

A new metaheuristic algorithm: Shrimp and Goby association search algorithm and its application for damage identification in large-scale and complex structures

Thanh Sang-To ^{a,b}, Hoang Le-Minh ^a, Magd Abdel Wahab ^{c,*}, Cuong-Le Thanh ^{a,*}

^a Center for Engineering Application & Technology Solutions, Ho Chi Minh City Open University, Ho Chi Minh, Viet Nam

^b Soete Laboratory, Faculty of Engineering and Architecture, Ghent University, Technologiepark Zwijnaarde 903, Zwijnaarde B-9052, Belgium

^c Faculty of Mechanical - Electrical and Computer Engineering, School of Engineering and Technology, Van Lang University, Ho Chi Minh City, Viet Nam



ARTICLE INFO

Keywords:

SGA

Damage detection

High dimension

Complex structures

Optimization

ABSTRACT

This paper presents the application of a new Shrimp and Goby Association Search algorithm (SGA) to solve large-scale global optimization problems. The performance of SGA is assessed using 13 benchmark high-dimensional functions, 10 classical benchmark functions, and several real-world engineering applications. For the first time, an efficient optimization approach for structural health monitoring (SHM) in truss-like structures is presented. The proposed approach is applied for damage identification of complex structures. A real structure, namely Canton Tower in Guangzhou, China, is served as an example for damage detection. Interestingly, this tower was the tallest structure in the world in 2009 with a height of 610 m. The great merit of this example is that it provides a real complex structure with a high-dimensional problem to assess the performance of SGA in the real world. The results show that SGA can deal with this problem effectively, at the same time, it operates better to escape from local optima with faster convergence rate than population-based algorithms.

1. Introduction

There are more and more inverse analysis challenges [1], which motivate the advent of more advanced techniques to solve them. Artificial intelligence (AI) methods [2–4] have emerged as one of the most effective tools to solve many different types of problems.

In real ecological systems, there is an interesting type of intelligence, which consists of the collective behaviour of the natural ecology of the Earth, i.e. swarm intelligence. In the recent decades, there are diverse methods using swarm intelligence, which have been more popular to solve problems in various fields. Inspired by swarm-intelligence, researchers model the social behaviour of animals to handle different complex problems [5–13]. Besides, several natural evolution processes supporting the optimization techniques are the evolution of several generation of creatures [14], phenomenon of universe [15], human activity [16], etc.

This approach is extremely effective, robust and stable to solve not only complex continuous problems, but also discrete optimization problems in many kinds of search spaces [17–21]. The main aim of simulating swarm intelligence is to solving lively two main phases,

namely local search and global search of optimization problems instead of depending on pure mathematics.

For this reason, the use of metaheuristics to solve problems in all fields from science to technology, from economic to politics, etc. has become more popular than ever. In addition, the simplicity of understanding and applying such algorithms makes them usable in daily life easily and grown sharply in recent years.

This study's key contribution is the use of a method that mimics the shrimp-goby association for solving complex problems. It is demonstrated that the proposed solution can be used to resolve both straightforward and challenging high-dimensional single-objective problems. 13 high-dimensional cases (from 30 to 1000 dimensions), together with real engineering problems, are solved to evaluate this characteristic, and the solutions of the most effective optimization techniques are then compared to the findings. Besides, ten fixed-dimensional multimodal standard functions are also used to further evaluate the behaviour of the proposed technique.

The remainder of this article is summarized as follows. In Section 2, a thorough literature review is provided. The proposed SGA algorithm is presented in Section 3, while the results and computational complexity

* Corresponding authors.

E-mail addresses: magd.a.w@vlu.edu.vn, magd.abdelwahab@ugent.be (M. Abdel Wahab), cuong.lt@ou.edu.vn (C.-L. Thanh).

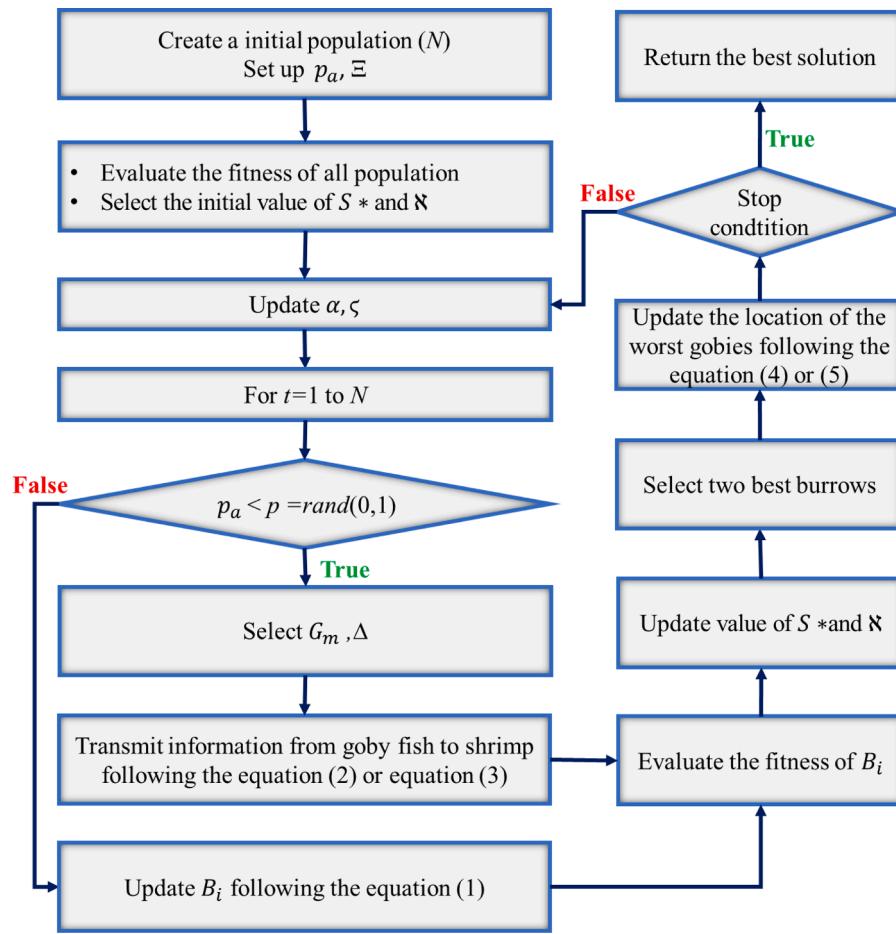


Fig. 1. Pseudo code of the SGA.

Table 1
Seven unimodal classical benchmark functions.

Function(F_i)	Range	f_{min}
$F_1 = \sum_{i=1}^n (x_i)^2$	[- 100, 100]	0
$F_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	[- 10, 10]	0
$F_3 = \sum_{i=1}^n (\sum_{j=1}^i (x_j))^2$	[- 100, 100]	0
$F_4 = \max_i \{ x_i , 1 \leq i \leq n\}$	[- 100, 100]	0
$F_5 = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[- 30, 30]	0
$F_6 = \sum_{i=1}^n (x_i + 0.5)^2$	[- 100, 100]	0
$F_7 = \sum_{i=1}^n ix_i^4 + \text{rand}[0, 1]$	[- 1.28, 1.28]	0

are provided in Section 4. Section 5 illustrates the application of SGA to solve real-world complex problems, and several conclusions are drawn in Section 6.

2. Literature review

The three primary types of optimization algorithms are evolutionary algorithms, physics-based algorithms, and swarm intelligence algorithms.

The concept of evolution in nature served as the inspiration for the first category. These algorithms are generally based on Darwin's ideas. These hypotheses are processes of optimization designed to increase an organism's capacity for survival in a changing environment. One of the elements that is essential to choose the ideal individual for the next

Table 2
Six multimodal classical benchmark functions.

Function(F_i)	Range	f_{min}
$F_8 = \sum_{i=1}^n - (x_i \sin \sqrt{ x_i })$	[- 500, 500]	- 418.9829
$F_9 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)^2$	[- 5.12, 5.12]	0
$F_{10} = - 20 \exp \left(- 0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) + 20 + e$	[- 32, 32]	0
$F_{11} = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \cup_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	[- 600, 600]	0
$F_{12} = \frac{\pi}{n} \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	[- 50, 50]	0
$F_{13} = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i + 0.5)^2 [1 + \sin^2(3\pi x_1 + 1)] + (x_n - 1)^2 [1 + \sin^2(3\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	[- 50, 50]	0

generation is its ability to survive in the living environment. In other words, many types of organisms that exist in a certain ecosystem change with time. Every living thing is viewed as nature's response to a query of how to adapt to live in a certain environment. Typical algorithms of this

Table 3

Ten fixed-dimension multimodal benchmark functions.

Function(F_i)	Range	Dim	f_{min}
$F_{14} = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{\sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	[- 65, 65]	2	1
$F_{15} = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_3 + x_4)}{b_i^2 + b_i x_3 + x_4} \right]^2$	[- 5, 5]	4	0.0003
$F_{16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 - x_1 x_2 - 4x_2^2 + 4x_2^4$	[- 5, 5]	5	-1.0316
$F_{17} = (x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 + 6) + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	[- 5, 5]	2	0.398
$F_{18} = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 48x_2 - 36x_1 x_2 + 72x_2^2)]$	[- 5, 5]	2	3
$F_{19} = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	[1, 3]	3	-3.86
$F_{20} = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	[0, 1]	6	-3.32
$F_{21} = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	-10.1532
$F_{22} = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	10.4028
$F_{23} = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	10.5363

type are: genetic algorithm (GA) [22], genetic programming [23], differential evolution (DE) [24,25], evolution strategy (ES) [26], etc.

The second group consists of physics-based optimization algorithms (i.e. Curved Space Optimization (CSO) [27], Planted Optimization Algorithm (POA) [28,29], Water Wave Optimization (WWO) [30]), in which the motion of particles is motivated by physical principles in magnetic fields, gravitational forces between galaxies' components, electron charge transfer, chemical interactions, etc.

Swarm-based algorithms make up the third type of meta-heuristic optimizers, namely particle swarm optimization (PSO) [5] algorithm, ant colony optimization (ACO) [31] algorithm, Cuckoo search (CS) [32, 33] algorithm, bat algorithm (BA) [34], whale optimization algorithm (WOA) [9], Salp Swarm Algorithm (SSA) [35], Dragonfly algorithm (DA) [36]. These algorithms frequently employ the swarm of particles, which prevents the detection and comparison of a single particle. The particles use collective communication to determine the reaction. A group of agents that are constantly moving and interacting throughout various tasks might be referred to as a swarm of particles. The study of the social behaviour of creatures and their interactions is often where meta-heuristic optimizers' problem-solving behaviour originates.

Despite the abundance of suggested methods, none of them can resolve every optimization issue. The No Free Lunch (NFL) [37] theorem has rationally demonstrated that this is the case. This is the reason why we try to construct a metaheuristic algorithm to enrich the selection for the challenging problems.

3. Shrimp and goby association search algorithm

Understanding how a survival-based model works in an obligatory partnership between goby and shrimp is vital for the inspiration to design a metaheuristic algorithm. Therefore, the first subsection is a real monitoring behaviour shrimps and goby based on literature reviews to create an appeal for the design. Next, we describe a mathematical model based on these behaviours of Shrimp and Goby.

3.1. Inspiration

Shrimps and Goby fishes live together as special evolution for their survival, and they are associated with each other like a mutually beneficial partnership. The goby employs the burrow dug by the shrimp as a shelter during the day and a permanent resting place at night. Meanwhile, the shrimp is utterly reliant on the goby to set an alarm system

against predators because of almost blind [38]. In particular, the more colourful the shrimp, the higher the chance of attracting a goby fish for cooperation. And hence, the more secure the locations with shrimp shelter, the more likely they are to attract gobies to reside. In other words, the degree of safety of a shelter depends on the shrimp's ability to attract the gobiid fish.

3.2. Mathematical model

In order to design SGA, a mathematical model is introduced to represent the associations between shrimp and gobiid fish. Herein, we assume the safest burrow like the fittest solution with the objective function (location: \vec{S}^*), and its value is N as shown Eq. (1).

$$f(\vec{S}^*) = \left\{ \vec{S}^* \in B | f(\vec{S}^*) = N \right\} = \min \left\{ \vec{K} \right\}, N \in \vec{K} \quad (1)$$

$$B = \begin{bmatrix} \vec{B}_i \end{bmatrix} = \begin{bmatrix} B_{1,1} & \cdots & B_{1,j} \\ \vdots & \ddots & \vdots \\ B_{i,1} & \cdots & B_{i,j} \end{bmatrix} \quad (2)$$

$$\vec{K} = \begin{bmatrix} K_1 \\ \dots \\ K_i \end{bmatrix} \quad (3)$$

where B is the space consisting of the location vectors for shrimps and gobiid fishes, and \vec{K} is the fitness of the objective function, in which goby plays an important role in an alarm system against predators for global search, and the burrowing shrimps digs a shelter for local search. SGA algorithm simulates the association shrimp-goby using the co-evolved partnership.

Goby fish is generally not associated with a specific shelter during their life cycle, but they change location frequently to shelter [39]. Because many of the shrimp species themselves are almost blind, the shrimps can attract only and be selected to become "preferred" gobiid partners via their bright colours when the gobies arrive nearly their burrow area. More brightly coloured shrimp will attract gobies to create an association relationship. This means that their chances of survival are higher, and therefore they, have better burrows to own burrows. Consequently, the best burrow will be an attractive destination for goby fishes. Eq. (4) is proposed to mathematically simulate this behaviour.

$$\vec{B}_i^{t+1} = \vec{B}_i^t + \alpha \times r_1 \otimes (\vec{S}^* - \vec{B}_i^t) \quad (4)$$

where t presents the current iteration, $\alpha \in [0, 1]$ is linear function, \vec{S}^* indicates the location vector of the best burrow, and \vec{B}_i is the position vector of a gobiid fish or shrimp, and $r_1 \in (0, 1)$ is a random vector. Meanwhile, α is a linearly decreased function that employs a mathematical model. It means α describes the range of movement of the goby over time.

3.2.1. A signal transmission and alarm system against predators (global search)

Kramer et al. [39] described a special communication mechanism between shrimp and gobiid fish. Antennae of shrimps are used to search a gobiid fish. While the shrimps are digging the burrow, the gobiid fish are always present to watch. And hence, gobiid fish will warn shrimp about the danger when predators approach for retreating into the shelter quickly.

In SGA, such signal transmission is mathematically modelled as follows: (1) If a shrimp is active outside the burrows, shrimps (B_i) request information from gobiid fishes (G_m) to ensure that its safety. (2) Via antenna contact of shrimp with the tail flicking or ventral fins signal, the goby will select (Δ) information to give its an answer. (3) The next decisions of the shrimp will depend on the received information about this communication mechanism. The special communication mechanism is

Table 4Comparison of SGA for the first 13 benchmark functions with $Dim = 30$.

<i>Fi</i>	<i>Measure</i>	<i>PSO</i>	<i>GA</i>	<i>GSA</i>	<i>SSA</i>	<i>BAT</i>	<i>DA</i>	<i>SGA</i>
F1	Worst	1.19115E-08	0.000250967	1.79766E-16	2.41858E-08	63638.35444	3095.22521	1.7867E-89
	Best	1.89903E-11	5.41649E-06	5.66383E-17	7.63573E-09	30234.50519	372.1630045	1.296E-100
	Mean	1.28281E-09	5.71873E-05	1.04877E-16	1.2054E-08	53202.03099	1164.693739	9.3586E-91
	Std	2.30806E-09	5.23581E-05	2.86612E-17	3.25521E-09	7037.946118	650.643065	3.4286E-90
F2	Worst	10.0000001	0.253162982	250.8925057	2.52462E+21	4.21401E+27	766.3530097	1.3598E-59
	Best	9.50173E-09	0.002688788	45.91301155	339.7052182	2.52073E+18	48.35702472	3.1707E-64
	Mean	0.6666666887	0.051231869	127.2070824	8.55737E+19	2.05425E+26	346.7037304	6.2227E-61
	Std	2.537081282	0.066074802	44.47436762	4.60726E+20	7.94073E+26	180.4523587	2.5027E-60
F3	Worst	453.5116986	7687.48212	677.6104966	930.3159528	170275.1266	25395.48345	1.63024573
	Best	96.79144857	766.90067	174.7702468	52.84865186	35995.80377	1539.589288	1.0721E-07
	Mean	248.403099	2559.664392	455.0021665	358.6635071	78619.29674	9669.454327	0.06969573
	Std	91.65872659	1574.684052	138.7534826	238.3921901	27942.25363	6120.98329	0.29729972
F4	Worst	6.13553401	73.04757996	5.5363887	15.14392538	83.09304886	39.94292681	7.3223E-11
	Best	1.831265971	33.78197818	1.35732E-08	2.332875962	67.0981496	10.23965403	3.2261E-18
	Mean	3.863481535	58.07284999	1.405357726	8.03283961	75.47627839	22.66981739	6.4992E-12
	Std	1.06726826	9.536301634	1.597831495	2.69417152	3.560209318	7.478534665	1.8003E-11
F5	Worst	4000009.635	19606.9025	69075.42518	680761.8993	5.70462E+11	743301966.8	107.567905
	Best	21.96830974	12.21364051	23.28631403	25.32779934	2.2545E+11	16879451.79	25.0739048
	Mean	134142.2413	3362.385806	2768.384952	45682.35674	3.86203E+11	254290251.7	28.7725397
	Std	730155.0421	6022.489556	12589.15826	138100.2246	93781131790	203572914.1	14.8961934
F6	Worst	4.12928E-09	0.000166887	1.61195E-16	1.9302E-08	66311.42065	2970.181837	0.00074149
	Best	8.41198E-12	5.87312E-06	3.88912E-17	7.62238E-09	37297.82185	387.6203679	0.00010624
	Mean	5.12463E-10	4.9505E-05	1.06951E-16	1.22682E-08	53348.42896	1228.703842	0.00041804
	Std	8.68229E-10	4.38114E-05	3.37806E-17	2.70846E-09	6972.301151	628.300811	0.00016302
F7	Worst	0.042572464	0.08199709	0.117445954	0.17769099	0.212836373	1.688885167	0.00280876
	Best	0.012379077	0.019662332	0.030778175	0.03193702	0.050963459	0.079870923	0.00031197
	Mean	0.022209552	0.049663077	0.060321258	0.097643832	0.127522346	0.365760168	0.00126848
	Std	0.007889558	0.013842154	0.022970283	0.039452127	0.036641845	0.300611738	0.0006706
F8	Worst	-6686.874731	-6818.965259	-1805.683906	-6063.825118	-2466.530255	-4390.212189	-8723.9917
	Best	-8937.323008	-9775.55817	-3232.71911	-8960.889442	-4647.751196	-7196.356508	-10440.659
	Mean	-7881.94393	-8351.408577	-2486.355459	-7407.754999	-3119.366664	-5528.740905	-9645.7983
	Std	638.4220522	648.621682	421.537923	625.1493589	502.0956666	657.4163508	455.968837
F9	Worst	86.56118553	235.8037252	43.77814305	101.4855512	285.5511728	308.1767868	1.6611E-06
	Best	25.8689205	80.59190232	11.93950365	28.85379755	143.2735333	88.14560159	0
	Mean	51.30640762	154.2844043	27.85883193	54.6563227	205.1925458	170.088687	5.537E-08
	Std	17.20874943	43.80576257	7.57286377	16.42365038	32.07835447	48.92439488	3.0327E-07
F10	Worst	0.000389207	19.96324461	1.15377E-08	3.461971697	19.96553835	11.46895332	4.4409E-15
	Best	9.01946E-07	0.002328418	6.03493E-09	1.86923E-05	19.95470594	5.366289847	4.4409E-15
	Mean	3.65168E-05	15.55001762	7.90426E-09	2.113011551	19.9629133	8.367928515	4.4409E-15
	Std	7.43114E-05	7.816387285	1.40856E-09	0.819959247	0.002010672	1.444896874	0
F11	Worst	0.083436053	0.073110064	14.2521124	0.041631323	697.0742975	31.70504566	0.0734106
	Best	1.86264E-10	4.5276E-06	1.924893133	3.103E-08	405.9656817	3.701576771	0
	Mean	0.01408819	0.007724682	8.130391319	0.012549719	594.0551975	12.73547402	0.00425366
	Std	0.018519105	0.014333012	2.632330714	0.010934052	62.70358321	6.301316803	0.01372796
F12	Worst	0.832470914	3.976990307	1.383151634	9.382469179	215118011.8	94967.48083	4.1692E-05
	Best	1.16878E-12	6.12699E-06	3.61774E-19	1.495014783	13349502.56	8.060083138	6.1926E-06
	Mean	0.076128157	0.596403093	0.210866057	5.259096023	114236067.3	3215.889468	1.6521E-05
	Std	0.168159827	0.905518879	0.32996828	2.437326838	43278555.83	17329.36143	7.7524E-06
F13	Worst	0.010987441	3.597522218	2.541412395	34.27164571	459412888.5	224257.9344	0.0549268
	Best	4.21936E-11	3.22767E-05	4.70688E-18	7.40714E-10	102349871.6	10.35665932	0.00011079
	Mean	0.004761209	0.371951501	0.144155746	2.02076033	279667889.2	31228.10471	0.00377484
	Std	0.005537702	0.828956267	0.483066779	7.421597353	85097409.42	54179.47924	0.01036565

Table 5Comparison results of SGA for the first 13 benchmark functions with $Dim = 100$.

<i>Fi</i>	<i>Measure</i>	PSO	GA	GSA	SSA	BAT	DA	SGA
F1	Worst	196.15346	6548.38838	1995.523	10.732861	262770.41	37638.983	1.7477E-33
	Best	14.296132	1123.12176	80.73054	0.4093725	144495.25	8266.9598	4.7791E-40
	Mean	58.682275	3580.20508	674.4992	3.6914469	238523.68	19361.966	1.8549E-34
	Std	35.659231	1293.04245	413.6181	2.7687465	22437.372	7917.9693	3.98E-34
F2	Worst	311.1893	2084.41489	951.2594	1.716E+85	2.5E+135	2492.241	2.6353E-27
	Best	1.1823855	583.533514	594.0675	2.642E+59	2.16E+118	414.00468	1.0439E-31
	Mean	98.258333	1431.13232	779.1757	5.722E+83	8.64E+133	1226.435	2.224E-28
	Std	124.28525	340.306531	71.22893	3.133E+84	4.55E+134	485.34317	5.3787E-28
F3	Worst	81550.499	193793.525	12382.87	78222.179	1717611.9	359424.6	54457.4657
	Best	34902.482	79760.4789	5867.29	19777.856	377729.42	92518.943	13288.1644
	Mean	52764.737	139987.768	8426.434	39461.615	950211.15	209690.76	32503.5332
	Std	12785.367	27543.2211	1622.805	14965.376	281650.13	78257.736	10916.4303
F4	Worst	36.834047	92.6566987	18.60109	34.47907	92.32292	60.374239	4.32592378
	Best	27.207148	83.4718335	13.49113	21.50373	83.89986	33.352024	1.12192636
	Mean	32.61691	88.335042	15.63213	26.62245	89.384561	49.393005	2.73022143
	Std	2.6626092	2.68517878	1.413522	2.8300778	1.8964209	7.9047532	0.77049078
F5	Worst	73928618	3.1332E+10	1.13E+09	2275863	2.412E+12	5.752E+10	98.6385085
	Best	3793460.5	2264772998	1.64E+08	22048.394	1.619E+12	8.345E+09	97.7549187
	Mean	18800492	1.0115E+10	5.51E+08	500315.76	2.042E+12	2.602E+10	98.3822335
	Std	16300929	6940064839	2.46E+08	522312.76	1.938E+11	1.228E+10	0.26225925
F6	Worst	278.98067	15049.7454	1267.083	14.91756	270874.31	34706.765	3.43177762
	Best	9.4899435	1479.85199	112.243	0.4364806	215033.55	3655.5672	1.43951874
	Mean	68.899183	4558.937	638.5467	2.8248372	246695.81	17560.379	2.56079854
	Std	50.477914	3994.84451	259.3443	2.6314861	13413.153	7891.0209	0.47135395
F7	Worst	1.1862871	2.7272918	5.502359	2.2009416	2.492078	34.511327	0.04707501
	Best	0.4471463	1.05623265	0.971437	0.6830555	1.1589751	0.3655382	0.00897938
	Mean	0.7081363	1.74034751	2.134801	1.323016	1.7568395	11.646658	0.02353628
	Std	0.1559912	0.42212079	0.918652	0.3584801	0.3837114	8.4607449	0.01058782
F8	Worst	-19974.26	-24584.672	-3656.36	-20273.97	-4386.193	-8151.281	-20576.548
	Best	-26066.64	-29117.959	-9031.01	-27732.66	-7266.309	-13046.74	-25995.449
	Mean	-23202.01	-27061.5	-5067.9	-23327.78	-5980.703	-10397.19	-23615.926
	Std	1692.6659	1341.3103	1077.542	1559.4717	705.94169	1293.6343	1080.774
F9	Worst	330.62325	963.776631	158.0901	235.52982	828.7954	1061.25	89.2268893
	Best	177.41906	607.424515	97.17747	108.92624	544.24011	311.04805	0
	Mean	238.65407	760.557497	136.475	170.84342	733.5127	775.32344	29.8932018
	Std	42.469697	97.8129577	16.69247	30.961915	65.175043	151.17897	28.2678011
F10	Worst	5.0395732	19.9401593	3.724461	8.4898461	19.966769	16.222032	2.2204E-14
	Best	2.7317918	19.108502	1.878893	3.8927864	19.96413	6.4751162	7.9936E-15
	Mean	3.483177	19.7288241	2.95277	6.6758218	19.966218	12.777219	1.4033E-14
	Std	0.5758238	0.23973596	0.513692	1.1548215	0.0006043	2.4920718	3.1154E-15
F11	Worst	92.372362	310.316883	125.4191	1.0569414	2694.1061	331.05841	0.08516804
	Best	1.2210922	25.4666145	64.0346	0.3066841	2041.6457	14.760138	0
	Mean	7.6361085	81.4854716	93.64504	0.6632802	2403.3467	161.69956	0.00859804
	Std	22.894483	61.0117738	14.74065	0.205708	131.26312	83.211822	0.02006191
F12	Worst	58.577967	1276777.11	8.740762	32.782533	1.646E+09	20619806	0.53941412
	Best	9.7251228	29204.2783	2.900192	8.0884336	1.053E+09	157.29247	0.05350412
	Mean	23.67217	275697.859	5.389221	17.59038	1.396E+09	4025679.5	0.19745661
	Std	11.411738	279409.88	1.555457	5.0939181	156318363	5291682.6	0.10457143
F13	Worst	5696.2818	5354914.88	176.8011	220.58934	3.456E+09	80509443	4.65750129
	Best	153.39719	344089.35	87.27473	146.99018	1.659E+09	1761044.9	2.44442067
	Mean	1419.3214	1812956.51	133.3381	183.25528	2.714E+09	21561926	3.67846409
	Std	1404.5502	1371842.19	19.16212	17.579194	368541588	22134512	0.54374363

Table 6Comparison results of SGA for the first 13 benchmark functions with $Dim = 1000$.

<i>Fi</i>	<i>Measure</i>	PSO	GA	GSA	SSA	BAT	SGA
F1	Worst	487742.301	2285644.7	107930.9665	208992.8582	3141992.8	0.000114354
	Best	284816.795	2061131.35	90561.44308	169379.7893	1554782.55	2.74728E-13
	Mean	352323.793	2154655.53	98162.925	192421.2527	2990655.66	5.22262E-06
	Std	44787.1249	60960.0536	4503.89146	8179.892165	292047.289	2.09754E-05
F3	Worst	9413915.8	19731127.5	4115238.594	8921483.071	204780615	9389125.454
	Best	4582872.16	9507211.12	961535.3889	1533203.332	36219224.4	5389977.394
	Mean	6634733.23	14324153.5	1665173.139	4733716.482	88412291	6972130.24
	Std	1228027.24	2460662.23	673937.876	1891890.44	40984974.7	1065356.175
F4	Worst	99.7217018	98.6491267	32.76535013	50.4002489	98.4532828	44.75914213
	Best	82.1566551	97.121565	26.56213849	38.66207	91.3219496	34.3633522
	Mean	95.7564028	97.9866172	29.02618175	44.30421323	97.4309357	38.38503279
	Std	6.17341641	0.39385265	1.513147064	3.01966241	1.5294457	2.235084541
F5	Worst	2.0459E+12	1.9088E+13	80057363117	1.74489E+11	2.9037E+13	9929482262
	Best	5.8501E+11	1.3784E+13	57009054280	1.22902E+11	2.2373E+13	5596230434
	Mean	1.0061E+12	1.6951E+13	66237783007	1.46532E+11	2.7723E+13	7579973851
	Std	3.4931E+11	1.2044E+12	6274388047	11655222936	1.2096E+12	976442537.9
F6	Worst	480302.142	2265917.92	109639.5684	219344.532	3089623.41	228.2066601
	Best	293483.823	2028189.38	89083.85752	173089.1671	1581385.73	217.9957106
	Mean	353399.849	2134251.46	98350.41168	191392.1904	2946338.41	222.6109698
	Std	48566.8189	64154.6174	4961.333586	11219.73693	336531.025	2.727076767
F7	Worst	14216.116	101605.863	6262.984712	1368.227682	28313.3404	71.58560739
	Best	5263.40026	86121.6852	4143.805376	891.9243055	17360.6315	38.61634641
	Mean	7368.71757	94082.7639	5264.895285	1107.490009	21939.0966	53.7971426
	Std	1792.73068	4593.12609	578.9378295	108.6136085	2446.94091	8.007160051
F8	Worst	-116915.93	-98193.6018	-10948.92927	-101033.4672	-15215.409	-95546.02213
	Best	-140159.703	-141757.409	-22511.77975	-130311.5693	-25558.45	-114660.4602
	Mean	-126685.747	-128147.234	-14949.15473	-117436.0174	-20220.394	-105067.2165
	Std	6487.35429	9506.80948	3019.173824	8120.511723	2518.46829	4529.657131
F9	Worst	8928.75785	13584.11	6210.298962	6895.028783	11218.09	725.3584075
	Best	7066.57076	12824.222	5370.969323	6046.840916	8727.23076	3.63798E-12
	Mean	7738.71593	13341.3002	5795.462049	6385.976021	10625.2427	125.2936317
	Std	382.750053	177.815497	209.5051558	209.7935408	466.759349	164.4986334
F10	Worst	18.0412743	20.5567903	10.68024149	14.79542341	19.9667687	2.19815E-06
	Best	15.8999685	20.4581444	9.977635556	13.85841269	19.9666675	3.9699E-09
	Mean	16.9921293	20.5234081	10.29322419	14.35003512	19.9667474	2.57334E-07
	Std	0.50036681	0.02413502	0.167299993	0.19420173	3.7598E-05	5.16499E-07
F11	Worst	3810.35484	22204.0674	14303.87331	1951.02318	28863.8483	0.167408008
	Best	2607.50151	19311.1381	13615.41243	1555.152413	14911.3373	2.2975E-12
	Mean	3162.05781	20341.4356	13932.8344	1733.516911	27687.1814	0.01109057
	Std	314.505786	604.365498	179.1025737	98.2274702	2451.39394	0.035487611
F12	Worst	1183458134	1.9782E+10	206342.3984	7920314.01	3.2586E+10	11122.77137
	Best	254032735	1.3789E+10	3781.003641	923557.9712	8456356496	467.8342268
	Mean	429466377	1.6264E+10	40053.19531	2978871.845	2.95E+10	4697.909035
	Std	205219304	1310799765	41016.39351	1372463.432	4076040365	2939.075443
F13	Worst	255999898	3.6996E+10	10782174.72	108866829.5	5.984E+10	2705667.663
	Best	1018563266	2.8396E+10	4178003.017	48110485.93	1.714E+10	692237.2365
	Mean	1485281103	3.2231E+10	6774940.487	73590535.52	5.3901E+10	1188998.748
	Std	380938574	2285595107	1533559.252	15859462.35	1.0114E+10	410733.8918

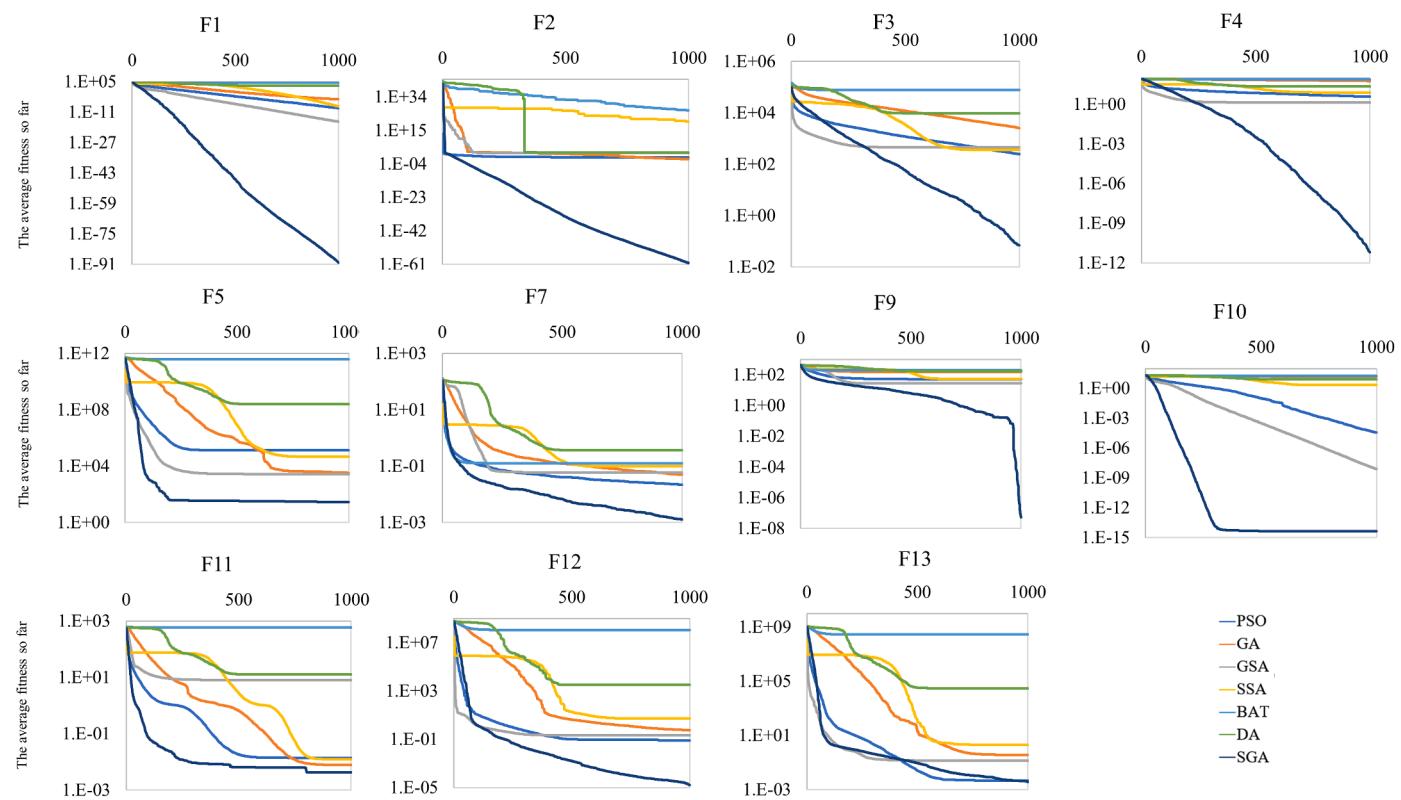
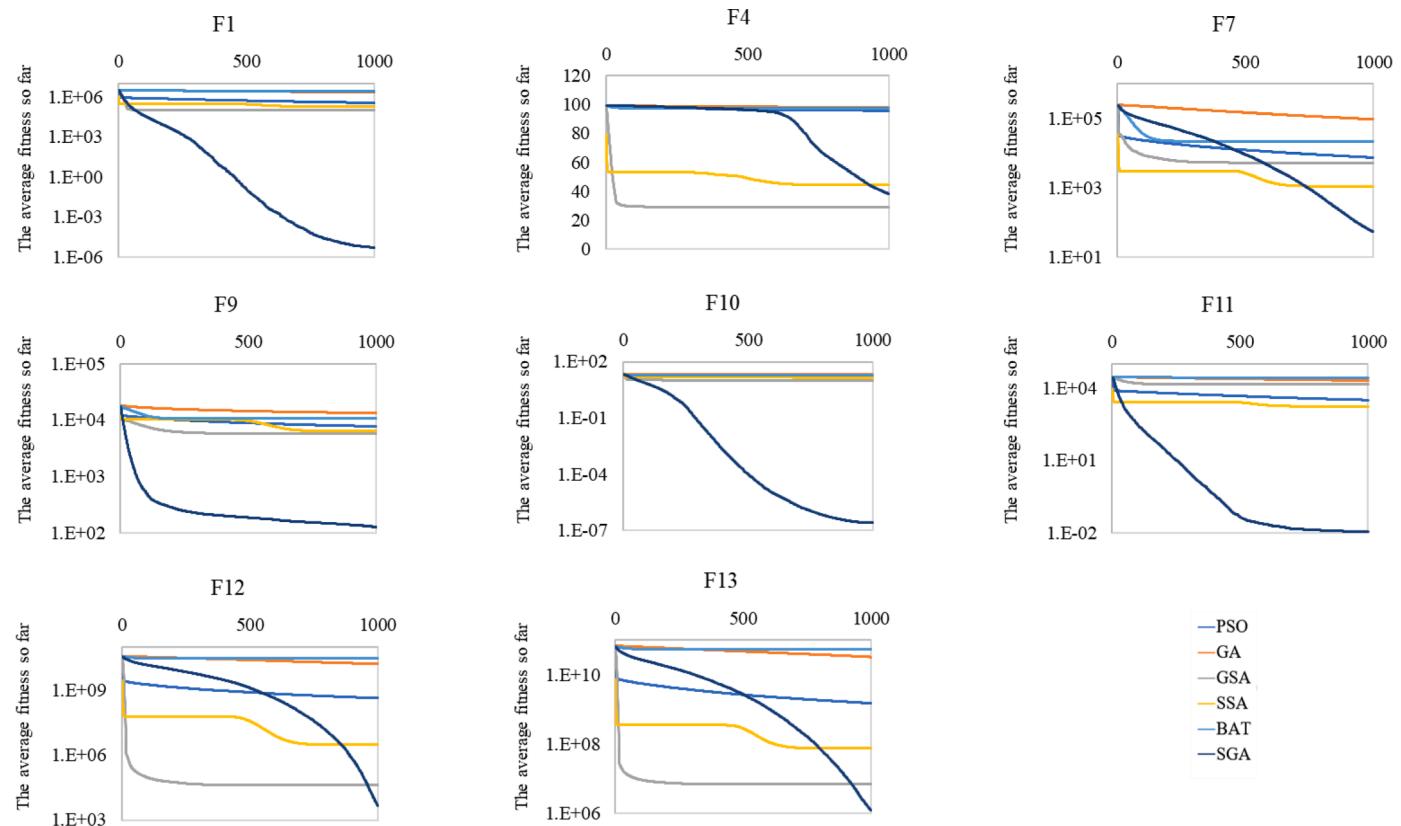
Fig. 2. Convergence curves of the average fitness with $\text{Dim} = 30$.Fig. 3. Convergence curve of the average fitness with $\text{Dim} = 1000$.

Table 7

Results of SGA on the high-dimensional benchmark functions.

<i>Fi</i>	<i>Measure</i>	<i>Dim 10</i>	<i>Dim 30</i>	<i>Dim 50</i>	<i>Dim 100</i>	<i>Dim 200</i>	<i>Dim 500</i>	<i>Dim 1000</i>
F1	Worst	8.9816E-89	1.79E-89	2.71E-57	1.75E-33	1.35E-18	8.7E-10	0.000114
	Best	1.528E-101	1.3E-100	6.26E-67	4.78E-40	1.58E-27	6.8E-16	2.75E-13
	Mean	4.9205E-90	9.36E-91	9.52E-59	1.85E-34	4.55E-20	7.6E-11	5.22E-06
	Std	1.8356E-89	3.43E-90	4.94E-58	3.98E-34	2.46E-19	2E-10	2.1E-05
F2	Worst	8.3715E-59	1.36E-59	3.53E-43	2.64E-27	3.53E-20	#NA#	#NA#
	Best	5.1041E-67	3.17E-64	2.41E-49	1.04E-31	1.31E-23	#NA#	#NA#
	Mean	2.8302E-60	6.22E-61	1.81E-44	2.22E-28	3.95E-21	#NA#	#NA#
	Std	1.5277E-59	2.5E-60	6.57E-44	5.38E-28	8.63E-21	#NA#	#NA#
F3	Worst	0.35403591	1.630246	1222.582	54457.47	350396	2541913	9389125
	Best	3.9014E-10	1.07E-07	19.34275	13288.16	166419.7	1297902	5389977
	Mean	0.02292397	0.069696	458.9612	32503.53	248673.7	1766869	6972130
	Std	0.06742339	0.2973	343.3555	10916.43	45033.03	250300	1065356
F4	Worst	2.0251E-09	7.32E-11	0.038508	4.325924	16.36172	32.0265	44.75914
	Best	4.6995E-21	3.23E-18	0.000467	1.121926	9.660197	24.084	34.36335
	Mean	8.9385E-11	6.5E-12	0.006458	2.730221	12.22259	28.0339	38.38503
	Std	3.7056E-10	1.8E-11	0.007792	0.770491	1.669327	1.94715	2.235085
F5	Worst	547.435434	107.5679	48.43585	98.63851	10252.57	2.6E+08	9.93E+09
	Best	25.1139478	25.0739	45.78478	97.75492	199.2856	9.1E+07	5.6E+09
	Mean	46.9026376	28.77254	46.84292	98.38223	1216.451	1.6E+08	7.58E+09
	Std	96.5448392	14.89619	0.745231	0.262259	1968.199	4.7E+07	9.76E+08
F6	Worst	0.00081816	0.000741	0.336607	3.431778	22.82286	100	228.2067
	Best	0.00015133	0.000106	0.004253	1.439519	17.81788	91.4359	217.9957
	Mean	0.00040534	0.000418	0.03162	2.560799	20.54541	95.8699	222.611
	Std	0.00015357	0.000163	0.062617	0.471354	1.405948	2.78121	2.727077
F7	Worst	0.00824274	0.002809	0.013821	0.047075	0.314735	4.41612	71.58561
	Best	0.00018391	0.000312	0.000992	0.008979	0.117262	2.34306	38.61635
	Mean	0.00164117	0.001268	0.004289	0.023536	0.193996	3.35833	53.79714
	Std	0.00149543	0.000671	0.002755	0.010588	0.054216	0.56772	8.00716
F8	Worst	-8875.501	-8723.99	-12836.4	-20576.5	-33894.4	-64590.4	-95546
	Best	-10727.901	-10440.7	-15726.7	-25995.4	-42121.8	-75723.2	-114660
	Mean	-9578.0125	-9645.8	-14255.8	-23615.9	-38557.5	-70280.1	-105067
	Std	405.470671	455.9688	598.0573	1080.774	1856.026	3017.8	4529.657
F9	Worst	1.1092E-09	1.66E-06	23.37258	89.22689	201.8739	395.964	725.3584
	Best	0	0	0	0	2.781305	1.2E-11	3.64E-12
	Mean	3.6973E-11	5.54E-08	1.155846	29.8932	62.98336	126.253	125.2936
	Std	2.0251E-10	3.03E-07	4.317983	28.2678	59.51329	118.909	164.4986
F10	Worst	7.9936E-15	4.44E-15	7.99E-15	2.22E-14	5.2E-11	2.7E-07	2.2E-06
	Best	8.8818E-16	4.44E-15	4.44E-15	7.99E-15	5.06E-14	4.1E-11	3.97E-09
	Mean	4.5593E-15	4.44E-15	5.51E-15	1.4E-14	2.82E-12	1.5E-08	2.57E-07
	Std	1.1363E-15	0	1.66E-15	3.12E-15	9.83E-12	4.8E-08	5.16E-07
F11	Worst	0.10052874	0.073411	0.036858	0.085168	0.091946	0.08624	0.167408
	Best	0	0	0	0	0	1.1E-16	2.3E-12
	Mean	0.00998401	0.004254	0.004262	0.008598	0.008418	0.00942	0.011091
	Std	0.0232597	0.013728	0.010467	0.020062	0.02205	0.02369	0.035488
F12	Worst	3.8041E-05	4.17E-05	0.012771	0.539414	2.701036	15.3969	11122.77
	Best	5.4964E-06	6.19E-06	0.000129	0.053504	0.788402	7.20764	467.8342
	Mean	1.7166E-05	1.65E-05	0.001994	0.197457	1.740839	10.8963	4697.909
	Std	7.8684E-06	7.75E-06	0.002904	0.104571	0.514701	2.33047	2939.075
F13	Worst	0.10896904	0.054927	0.310624	4.657501	28.78748	4056.76	2705668
	Best	0.00013614	0.000111	0.009237	2.444421	17.69637	504.507	692237.2
	Mean	0.00656823	0.003775	0.127479	3.678464	23.00775	1031.38	1188999
	Std	0.01989548	0.010366	0.082593	0.543744	2.888628	759.504	410733.9

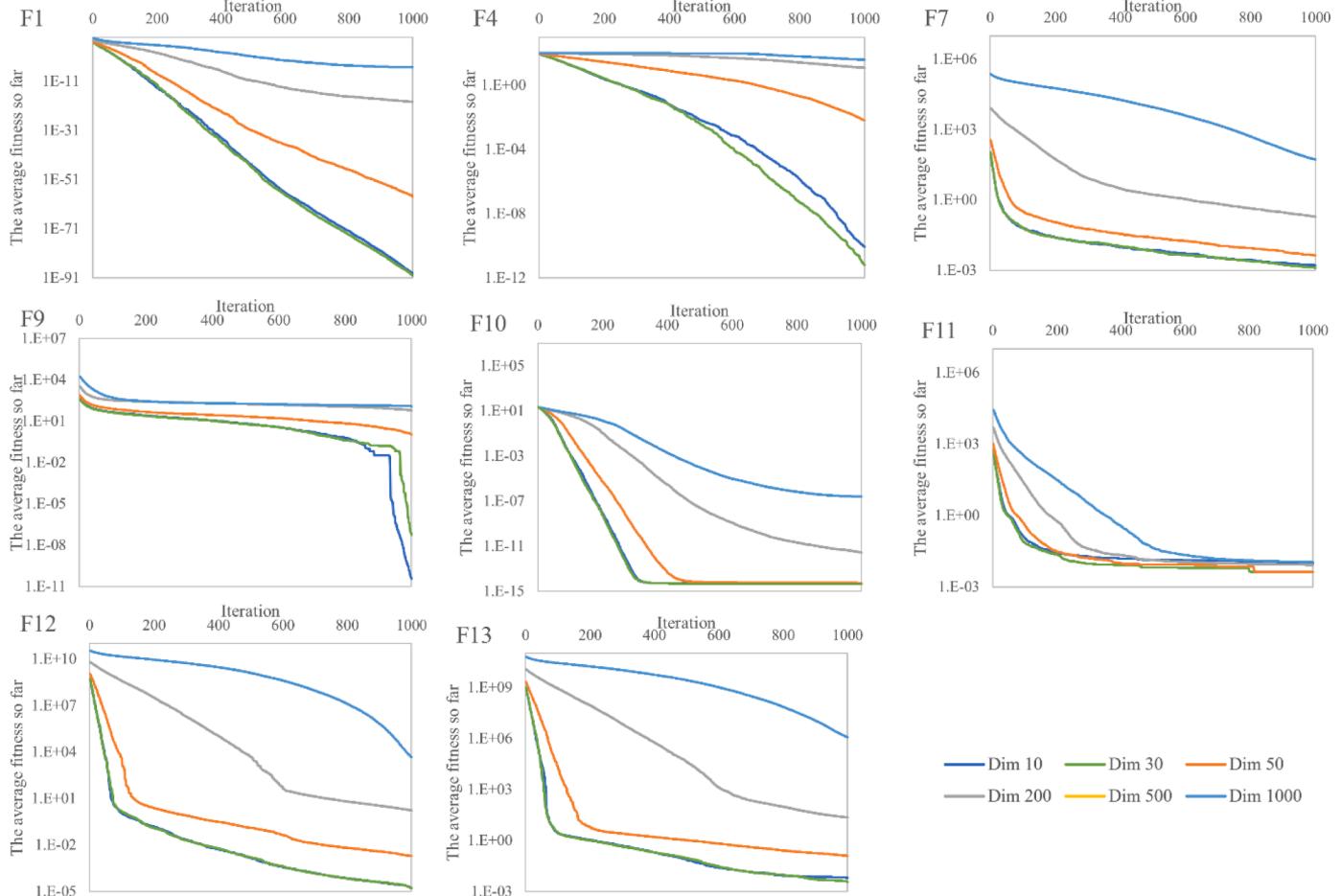


Fig. 4. Convergence curve (SGA) of the average fitness with the difference of Dim.

presented. Shrimp can receive two kinds of information. One of them is related to the components of goby's position vector (G_m) as shown in Eq. (5), or the information of the around environment as shown in Eq. (6).

$$B_{i,d}^{t+1} = r_2 \times G_{m,d}^t \quad (5)$$

$$B_{i,d}^{t+1} = lb_d + r_2 \times (ub_d - lb_d) \quad (6)$$

where $B_{m,d}^t$ is the d^{th} information of gobiid fish, $B_{i,d}$ indicates the d^{th} component of shrimp, and lb and ub are lower and upper boundaries, respectively.

The selection of a goby (G_m) for signal transmission must satisfy two factors. One is to ensure information accuracy, and the other is to ensure diversity to avoid local optimization. Therefore, the selection (m) using the roulette wheel selection method is considered, or proportionate fitness selection (FPS) [8,14,15]. This allows for the selection of well-positioned individuals, at the same time, and it does not ignore other individuals that make it possible for finding out potential regions.

3.2.2. The shelter and inside activity (local search)

In the real world, it is interesting to note that shrimps generally extend their shelters due to living at least in pairs [39]. At the same time, all of them generally move between adjacent burrows as well as a part of their reproductive behaviour. Paired openings to the burrows of shrimps were found by the study of Lipowski and Lipowska [40].

This means an association, including at least two shrimps and goby, is created in the small space, and all of them share the burrow system to

shelter. This behaviour is employed to simulate the local search process using a small number of candidates (Ξ) to search around two of the best adjacent burrows.

In fact, such shelter system always has a group of two, three individuals, or even more. To simulate local search, SGA employs this mechanism to activate Ξ individuals for the searching local in the two best burrows.

Clearly, individuals do not have a safe burrow, and it is believed that they are more vulnerable to predators. It means that if we choose a less potential search space to explore, of course, the search solution will be less effective. For this reason, we should find solution for how to control the walk in such a way that it can move toward the optimal solutions more quickly. Therefore, we set up the worst candidates into the two best shelter for increased survival. At the same time, the local search process of SGA is also reinforced effectively. In order to represent the model, the following formulas are proposed in this regard:

$$B_i^{t+1} = S_1^t \otimes (1 + \varsigma \times r_3) \quad (7)$$

$$B_i^{t+1} = S_2^t \otimes (1 + \varsigma \times r_3) \quad (8)$$

where S_{b1} and S_{b2} are the best positions of adjacent burrows, $\varsigma \in [2, 0]$ is a decreasing linear function, and r_3 is a random vector in the range (-1,1).

The pseudo code of the SGA search algorithm is described in Fig. 1.

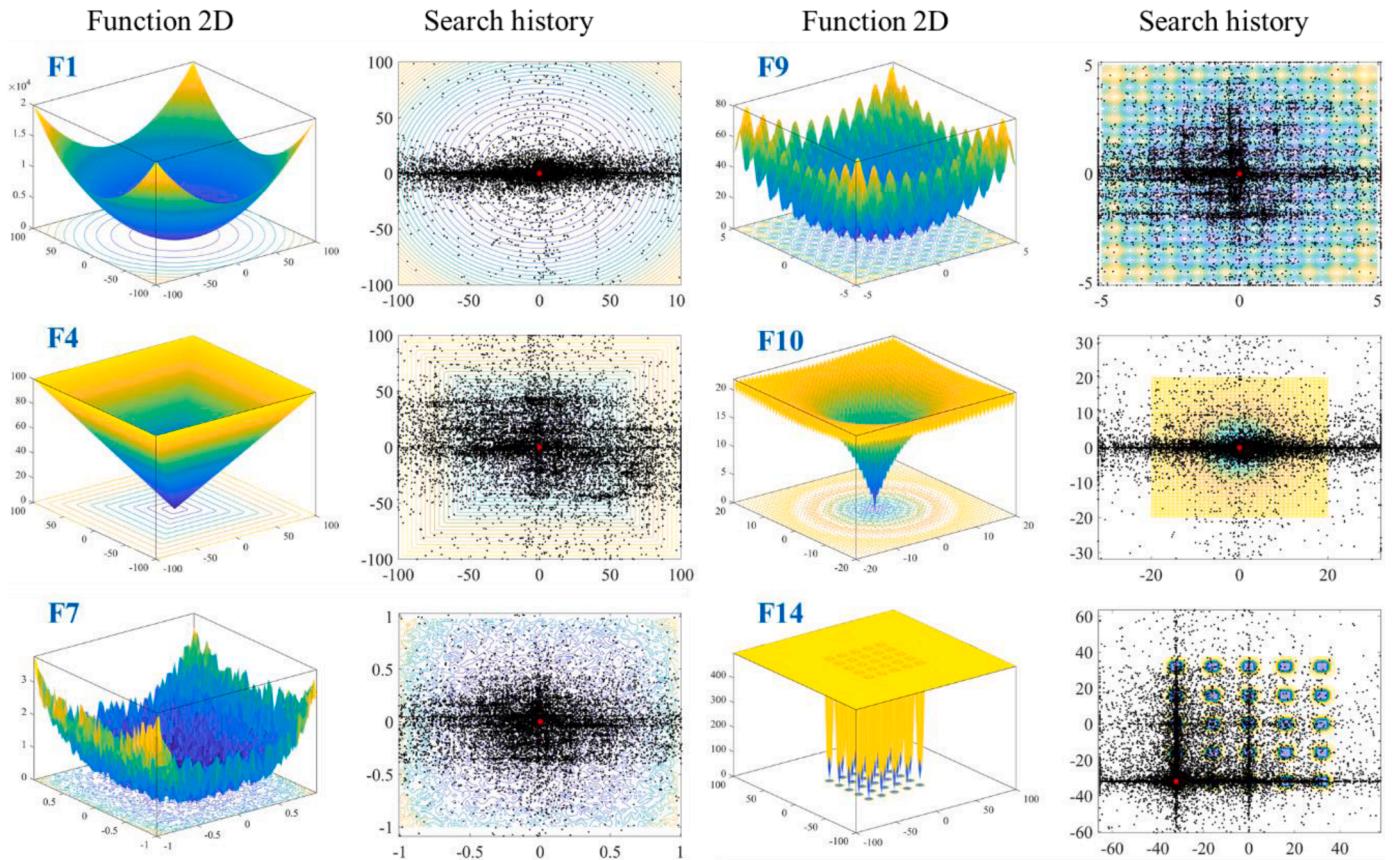


Fig. 5. Search history of SGA.

4. SGA for high-dimensional optimization problems

For single-objective optimization using metaheuristic algorithms, a series of investigations should be adopted to determine the performance of an algorithm. In order to confidently make sure that the superior results of any algorithm, which is the commonly stochastic in nature, does not occur by chance, we should employ a powerful and sufficient set of experiments and case studies.

It is a fact that there is, however, so far, no correct definition for a proper set of benchmark cases of research. For this reason, all studies try to assess their algorithms using as many benchmark functions as possible. In this paper, metaheuristic algorithms are proposed to deal with a wide range of benchmark problems [14,39,41] from simple to complex functions. The second test is a series engineering problem on real-parameter optimization

Moreover, the real challenge in damage detection is also presented to assess this algorithm in the next section. This will be a strong proof for the efficiency of proposed algorithms.

4.1. Classical benchmark functions

In this subsection, SGA is investigated on a set of 13 benchmark functions, including unimodal and multimodal classical benchmark functions (see Tables 1 and 2), with $Dim = 10$ to 1000 . The main aim is to assess the performance of SGA using the challenge of high-dimensional problems. Then, to test operation of this algorithm, ten fixed-dimension multimodal benchmark functions (see Table 3) are also used as additional tests to study the search history of SGA in different search spaces.

A series of modern algorithms, such as PSO, GA, GSA, SSA, BAT and

DA, are selected to compare with SGA. Each candidate is independent with 30 runs, 30 agents and 1000 iterations. The obtained results are shown in the Table 4.

To investigate thoroughly the movement trajectories of individuals in SGA, we conduct some more examples on 10 of the Fixed-dimension multimodal benchmark functions. The results are shown in Fig. 5 and Table 8.

The best way to highlight the algorithm's capability during the exploration phase is to use unimodal test functions. Series of uni-modal benchmarks, including 7 first functions, are employed. f_{min} displays the best cost and other pertinent best places of these functions listed in Table 1. The word "Dim" stands for the number of issue dimensions (number of variables).

The SGA results are quite close to the absolute ideal point. SGA's best cost (Best) and standard deviation (Std) results are much superior to those of any other algorithms. In many functions, even the worst (Worst) of SGA outperforms better than other algorithms. These results are shown in the Table 4 ($Dim = 30$), Table 5 ($Dim = 100$) and Table 6 ($Dim = 1000$).

In comparison with other algorithms in Figs. 2 and 3, the results are very different, and SGA's convergence curve demonstrates its very effective performance. In many cases, SGA's responses at 500th iteration are more accurate than most algorithms at the last iteration. It can be seen that the suggested algorithm performs exceptionally well in terms of speed and accuracy. Without a doubt, SGA offers the best solutions.

An efficient criterion for testing an algorithm's exploitation potential is multimodal benchmark functions. In this section, six separate functions, compromising the F8 to F13 functions, are employed. These functions put optimization algorithms at danger of becoming caught in

Table 8

Comparison results of SGA for the fixed-dimension multimodal benchmark functions.

<i>Fi</i>	<i>Measure</i>	PSO	GA	GSA	SSA	BAT	DA	SGA
F14	Worst	10.76318067	9.803897943	11.80606628	0.998003838	23.80943447	0.998003846	0.99800384
	Best	0.998003838	0.998003838	0.998003838	0.998003838	3.968250106	0.998003838	0.99800384
	Mean	3.850351953	3.069528684	3.891498542	0.998003838	14.12467676	0.998003838	0.99800384
	Std	3.149823968	2.410145189	3.016545154	2.59143E-16	6.822981448	1.49272E-09	2.6721E-14
F15	Worst	0.020363339	0.020363339	0.005144812	0.001247279	0.084250802	0.020363339	0.00122317
	Best	0.000307486	0.000378933	0.001035502	0.000493018	0.000565713	0.000636556	0.00030749
	Mean	0.003229314	0.002767988	0.00234697	0.000828769	0.009978421	0.002358843	0.00042512
	Std	0.006850629	0.005969657	0.001007878	0.000260886	0.01868594	0.003836244	0.00020311
F16	Worst	-1.031628453	-1.031628453	-1.031628453	-1.031628453	2.10425031	-1.031547498	-1.0316285
	Best	-1.031628453	-1.031628453	-1.031628453	-1.031628453	-1.031628453	-1.031628453	-1.0316285
	Mean	-1.031628453	-1.031628453	-1.031628453	-1.031628453	-0.740953406	-1.031623578	-1.0316285
	Std	6.77522E-16	6.77522E-16	4.87871E-16	2.58336E-14	0.812214637	1.54313E-05	9.7995E-13
F17	Worst	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397936652	0.39788736
	Best	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.39788736
	Mean	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397889061	0.39788736
	Std	0	0	0	2.36428E-15	0	8.99001E-06	6.8822E-11
F18	Worst	3	30	3	3	840	3.003383014	3
	Best	3	3	3	3	3	3	3
	Mean	3	3.9	3	3	77.7	3.000140162	3
	Std	1.55377E-15	4.929503018	2.86737E-15	3.57415E-13	209.3433329	0.000616731	1.4542E-14
F19	Worst	-3.862782148	-3.862782148	-3.862782148	-3.862782148	-3.862782148	-3.862163586	-3.8627821
	Best	-3.862782148	-3.862782148	-3.862782148	-3.862782148	-3.862782148	-3.862782147	-3.8627821
	Mean	-3.862782148	-3.862782148	-3.862782148	-3.862782148	-3.862782148	-3.862719939	-3.8627821
	Std	2.71009E-15	2.71009E-15	2.38722E-15	3.33053E-14	2.65431E-15	0.000119905	1.4403E-09
F20	Worst	-3.20310205	-3.20310205	-3.321995172	-3.193701121	-3.203099305	-3.148398974	-3.2031019
	Best	-3.321995172	-3.321995172	-3.321995172	-3.321995172	-3.321995172	-3.321995172	-3.3219952
	Mean	-3.290290339	-3.258585507	-3.321995172	-3.237032199	-3.246695927	-3.267421156	-3.2625485
	Std	0.053475325	0.060328303	1.50485E-15	0.056630206	0.058273592	0.067520329	0.0604628
F21	Worst	-2.630471668	-2.630471668	-2.630471668	-2.630471668	-2.630471668	-2.625423258	-2.630471
	Best	-10.15319968	-10.15319968	-10.15319968	-10.15319968	-10.15319968	-10.15319968	-10.153199
	Mean	-6.735633685	-6.321246418	-6.648410631	-8.141657205	-5.452308543	-7.523407742	-6.8673377
	Std	3.558824404	3.511440623	3.503111214	2.977099154	2.846594688	2.681256524	2.82087886
F22	Worst	-2.751933564	-1.837592971	-6.378051225	-5.128822797	-1.837592971	-1.83682672	-1.8375929
	Best	-10.40294057	-10.40294057	-10.40294057	-10.40294057	-10.40294057	-10.40294057	-10.40294
	Mean	-7.266297812	-6.647453935	-10.26877759	-10.22713664	-4.354404702	-8.262446264	-6.3540556
	Std	3.67773811	3.834912796	0.734840895	0.962917758	2.856553676	2.914749869	3.47592303
F23	Worst	-2.421734027	-2.421734027	-10.53640982	-3.835426803	-1.67655325	-2.420358443	-2.4217332
	Best	-10.53640982	-10.53640982	-10.53640982	-10.53640982	-10.53640982	-10.53640982	-10.536409
	Mean	-8.511870031	-7.794648638	-10.53640982	-9.596703108	-5.464103019	-7.633137889	-7.1911485
	Std	3.426899089	3.684379903	2.18805E-15	2.148542016	3.704281966	3.451825291	3.50295678

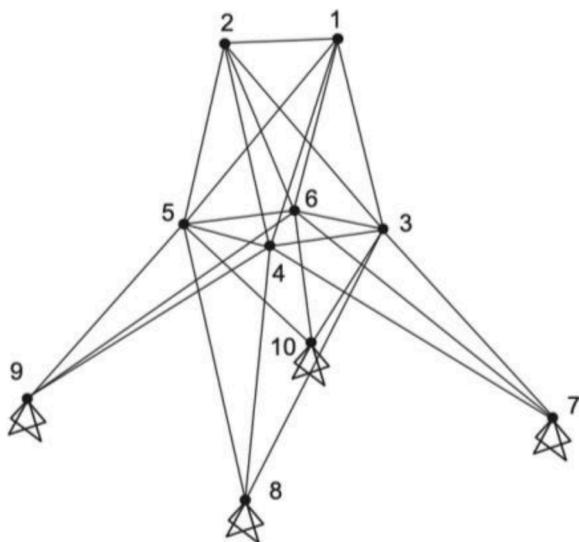


Fig. 6. A 25-bar spatial truss structure.

one of the numerous local minima. The difficulty of these situations increases dramatically as the problem dimensions rise. As mentioned, SGA, together with PSO, GA, GSA, SSA, BAT and DA algorithms, is used to optimize these functions at 30, 100, 500 and 1000 dimensions, respectively.

These are complicated functions that provide good illustrations of a non-linear multimodal function. Because of the huge search space and the high number of local minima, determining the minimum value of these functions is a tough job. In this situation, SGA performs impressively, with the best cost and Std of the stated findings is the smallest. The other algorithms produce no acceptable response (e.g. $Dim = 1000$ as shown Fig. 3 and Table 6). The global best cost obtained by SGA at all dimensions is calculated to be equal to zero in each analysis (except for F13 function). It is worth noting that the result achieved by HOA is significantly superior to those obtained by other contenders, even whether the result reaches the desired value of F13 or not.

In more detail, Table 7 summarizes the Results of SGA on the high-dimensional benchmark functions from $Dim = 10$ to 1000. We see that when Dim changes from 10 to 30 the results of the SGA are almost unaffected, which means that the results obtained from SGA are still almost correct even though the difficulty of the problem increases many times. When Dim has increased from 50 to 1000, there is a big variation in the result obtained in most functions (see Fig. 4). However, it should be emphasized that even when $Dim = 1000$, and when the problems become extremely complex, SGA is still more efficient than many other

Table 9

Results of the 25-bar truss problem.

Groups	Element's Name	Optimization algorithm					
		SGA	WOA	SSA	CS	GWO	AOA
G1	1 to 4	0.1	0.10194	0.10623	0.1	0.11098	0.1
G2	5 to 12	2.022	2.16221	1.90498	1.99736	2.01912	1.81687
G3	13 to 16	2.98121	2.9891	3.11243	3.02744	2.99717	3.4
G4	17, 18	0.1	0.10439	0.1	0.1	0.12093	0.1
G5	19 to 22	0.1	0.1516	0.1	0.10034	0.10312	0.1
G6	23 to 30	0.6905	0.92273	0.72656	0.66907	0.65215	0.77972
G7	31 to 34	1.65097	1.63212	1.69146	1.6519	1.64865	1.86934
G8	35, 36	2.65838	2.43204	2.5734	2.67309	2.70526	2.37071
Best Weigh (lb)		548.233	560.122	548.786	548.227	548.764	562.345
No. of Analysis		14280	14370	14880	25440	15000	14970

candidates. In practice, many problems often have a variable number that ranges from 10 to 50, so the accuracy of a stable SGA in this range is a huge advantage. Therefore, it can be applied to many problems and will provide good results and stability.

To investigate thoroughly the movement trajectories of individuals in SGA, we conduct some more examples on 10 of the Fixed-dimension multimodal benchmark functions. The results are shown in Fig. 5 and Table 8.

Fig. 5 shows a two-dimensional view and the position of the individuals in the search space. Meanwhile, Fig. 5 illustrates the SGA convergence behaviour of the optimal response for the F1 and F7 functions, which have just one global optimum. Because these functions have just one optimum solution, individuals tend to fully exploit the space around the global optimum. Although F4 is also a unimodal function, it displays the ideal solution convergence equilibrium between exploration and exploitation.

The F9 and F10 functions are commonly used to evaluate exploration and exploitation. Although there are a huge number of local optima, the SGA still converges strongly around the global optimal point.

For fixed-dimension multimodal benchmark functions, F14 is selected to investigate the operation of SGA, which is a function with complex local locations. This makes it easy for the algorithms to get stuck in local optimization positions. The proof is that the SGA spent a lot of resources looking around for a local value. Fortunately, SGA has finally determined the global optimal and the search history shows that it has explored the region quite well around the global optimal point.

To summarize, there are several meta-heuristic algorithms to solve various sorts of optimization issues, which profit from two fundamental properties, namely exploration and exploitation, to explore the optimization problem spaces and to find the good-enough replies. Exploration refers to an algorithm's capacity to search freely, regardless of the correctness of the findings. Exploitation also refers to an algorithm's performance in terms of its past iteration loop successes. Obviously, as search capabilities improve, an algorithm's behaviour becomes more random and unpredictable. On the contrary, increased exploitation ratio in an algorithm leads to more cautious performance. In fact, balancing these two factors in an algorithm is already an optimization problem that is not easy to implement. And each algorithm has its own advantages for different problems. However, the initial results obtained in the evaluation of 23 functions show that SGA has a good ability to solve some complex high-dimensional problems compared to other candidates.

To further evaluate the practical application of SGA in specific fields, in the next section, we use it to solve some real engineering problems.

4.2. Engineering problems

In this section, a series of truss structures having 25 bars and 72 bars

are used to evaluate the performance of the SGA. These problems are run 50 times (with $N=30$ and $T=500$), and then choose the best result.

4.2.1. A 25-bar truss

Fig. 6 depicts a 25-bar spatial truss structure that should be constructed for a load scenario in accordance to Camp and Bichon [42,43]. Material density and elasticity modulus are assumed to be 0.1 pound per cubic inch and 10,000 ksi, respectively. Every node's maximum displacement in each of the three directions is restricted to 0.35 in. As design variables, the truss's twenty-five members are grouped into eight groups of cross-sectional areas. The domain of cross-sectional areas alters from 0.01 to 3.4 in².

Table 9 and Fig. 7 present the optimization results, which show that SGA is superior to all algorithms (except CS). For all candidates, there have been 15000 structural investigations. However, after 14,280 times analysis, SGA arrive at its solution, instead of 14970 for AOA [44], even 15000 for the GWO.

On the other hand, it is noted that although CS provides better results than SGA, by only about 0.001%, it consumes more number of analyses, i.e. 1.78 times as high as SGA. Therefore, If we consider the same number of times of analyses in this case, then SGA is naturally superior to CS.

4.2.2. A 72-bar truss

The second selection test is a spatial 72-bar truss [42,43], as shown in Fig. 8. The material's elastic modulus is 10,000 ksi, and its density is 0.1 lb/in³. Members of cross-sectional regions are used as design variables and are divided into 16 groups. All members have the same permissible stress in tension and compression, which is equivalent to 25 ksi. Top node displacement must be less than 0.25 in both the x and y directions. The lowest authorized cross-sectional area of each member in this construction is 0.1 in², while the highest is 4 in².

Fig. 9 illustrates the convergence history of algorithms with SGA values smaller than all algorithms. Table 10 shows the solutions of some algorithms, as well as SGA, with the value of 379.917 lb by SGA achieving the lightest structure.

5. A real large-scale global optimization problems using SGA

A structural damage prediction problem is solved in this paper using the SGA algorithm. The main stages of the algorithm are summarized in Fig. 10.

In the beginning, a population is created, in which the position of each member is randomly selected. In the end of process, prey is caught. In other words, the aim of the work is to find the location and severity of damage.

The objective function (*obj*) is a combination of natural frequencies and mode shapes.

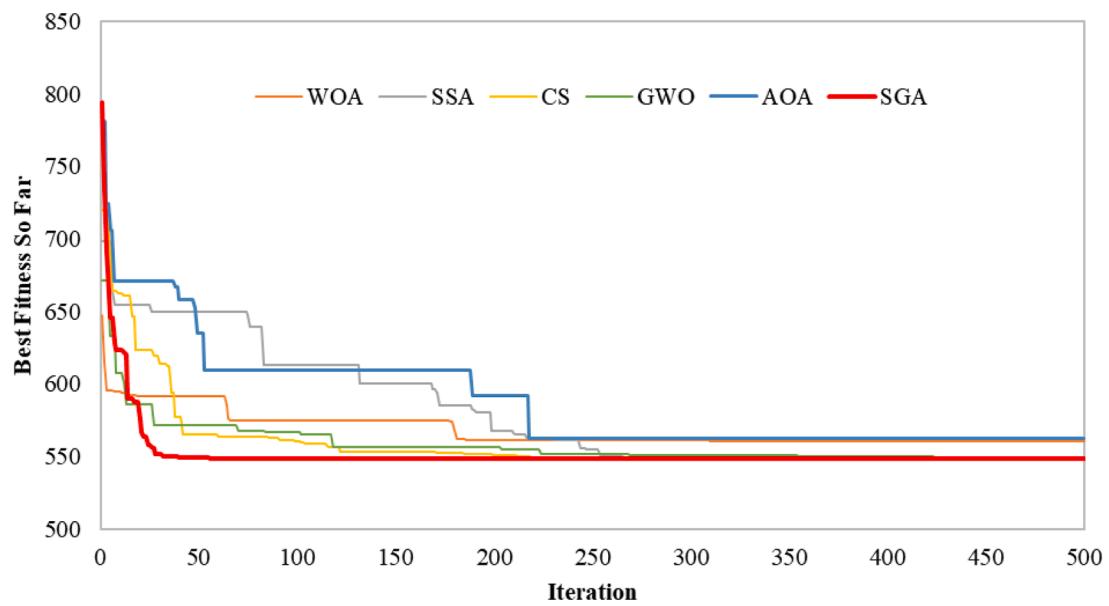


Fig. 7. Convergence curves the 25-bar truss problem.

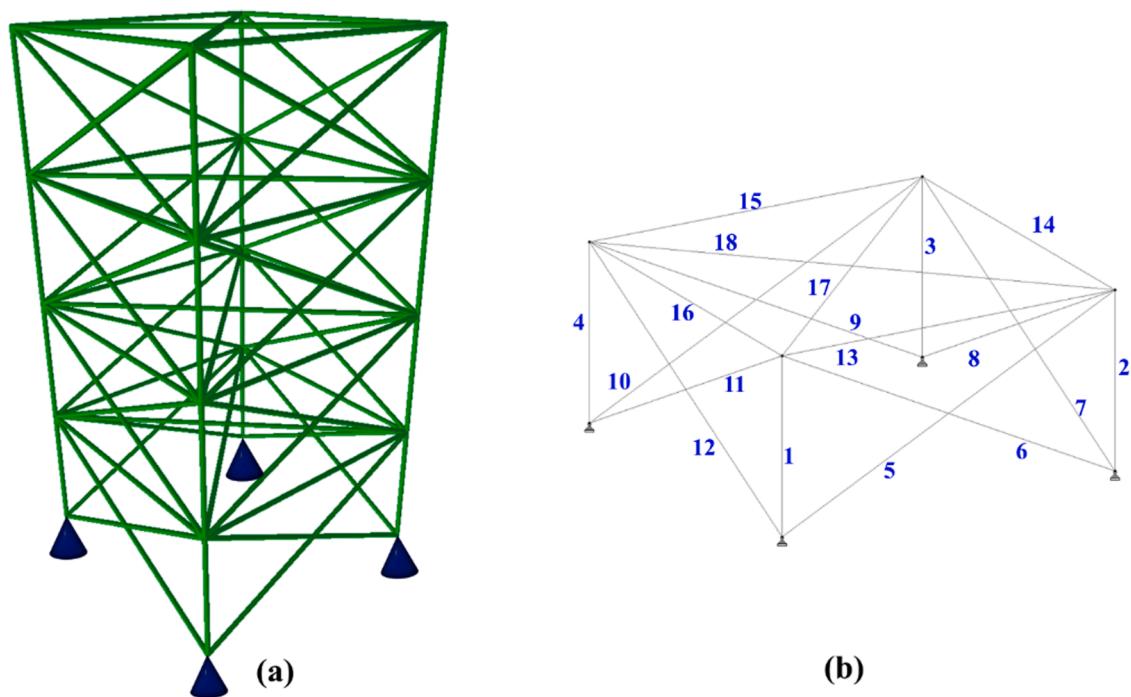


Fig. 8. A 72-bar truss problem: (a) 3D view this problem and (b) bar numbering pattern for first story.

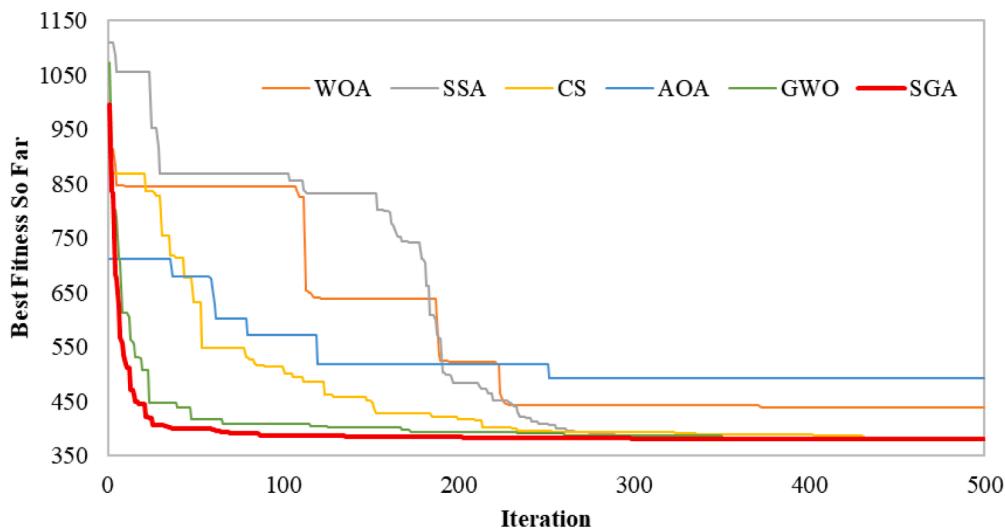


Fig. 9. Convergence curves for the 72-bar truss problem.

Table 10
Results of the 72-bar truss problem.

Groups	Element's Name	Optimization algorithm					
		SGA	WOA	SSA	CS	AOA	GWO
G1	1 to 4	1.87058	2.08302	1.94803	1.88248	3	1.844957
G2	5 to 12	0.51752	0.60808	0.48217	0.49107	0.68896	0.523981
G3	13 to 16	0.10003	0.1	0.10001	0.11372	0.1	0.100427
G4	17, 18	0.10095	1.02169	0.11506	0.1	0.1	0.103924
G5	19 to 22	1.30958	1.21612	1.50088	1.42947	2.02333	1.255251
G6	23 to 30	0.51186	0.63552	0.48855	0.50863	0.53606	0.519107
G7	31 to 34	0.1001	0.1	0.1	0.1	0.1	0.100411
G8	35, 36	0.10004	0.11813	0.1	0.1	0.1	0.100013
G9	37 to 40	0.50452	1.17955	0.62293	0.46821	0.49369	0.543616
G10	41 to 48	0.52903	0.45615	0.46332	0.54783	0.75294	0.520211
G11	49 to 52	0.10051	0.1	0.1001	0.1	0.42494	0.1
G12	53, 54	0.10192	0.24066	0.10314	0.10633	0.28603	0.100583
G13	55 to 58	0.15732	0.16553	0.15505	0.16152	0.1747	0.15667
G14	59 to 66	0.53101	0.534	0.53785	0.55081	0.65459	0.548238
G15	67 to 70	0.42875	0.38158	0.44606	0.43984	0.42494	0.383013
G16	71, 72	0.53492	0.2349	0.69189	0.47234	0.21432	0.559663
Best Weigh (lb)		379.917	438.756	383.191	382.416	493.321	380.0138
No. of Analysis		14820	14610	14820	26580	7590	14910

To find the frequencies and mode shapes, it is necessary to solve Eq. (9):

$$[M]\ddot{x} + [C]\dot{x} + [K]x = \vec{F}$$

Ignoring force and damping, we can write:

$$[M]\ddot{x} + [K]x = \vec{0}$$

From which

$$x = \psi \sin(\omega t + \varphi)$$

$$\ddot{x} = -\omega^2 \psi \sin(\omega t + \varphi)$$

From Eqs. (10)–(12), we can rewrite (9) as follows:

$$([K] - \omega^2[M])\vec{\psi} = \vec{0}$$

any member is damaged.

$$K = \begin{bmatrix} K_{11} & K_{12} & \dots & K_{1n} \\ K_{21} & \dots & K_{2n} \\ \dots & \dots & \dots \\ Sym & & & K_{nn} \end{bmatrix} \quad (16)$$

where K is the system stiffness matrix of structure and K_e is stiffness matrix of element. If K_e goes down $r = \frac{r_e^d}{r_e} \%$ in $[0, 1]$, K also declines to certain extent. K_e is defined as:

$$K_e^d = \left(1 - \frac{r_e^d}{r_e}\right) K_e \quad (17)$$

A technique is proposed to locate and find the severity of damage in a structure, which is based on Eq. (17). Using simulation, the damage in the structure can be easily detected. This is operated through a review of a reduction of stiffness in each member of the structure, both natural frequencies (f_i) and mode shapes (ψ_i) of each mode are evaluated by an objective function, where (f_i) and (ψ_i) are derived from the solution of Eq. (13).

The objective function is illustrated by Eq. (20) as follows:

Eq. (10) shows that the global stiffness matrix is going to decrease if

$$f_i = \frac{\omega_i}{2\pi} \quad (15)$$

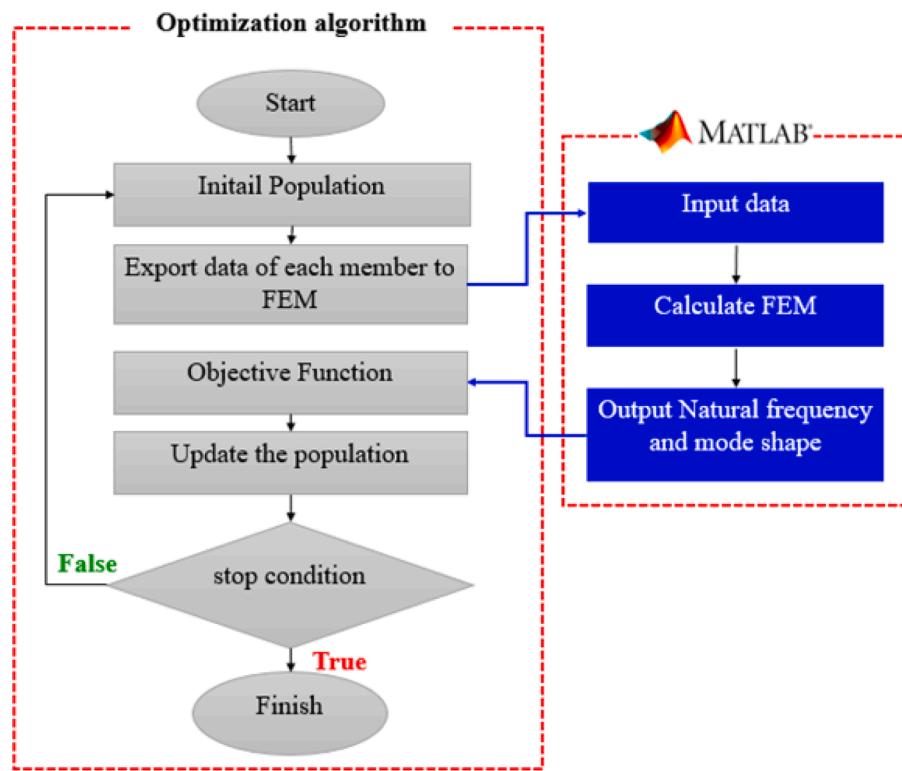


Fig. 10. Flowchart for the prediction of damage.

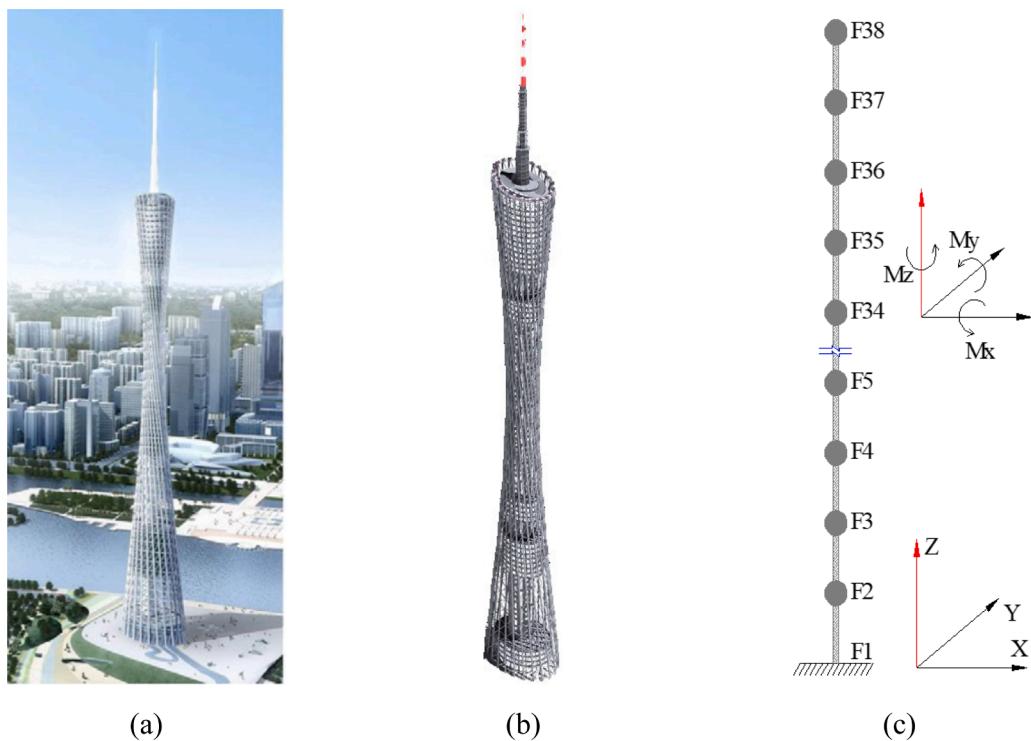


Fig. 11. Canton Tower [45].

Table 11

The position (Z) of floors in terms of mass assignment for reduce-order model.

Floor	Height (m)						
F1	-10.00	F11	147.05	F21	375.85	F31	505.20
F2	0.00	F12	157.45	F22	381.20	F32	520.70
F3	12.00	F13	168.00	F23	396.65	F33	531.20
F4	22.25	F14	204.25	F24	407.05	F34	545.20
F5	27.60	F15	225.20	F25	417.45	F35	565.20
F6	58.65	F16	272.00	F26	427.85	F36	580.70
F7	84.65	F17	308.25	F27	438.25	F37	598.00
F8	95.05	F18	329.20	F28	443.60	F38	610.00
F9	105.45	F19	344.64	F29	480.00		
F10	116.20	F20	355.05	F30	497.00		

Table 12

Natural frequencies (Hz) of the full-scale model and reduced order one without damage.

Mode	Full-scale model	reduced-order model	Difference (%)
1	0.1100	0.1104	0.364
2	0.1590	0.1587	0.189
3	0.3470	0.3463	0.202
4	0.3680	0.3688	0.217
5	0.4000	0.3994	0.150
6	0.4610	0.4605	0.108
7	0.4850	0.4850	0.000
8	0.7380	0.7381	0.014
9	0.9020	0.9026	0.067
10	0.9970	0.9972	0.020

$$F_i = \frac{(f_i^c - f_i^m)^2}{(f_i^m)^2} \quad (18)$$

$$MAC(c, m)_j = \frac{\left(\left| \sum_{i=1}^n \{\psi_c\}_j \{\psi_m\}_j \right| \right)^2}{\left(\sum_{i=1}^n \{\psi_c\}_j^2 \right) \left(\sum_{i=1}^n \{\psi_m\}_j^2 \right)} \quad (19)$$

$$Obj = w_1 \sqrt{\sum_i^n F_i} + w_2 \sum_j^m \left(1 - MAC(c, m)_j \right) \quad (20)$$

where f_i^c and f_i^m indicate the natural frequencies computed and measured accordingly; and $\{\psi_c\}_j, \{\psi_m\}_j$ are the mode shapes of computation and measurement, respectively. Meanwhile, w_1 and w_2 are weighting factors.

To apply SHM to real structure, we present an damage evaluation in Canton Tower shown in Fig. 11 using SGA. Some researchers from China

$$K_U^e = \begin{bmatrix} \alpha_E K_{11}^e & \alpha_E K_{12}^e & \alpha_E \alpha_G K_{13}^e & \alpha_E \alpha_G K_{14}^e & \alpha_E \alpha_G K_{15}^e & \alpha_E K_{16}^e & \alpha_E K_{17}^e & \alpha_E \alpha_G K_{18}^e & \alpha_E \alpha_G K_{19}^e & \alpha_E \alpha_G K_{110}^e \\ \alpha_E K_{22}^e & \alpha_E \alpha_G K_{23}^e & \alpha_E \alpha_G K_{24}^e & \alpha_E \alpha_G K_{25}^e & \alpha_E K_{26}^e & \alpha_E K_{27}^e & \alpha_E \alpha_G K_{28}^e & \alpha_E \alpha_G K_{29}^e & \alpha_E \alpha_G K_{210}^e & \\ \alpha_E \alpha_G K_{33}^e & \alpha_E \alpha_G K_{34}^e & \alpha_E \alpha_G K_{35}^e & \alpha_E \alpha_G K_{36}^e & \alpha_E \alpha_G K_{37}^e & \alpha_E \alpha_G K_{38}^e & \alpha_E \alpha_G K_{39}^e & \alpha_E \alpha_G K_{310}^e & \\ \alpha_E \alpha_G K_{44}^e & \alpha_E \alpha_G K_{54}^e & \alpha_E \alpha_G K_{46}^e & \alpha_E \alpha_G K_{47}^e & \alpha_E \alpha_G K_{55}^e & \alpha_E \alpha_G K_{56}^e & \alpha_E \alpha_G K_{57}^e & \alpha_E \alpha_G K_{48}^e & \alpha_E \alpha_G K_{49}^e & \alpha_E \alpha_G K_{410}^e \\ & & & & \alpha_E K_{66}^e & \alpha_E K_{67}^e & \alpha_E K_{77}^e & \alpha_E \alpha_G K_{68}^e & \alpha_E \alpha_G K_{69}^e & \alpha_E \alpha_G K_{610}^e \\ & & & & & & & \alpha_E \alpha_G K_{78}^e & \alpha_E \alpha_G K_{79}^e & \alpha_E \alpha_G K_{710}^e \\ & & & & & & & \alpha_E \alpha_G K_{88}^e & \alpha_E \alpha_G K_{89}^e & \alpha_E \alpha_G K_{810}^e \\ & & & & & & & \alpha_E \alpha_G K_{99}^e & \alpha_E \alpha_G K_{910}^e & \alpha_E \alpha_G K_{1010}^e \\ & Sym & & & & & & & & \end{bmatrix} \quad (21)$$

Table 13

Natural frequencies (Hz) of the reduced-order model with damage.

Mode	Natural frequency (Hz)	Mode	Natural frequency (Hz)
1	0.1123	11	1.0209
2	0.1705	12	1.1177
3	0.3410	13	1.2314
4	0.3920	14	1.4878
5	0.4135	15	1.7183
6	0.4552	16	1.7963
7	0.4769	17	1.9561
8	0.7378	18	1.9722
9	0.8968	19	2.1831
10	0.9796	20	2.3417

and Hong Kong made effort to use several methods for the detection of damage in this structure. Canton Tower was the tallest building in the world in 2009. This tower was built for the Asian Games in Guangzhou, China, the following year. This is also a highlight of human progress in conquering the heights in the design of construction works.

The main component of the Canton Tower is a 454 m high. Besides, it has a 146 m high antenna mast in the top. The main component of tower is a core with reinforced concrete structure that is covered by a steel structure type of lattice. Both components of the main structure have elliptical cross-sections. However, the cover of structure varies over height. It decreases from 50 m × 80 m to 20.65 m × 27.5 m before going up to 41 m × 55 m. Meanwhile, the cross-section of inner main tower is a fixed ellipse with 14 m × 17 m from base to top. The rest of tower is the antenna mast, which is a steel structure with an octagonal cross-section.

It is too complex to build matrix of the full-scale model (see Fig. 11(a) and (b)). In this study, we use the reduce-order FEM model (see in see Fig. 11(c)), which is presented by Weiler [46] and Karplus et al. [47] as shown in Eq. (21).

For the convenience of calculation, the tower is simulated as a beam, which is a type of three-dimensional cantilever consisting of 37 vertical components. Each vertical component is located as indicated in Table 11. Each component is a three-dimensional bar with mass concentrated at the top. By doing so, DOFs of each component are two horizontal translations (X, Y) and three rotations ($\theta_x, \theta_y, \theta_z$). Therefore, a total of 185 DOFs is simulated for the reduced-order model.

Each floor of the Canton Tower has a different geometrical centroid location. The centroidal axis of the antenna mast is assumed at the centroidal axis of the new reduced-order model. For this reason, there is a difference between the reduced-order model element stiffness matrix and Euler-Bernoulli beam. To solve this problem, whereas the mass matrices are fixed, element stiffness matrices must be modified to fit the natural frequencies of the full-scale model to the reduced-order one. Two factors α_E and α_G are added in the element stiffness matrices to eliminate this difference.

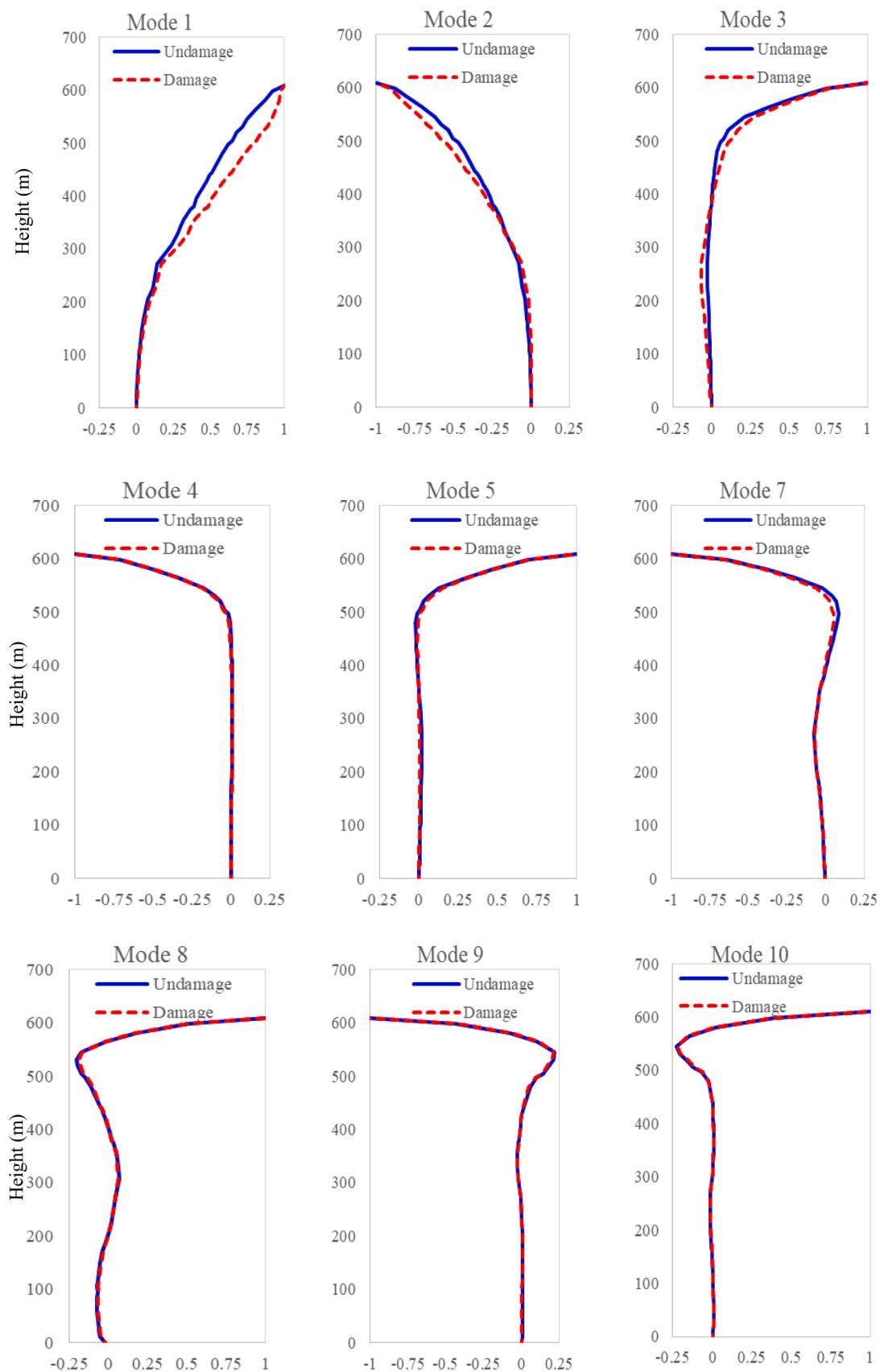


Fig. 12. Mode shapes in the X direction for undamaged tower and damage scenario.

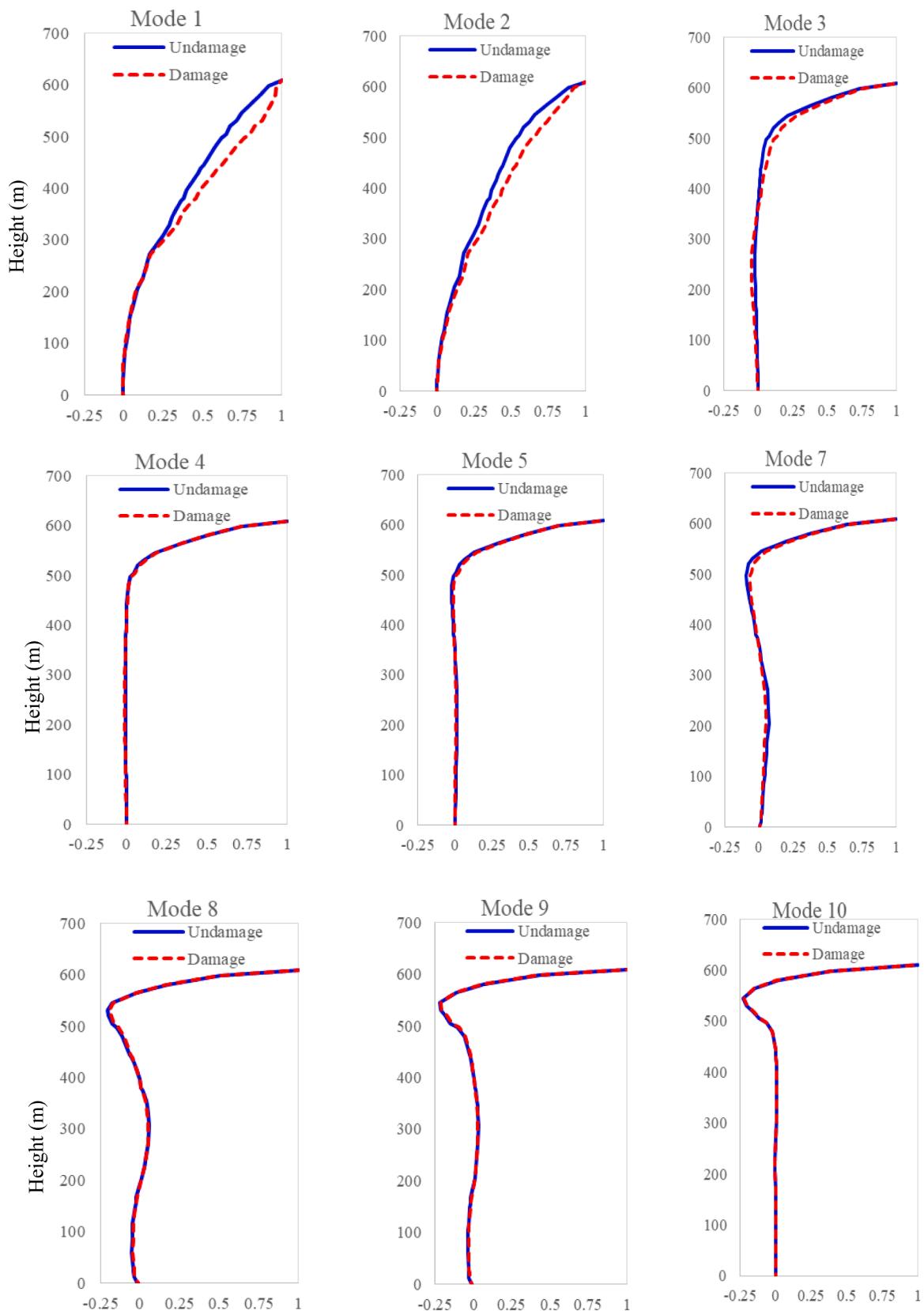


Fig. 13. Mode shapes in the Y direction for undamaged tower and damage scenario.

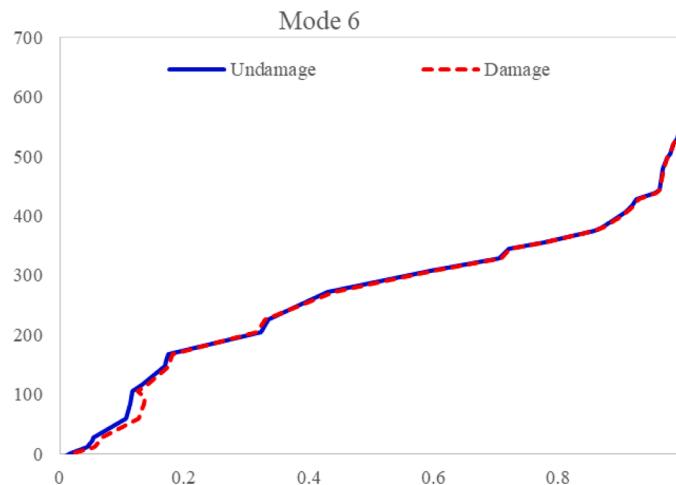


Fig. 14. Torsional mode shapes for undamaged tower and damage scenario.

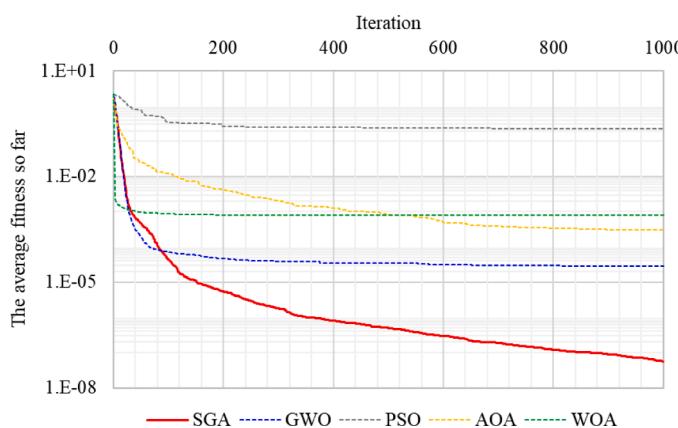


Fig. 15. Convergence curve of the average fitness after 10 independent runs.

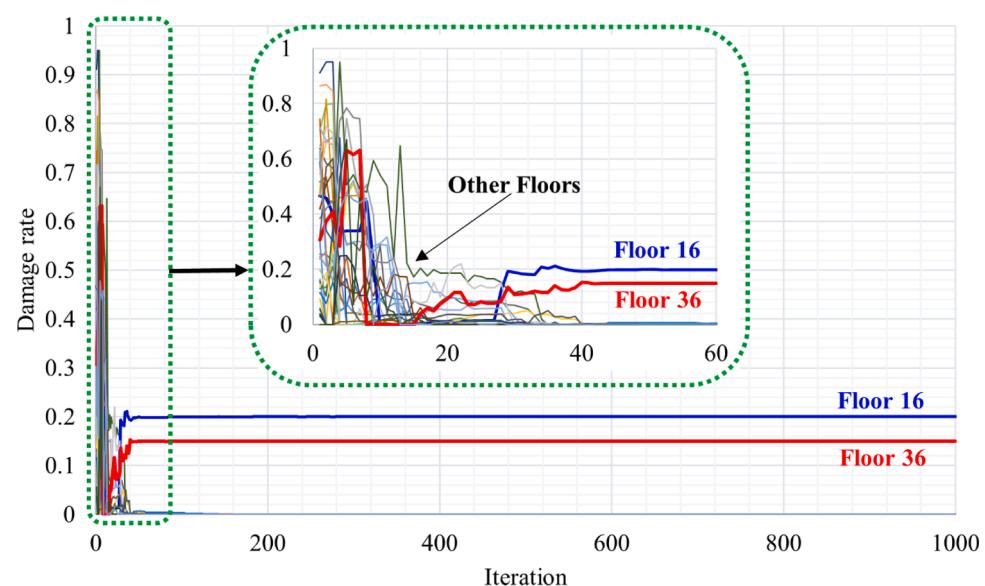


Fig. 16. Trajectory of the damage severity using SGA.

where K_M^e is the modified element stiffness matrix, and K_{ij}^e is the initial element stiffness matrix. While coefficient α_E illustrates the modified modulus that is associated with all entries in the element stiffness matrix, α_G indicates the modified bending and rotational coefficient of DOFs only. Therefore, a typical modified element stiffness matrix can be seen as in Eq. (21).

To predict the damage for this structure, we propose a damage scenario at two floors. More precisely, its severity is 20% and 15% at 16th and 36th floor, respectively. SGA is used as a tool for forecasting damage. Objective function is also a combination between natural frequencies and mode shapes as shown in Eq. (20).

Table 12 lists the natural frequencies of the full-scale model and reduced-order one, respectively. Clearly, the difference in the natural frequencies between full-scale and reduced-order model is small. Therefore, the reduced-order model can be used for damaged forecasting. The reduce-order model supporting the calculation is simpler than original model, but its results are reliable.

Table 13 shows the natural frequencies of reduced-order model with the proposed damage scenario. Besides, some mode shapes of the damaged and the healthy structure are presented in Figs. 12–14.

Fig. 15 illustrates the historical process of the match of objective function over each generation. We can see that SGA is faster than other algorithms to get the promising results, even in the first iterations. Besides, SGA also shows its great exploitation to enhance the accuracy of the algorithm in the last iterations.

The whole updating process for damaging level evaluation of each floor is presented in Fig. 16. SGA is stable in the most members at the same time. Although GWO, AOA, PSO and WOA undergo 500 iterations, these results cannot be competitive with SGA at 200-iteration as shown in Fig. 15.

The severity of damage is indicated in Fig. 17. Due to the poor results of PSO, the damage location could not be detected correctly. Meanwhile the group, including GWO, AOA and SGA, can provide a positive detection. The difference in the prediction of GWO and true measurement is 2% at the 22th, whereas, this value in SGA is small. Additionally, this difference when applying SGA does not exceed 0.1% in the rest of

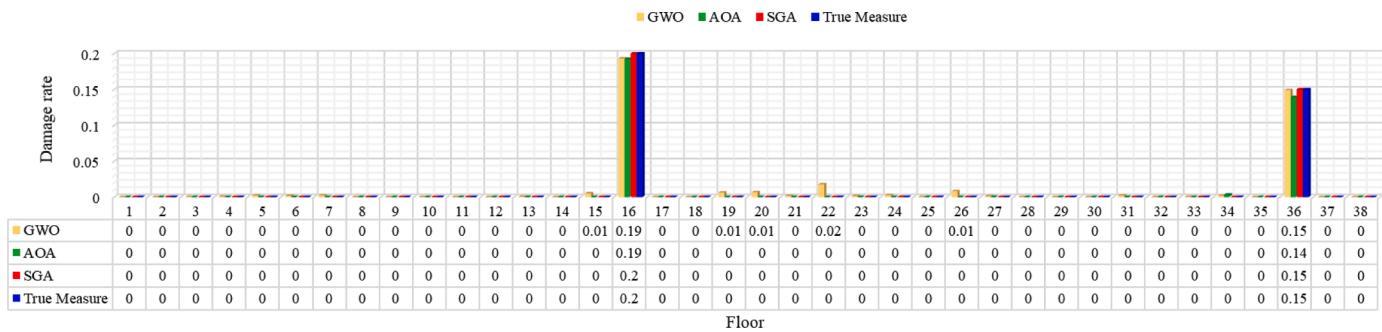


Fig. 17. Comparison between the calculated and measured damage detection results.

floors. Thus, the comparison of SGA with other algorithms with true measurement proves that the accuracy of SGA is not only better than other algorithms, but also suitable for the proposed damage scenario.

6. Conclusions

This current work provides the application of an optimization method inspired by the general behaviours of Shrimp and Goby Association (SGA) to address very complicated optimization problems. SGA was tested in high dimensions using 13 well-known test functions, and it was discovered that this algorithm was very efficient in terms of exploration and exploitation. The search history of SGA is also examined for 10 fixed-dimension multimodal benchmark functions. We have also applied the proposed algorithm to two real-world engineering challenging problems, namely a 25-bar truss and a 72-bar truss.

For the first time, SGA was used to optimize the damage detection process by obtaining the best solutions in SHM field. Performance comparison of SGA with other similar metaheuristic optimization algorithms shows its effective applicability to complex problems.

CRediT authorship contribution statement

Thanh Sang-To: Investigation, Methodology, Validation, Writing – original draft. **Hoang Le-Minh:** Investigation. **Magd Abdel Wahab:** Conceptualization, Supervision, Validation, Writing – review & editing. **Cuong-Le Thanh:** Conceptualization, Supervision, Validation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgments

The authors acknowledge the financial support of VLIR-UOS TEAM Project, VN2018TEA479A103, 'Damage assessment tools for Structural Health Monitoring of Vietnamese infrastructures', funded by the Flemish Government.

The authors wish to express their gratitude to Van Lang University, Vietnam for financial support for this research. The authors would like to acknowledge the support from Ho Chi Minh City Open University under the basic research fund (E2022.01.1CD).

References

- [1] Nanthakumar S, Lahmer T, Zhuang X, Zi G, Rabczuk T. Detection of material interfaces using a regularized level set method in piezoelectric structures. *Inverse Prob Sci Eng* 2016;24(1):153–76.
- [2] Samaniego E, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications. *Comput Meth Appl Mech Eng* 2020;362:112790.
- [3] Anitescu C, Atroshchenko E, Alajlan N, Rabczuk T. Artificial neural network methods for the solution of second order boundary value problems. *Comput Mater Contin* 2019;59(1):345–59.
- [4] Nguyen-Thanh VM, Anitescu C, Alajlan N, Rabczuk T, Zhuang X. Parametric deep energy approach for elasticity accounting for strain gradient effects. *Comput Meth Appl Mech Eng* 2021;386:114096.
- [5] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the ICNN’95-international conference on neural networks. 4. IEEE; 1995. p. 1942–8.
- [6] Gandomi AH, Yang XS, Alavi AH, Talatahari S. Bat algorithm for constrained optimization tasks. *Neural Comput Appl* 2013;22(6):1239–55.
- [7] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H. Harris hawks optimization: algorithm and applications. *Future Gener Comput Syst* 2019;97: 849–72.
- [8] Meraih I, Ramdane-Cherif A, Acheli D, Mahseur M. Dragonfly algorithm: a comprehensive review and applications. *Neural Comput Appl* 2020;32(21): 16625–46.
- [9] Mirjalili S, Lewis A. The whale optimization algorithm. *Adv Eng Softw* 2016;95: 51–67.
- [10] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Adv Eng Softw* 2014;69: 46–61.
- [11] Yang X-S. A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer; . p. 65–74.
- [12] Sang-To T, Le-Minh H, Mirjalili S, Wahab MA, Cuong-Le T. A new movement strategy of grey wolf optimizer for optimization problems and structural damage identification. *Adv Eng Softw* 2022;173:103276.
- [13] Le-Duc T, Nguyen QH, Nguyen-Xuan H. Balancing composite motion optimization. *Inf Sci* 2020;520:250–70.
- [14] Goldberg, D.E., Holland, J.H., "Genetic algorithms and machine learning," 1988.
- [15] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. *Inf Sci* 2009;179(13):2232–48.
- [16] Rao R. Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decis Sci Lett* 2016;5(1):1–30.
- [17] To TS, Le MH, Danh TT, Khatir S, Abdel Wahab M, Le TC. Combination of intermittent search strategy and an improve particle swarm optimization algorithm (IPSO) for model updating. *Fratutura ed Integrità Strutturale-Fract Struct Integr* 2022;16(59):141–52.
- [18] Minh HL, Khatir S, Rao RV, Abdel Wahab M, Cuong-Le T. A variable velocity strategy particle swarm optimization algorithm (VVS-PSO) for damage assessment in structures. *Eng Comput* 2021;1–30.
- [19] Minh HL, Khatir S, Wahab MA, Cuong-Le T. An enhancing particle swarm optimization algorithm (EHVPSO) for damage identification in 3D transmission tower. *Eng Struct* 2021;242:112412.
- [20] Minh HL, Sang-To T, Danh TT, Phu NN, Abdel Wahab M, Cuong-Le T. A two-step approach for damage detection in a real 3D tower using the reduced-order finite element model updating and atom search algorithm (ASO). In: Proceedings of the 2nd international conference on structural damage modelling and assessment. Springer; 2022. p. 13–26.
- [21] Minh HL, Sang-To T, Wahab MA, Cuong-Le T. A new metaheuristic optimization based on K-means clustering algorithm and its application for structural damage identification in a complex 3D concrete structure. *Knowl Based Syst* 2022;251: 109189. <https://doi.org/10.1016/j.knosys.2022.109189>.
- [22] John H. Holland. genetic algorithms. *Sci Am* 1992;267(1):44–50.
- [23] Koza JR, Koza JR. Genetic programming: on the programming of computers by means of natural selection. MIT Press; 1992.
- [24] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 1997;11(4):341–59.

- [25] Fleetwood K. An introduction to differential evolution. In: Proceedings of the mathematics and statistics of complex systems (MASCOS) one day symposium; 2004. p. 785–91. *26th November, Brisbane, Australia*.
- [26] Knowles J, Corne D. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: Proceedings of the congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). 1. IEEE; 1999. p. 98–105.
- [27] Moghaddam, F.F., Moghaddam, R.F., Cheriet, M., "Curved space optimization: a random search based on general relativity theory," *arXiv preprint arXiv:1208.2214*, 2012.
- [28] Sang-To T, Hoang-Le M, Khatir S, Mirjalili S, Wahab MA, Cuong-Le T. Forecasting of excavation problems for high-rise building in Vietnam using planet optimization algorithm. *Sci Rep* 2021;11(1):1–10. <https://doi.org/10.1038/s41598-021-03097-y>.
- [29] Sang-To T, Hoang-Le M, Wahab MA, Cuong-Le T. An efficient planet optimization algorithm for solving engineering problems. *Sci Rep* 2022;12(1):1–18.
- [30] Zheng YJ. Water wave optimization: a new nature-inspired metaheuristic. *Comput Oper Res* 2015;55:1–11. <https://doi.org/10.1016/j.cor.2014.10.008>.
- [31] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Comput Intell Mag* 2006;1(4):28–39.
- [32] Yang XS, Deb S. Cuckoo search via Lévy flights. In: Proceedings of the world congress on nature & biologically inspired computing (NaBIC). IEEE; 2009. p. 210–4.
- [33] Minh HL, Sang-To T, Wahab MA, Cuong-Le T. Structural damage identification in thin-shell structures using a new technique combining finite element model updating and improved Cuckoo search algorithm. *Adv Eng Softw* 2022;173:103206.
- [34] Yang XS, Gandomi AH. Bat algorithm: a novel approach for global engineering optimization. *Eng Comput* 2012;29(5):464–83. <https://doi.org/10.1108/02644401211235834>.
- [35] Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 2017;114:163–91. <https://doi.org/10.1016/j.advengsoft.2017.07.002>. 2017/12/01/.
- [36] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 2016;27(4):1053–73.
- [37] Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1997;1(1):67–82.
- [38] Karplus I, Szlep R, Tsuranmal M. Associative behavior of the fish *Cryptocentrus cryptocentrus* (Gobiidae) and the pistol shrimp *Alpheus djiboutensis* (Alpheidae) in artificial burrows. *Mar Biol* 1972;15(2):95–104.
- [39] Kramer A, Van Tassel JL, Patzner RA. A comparative study of two goby shrimp associations in the Caribbean Sea. *Symbiosis* 2009;49(3):137–41.
- [40] Lipowski A, Lipowska D. Roulette-wheel selection via stochastic acceptance. *Physica A* 2012;391(6):2193–6.
- [41] Dworschak PC, Ott JA. Decapod burrows in mangrove-channel and back-reef environments at the Atlantic barrier reef, Belize. *Ichnos Intl J Plant Anim* 1993;2(4):277–90.
- [42] Camp CV, Bichon BJ. Design of space trusses using ant colony optimization. *J Struct Eng* 2004;130(5):741–51.
- [43] Adeli H, Kamal O. Efficient optimization of space trusses. *Comput Struct* 1986;24(3):501–11.
- [44] Abualigah L, Diabat A, Mirjalili S, Elaziz MA, Gandomi AH. The arithmetic optimization algorithm. *Comput Meth Appl Mech Eng* 2021;376:113609.
- [45] Karplus I. The association between gobiid fishes and burrowing alpheid shrimps. *Oceanogr Mar Biol* 1987;25:507–62.
- [46] Weiler D. Burrow-dwelling fishes in a back-reef area and their relation to sediment grain size. Mayaguez, Puerto Rico: University of Puerto Rico; 1976.
- [47] Karplus I, Szlep R, Tsuranmal M. The burrows of alpheid shrimp associated with gobiid fish in the northern Red Sea. *Mar Biol* 1974;24(3):259–68.