Content Technology & Its Applications* 6 (17) (2012) 620-626,
461 <http://doi.org/10.4156/jdcta.vol6.issue17.68>.
- 462 [13] X.H. Han, X.M. Chang, L. Quan, X.Y. Xiong, J.X. Li, et al, Feature subset selection by gravitational search
463 algorithm optimization, *Inf. Sci.* 281 (2014) 128–146, <http://doi.org/10.1016/j.ins.2014.05.030>.
- 464 [14] X.H. Han, L. Quan, X.Y. Xiong, B. Wu, Facing the classification of binary problems with a hybrid system based on
465 quantum-inspired binary gravitational search algorithm and K-NN method, *Eng. Appl. Artif. Intel* 26 (10) (2013)
466 2424-2430, <http://doi.org/10.1016/j.engappai.2013.05.011>.
- 467 [15] B. Jia, B. Yu, Q. Wu, C. Wei, R. Law, Adaptive affinity propagation method based on improved cuckoo search,
468 *Knowl-based. Syst.* 111 (C) (2016) 27-35, <http://dx.doi.org/10.1016/j.knosys.2016.07.039>.
- 469 [16] J. Jiang, J. Lu, G. Zhang, An Innovative Self-Adaptive Configuration Optimization System in Cloud Computing, in:
470 *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, 2011, pp. 621-627,
471 <http://dx.doi.org/10.1109/dasc.2011.112>.
- 472 [17] S. Jiang, Z. Ji, Y. Shen, A novel hybrid particle swarm optimization and gravitational search algorithm for solving
473 economic emission load dispatch problems with various practical constraints, *Int. J. Elec. Power* 55 (2) (2014)
474 628-644, <http://dx.doi.org/10.1016/j.ijepes.2013.10.006>.
- 475 [18] S. Jiang, Y. Wang, Z. Ji, Convergence analysis and performance of an improved gravitational search algorithm, *Appl.*
476 *Math. Comput.* 24 (2014) 363-384, <http://dx.doi.org/10.1016/j.asoc.2014.07.016>.
- 477 [19] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization
478 problems, in: *International Fuzzy Systems Association World Congress*, Springer, Berlin/Heidelberg, 2007, pp.
479 789-798, http://dx.doi.org/10.1007/978-3-540-72950-1_77.
- 480 [20] M. Khajehzadeh, M.R. Taha, A. El-Shafie, M. Eslami, A modified gravitational search algorithm for slope stability
481 analysis, *Eng. Appl. Artif. Intel* 25 (8) (2012) 1589-1597, <http://dx.doi.org/10.1016/j.engappai.2012.01.011>.
- 482 [21] S.U. Khan, S. Yang, L. Wang, L. Liu, A Modified Particle Swarm Optimization Algorithm for Global Optimizations
483 of Inverse Problems, *IEEE Transactions on Magnetics* 52 (3) (2016) 1-4,
484 <http://dx.doi.org/10.1109/TMAG.2015.2487678>.
- 485 [22] C. Li, L. Chang, Z. Huang, Y. Liu, N. Zhang, Parameter identification of a nonlinear model of hydraulic turbine
486 governing system with an elastic water hammer based on a modified gravitational search algorithm, *Eng. Appl. Artif.*
487 *Intel* 50 (2016) 177-191, <http://dx.doi.org/10.1016/j.engappai.2015.12.016>.
- 488 [23] C. Li, H. Li, P. Kou, Piecewise function based gravitational search algorithm and its application on parameter
489 identification of AVR system, *Neurocomputing* 124 (2014) 139-148,
490 <http://dx.doi.org/10.1016/j.neucom.2013.07.018>.
- 491 [24] X. Li, J. Wang, J. Zhou, M. Yin, An effective GSA based memetic algorithm for permutation flow shop scheduling,
492 in: *Evolutionary Computation*, 2010, pp. 1-6, <http://doi.org/10.1109/cec.2010.5586392>

- 493 [25] X. Li, M. Yin, Z. Ma, X. Li, Z. Ma, Hybrid differential evolution and gravitation search algorithm for unconstrained
494 optimization, *Int. J. Phys. Sci.* 6 (25) (2011) 5961-5981, <http://dx.doi.org/10.5897/IJPS11.029>.
- 495 [26] J. Liang, B. Qu, P. Suganthan, Q. Chen, Problem definition and evaluation criteria for the CEC 2015 competition on
496 learning-based real-parameter single objective optimization (2014), Computational Intelligence Laboratory,
497 Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore
498 (2014).
- 499 [27] W.H. Lim, N.A. Mat Isa, Teaching and peer-learning particle swarm optimization, *Appl. Math. Comput.* 18 (2014)
500 39-58, <http://dx.doi.org/10.1016/j.asoc.2014.01.009>.
- 501 [28] H. Liu, X. Zhang, X. Zhang, Y. Cui, Self-adapted mixture distance measure for clustering uncertain data,
502 *Knowl-based. Syst.* (2017) <http://dx.doi.org/10.1016/j.knsys.2017.04.002>.
- 503 [29] N. Lu, H. Lin, J. Lu, G. Zhang, A Customer Churn Prediction Model in Telecom Industry Using Boosting, *IEEE*
504 *Transactions on Industrial Informatics* 10 (2) (2014) 1659-1665, <http://dx.doi.org/10.1109/tii.2012.2224355>.
- 505 [30] N. Lu, J. Lu, G. Zhang, An Integrated Knowledge Adaption Framework for Case-Based Reasoning Systems, in:
506 *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2009, pp.
507 372-379.
- 508 [31] S. Mirjalili, S.Z.M. Hashim, A new hybrid PSO-GSA algorithm for function optimization, in: *Computer and*
509 *Information Application (ICCIA)*, 2010 International Conference on, 2010, pp. 374-377,
510 <http://dx.doi.org/10.1109/ICCIA.2010.6141614>.
- 511 [32] S. Mirjalili, A. Lewis, Adaptive gbest-guided gravitational search algorithm, *Neural. Comput. Appl.* 25 (7-8) (2014)
512 1569-1584, <http://dx.doi.org/10.1007/s00521-014-1640-y>.
- 513 [33] H. Nezamabadi-Pour, F. Barani, Gravitational Search Algorithm: Concepts, Variants, and Operators, *Modern*
514 *Optimization Algorithms and Applications in Engineering and Economics* (2016).
- 515 [34] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, *Swarm intelligence* 1 (1) (2007) 33-57,
516 <http://dx.doi.org/10.1007/s11721-007-0002-0>.
- 517 [35] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: A Gravitational Search Algorithm, *Inf. Sci.* 179 (13) (2009)
518 2232-2248, <http://dx.doi.org/10.1016/j.ins.2009.03.004>.
- 519 [36] F. Saeidi-Khabisi, E. Rashedi, Fuzzy gravitational search algorithm, in: *Proceedings 2nd international conference*
520 *on computer and knowledge engineering (ICCKE)*, 2012, pp. 156-160, <http://dx.doi.org/10.1109/iccke.2012.6395370>
- 521 [37] S. Sarafrazi, H. Nezamabadi-Pour, S. Saryazdi, Disruption: A new operator in gravitational search algorithm, *Sci*
522 *Iranica* 18 (3) (2011) 539-548, <http://dx.doi.org/10.1016/j.scient.2011.04.003>.
- 523 [38] V.J. Savsani, G.G. Tejani, V.K. Patel, Truss topology optimization with static and dynamic constraints using modified
524 subpopulation teaching-learning-based optimization, *Eng. Optimiz.* 48 (11) (2016) 1-17,
525 <http://dx.doi.org/10.1080/0305215x.2016.1150468>.
- 526 [39] S. Sheikhpour, M. Sabouri, S.H. Zahiri, A hybrid Gravitational search algorithm — Genetic algorithm for neural
527 network training, in: *Electr. Eng.*, 2013, pp. 1-5, <http://dx.doi.org/10.1109/iranianee.2013.6599894>.
- 528 [40] N. Siddique, H. Adeli, Gravitational Search Algorithm and Its Variants, *International Journal of Pattern Recognition*
529 *& Artificial Intelligence* (2015) <http://dx.doi.org/10.1142/S0218001416390018>.
- 530 [41] S. Sileasom, S. Bureerat, Four-Bar Linkage Path Generation through Self-Adaptive Population Size
531 Teaching-Learning Based Optimization, *Knowl-based. Syst.* (2017) <http://dx.doi.org/10.1016/j.knsys.2017.08.012>.
- 532 [42] M. Soleimanpour-Moghadam, H. Nezamabadi-Pour, An improved quantum behaved gravitational search algorithm,
533 in: *Electr. Eng.*, 2012, pp. 711-715, <http://dx.doi.org/10.1109/iranianee.2012.6292446>.
- 534 [43] A. Sombra, F. Valdez, P. Melin, O. Castillo, A new gravitational search algorithm using fuzzy logic to parameter
535 adaptation, in: *Evolutionary Computation (CEC)*, 2013 IEEE Congress on, IEEE, 2013, pp. 1068-1074,
536 <http://dx.doi.org/10.1109/CEC.2013.6557685>.

- 537 [44] R. Sridhar, M. Chandrasekaran, C. Sriramy, T. Page, Optimization of heterogeneous Bin packing using adaptive
538 genetic algorithm, (2017) 012026, <http://dx.doi.org/10.1088/1757-899x/183/1/012026>.
- 539 [45] R. Storn, K. Price, Differential evolution-a simple and efficient adaptive scheme for global optimization over
540 continuous spaces, ICSI Berkeley, 1995, <http://dx.doi.org/10.1063/1.4930637>.
- 541 [46] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, et al, Problem Definitions and Evaluation Criteria for the
542 CEC 2005 Special Session on Real-Parameter Optimization, Nanyang Technological University (2005).
- 543 [47] G. Sun, A. Zhang, A Hybrid Genetic Algorithm and Gravitational Search Algorithm for Image Segmentation Using
544 Multilevel Thresholding, in: Iberian Conference on Pattern Recognition and Image Analysis, Springer, Berlin
545 Heidelberg, 2013, pp. 707-714, http://dx.doi.org/10.1007/978-3-642-38628-2_84.
- 546 [48] G. Sun, A. Zhang, Z. Wang, Y. Yao, J. Ma, G.D. Couples, Locally informed Gravitational Search Algorithm,
547 Knowl-based. Syst. 104 (2016) 134-144, <http://dx.doi.org/10.1016/j.knosys.2016.04.017>.
- 548 [49] M. Taherkhani, R. Safabakhsh, A novel stability-based adaptive inertia weight for particle swarm optimization, Appl.
549 Math. Comput. 38 (2016) 281-295, <http://dx.doi.org/10.1016/j.asoc.2015.10.004>.
- 550 [50] G.G. Tejani, V.J. Savsani, V.K. Patel, Adaptive symbiotic organisms search (SOS) algorithm for structural design
551 optimization, Journal of Computational Design & Engineering 3 (3) (2016) 226-249,
552 <http://dx.doi.org/10.1016/j.jcde.2016.02.003>.
- 553 [51] G.G. Tejani, V.J. Savsani, V.K. Patel, Modified sub-population teaching-learning-based optimization for design of
554 truss structures with natural frequency constraints, Mechanics Based Design of Structures & Machines 44 (7) (2016)
555 págs. 495-513, <http://dx.doi.org/10.1080/15397734.2015.1124023>.
- 556 [52] J. Vijaya Kumar, D.M. Vinod Kumar, K. Edukondalu, Strategic bidding using fuzzy adaptive gravitational search
557 algorithm in a pool based electricity market, Appl. Math. Comput. 13 (5) (2013) 2445-2455,
558 <http://dx.doi.org/10.1016/j.asoc.2012.12.003>.
- 559 [53] J. Xiao, F. Qi, Y. Li, Gravitational chaotic search algorithm for partners selection with due date constraint in virtual
560 enterprise, in: Advanced Computational Intelligence (IWACI), 2011 Fourth International Workshop on, IEEE, 2011,
561 pp. 138-142, <http://dx.doi.org/10.1109/iwaci.2011.6159990>.
- 562 [54] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, Ieee. T. Evolut. Comput. 3 (2) (1999) 82-102,
563 <http://dx.doi.org/10.1109/4235.771163>.
- 564 [55] A. Zhang, G. Sun, J. Ren, X. Li, Z. Wang, X. Jia, A Dynamic Neighborhood Learning-Based Gravitational Search
565 Algorithm, IEEE Transactions on Cybernetics PP (99) (2016) 1-12.
- 566 [56] N. Zhang, C. Li, R. Li, X. Lai, Y. Zhang, A mixed-strategy based gravitational search algorithm for parameter
567 identification of hydraulic turbine governing system, Knowl-based. Syst. (2016)
568 <http://dx.doi.org/10.1016/j.knosys.2016.07.005>.
- 569