Engineering Optimization

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

Size optimization of planar truss systems using the modified salp swarm algorithm

Onur Altay, Oguzhan Cetindemir & Ibrahim Aydogdu

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2022.2160449

Taylor & Francis
Taylor & Francis Group

Check for updates

RESEARCH ARTICLE

Size optimization of planar truss systems using the modified salp swarm algorithm

Onur Altay [a,b], Oguzhan Cetindemir [c] and Ibrahim Aydogdu [d]

[a]Civil Engineering Department, Antalya Bilim University, Antalya, Turkey; [b]Earthquake Engineering Department, Kandilli Observatory and Earthquake Research Institute, Bogazici University, Istanbul, Turkey; [c]Civil Engineering Department, Gebze Technical University, Kocaeli, Turkey; [d]Civil Engineering Department, Akdeniz University, Antalya, Turkey

ABSTRACT

This work evaluates the performance of the salp swarm algorithm (SSA) for truss system optimization problems and presents a novel method called the modified salp swarm algorithm (MSSA). Five truss structures, previously optimized by metaheuristics and containing discrete and continuous variables, were used for the evaluations. Size and size–shape optimization types were considered. Although the SSA performs poorly and has convergence issues in initial random solutions, it reaches comparable solutions to previously published results, particularly in continuous problems. In contrast, the MSSA achieves the best solutions for discrete problems and is relatively close to the best results in the reference literature on continuous problems. Moreover, the MSSA convergence curves exhibit a modest increase in convergence rates, especially for discrete problems. It is envisaged that the findings will contribute to improving solution performance, convergence speed and security for future real-world applications.

ARTICLE HISTORY
Received 28 May 2022
Accepted 14 December 2022

KEYWORDS
Structural optimization; metaheuristic techniques; swarm intelligence; truss systems; salp swarm algorithm

1. Introduction

Engineering design problems are quite challenging because of many constraints, such as strength, displacement limits and matrix computations, requiring complex solutions. These problems are also difficult to optimize and cannot be solved effectively using classical derivative-based mathematical optimization techniques. Metaheuristic techniques are overly critical in solving such problems. Novel metaheuristic methodologies, particularly swarm intelligence techniques, are phenomenally successful at tackling structural optimization problems. Although metaheuristic or swarm intelligence techniques are suitable for discrete engineering problems, one of the most significant drawbacks is that they employ stochastic solutions, which lack stability or constancy of performance in their results. Therefore, metaheuristic techniques yield reliable results for a specific mathematical problem but may have decreased performance when applied to another optimization problem. Thus, such techniques may require validation for various problems, and performing such analyses provides essential insight into the robustness of the solutions. As a result, applying recently developed optimization techniques to different problems is beneficial.

Since truss structures are a frequently used optimization problem in the literature, several metaheuristic algorithms have been developed to solve similar problems. The general categories of

CONTACT Ibrahim Aydogdu aydogdu@akdeniz.edu.tr

Supplemental data for this article can be accessed here. https://doi.org/10.1080/0305215X.2022.2160449

metaheuristic algorithms are (1) local search-based, (2) evolutionary search-based, and (3) swarm search-based algorithms (Abualigah *et al.* 2020). Local search-based algorithms consist of four different algorithms, namely simulated annealing, β-hill climbing, tabu search and hill-climbing. Some problems include truss optimization using the local search-based algorithms simulated annealing (Lamberti 2008; Millan-Paramo and Abdalla Filho 2020) and tabu search (Li and Lian 2018; Bennage and Dhingra 1995). To the best of the authors' knowledge, examples of β-hill climbing and hill-climbing algorithms are lacking in the literature for solving truss optimization problems. The evolutionary search-based algorithms, including genetic algorithms (Galante 1996; Lingyun *et al.* 2005; Xu *et al.* 2010), genetic programming (Rajeev and Krishnamoorthy 1997; Yang and Soh 2002; Assimi, Jamali, and Nariman-Zadeh 2017), differential evaluation algorithms (Wu and Tseng 2010; Bureerat and Pholdee 2016) and harmony search algorithms (Lee and Geem 2004; Degertekin 2012; Miguel and Miguel 2012), perform well in truss optimization. Finally, the third category is swarm-based algorithms, which include but are not limited to the krill herd algorithm (Gandomi and Alavi 2016), particle swarm algorithm (Gomes 2011; Perez and Behdinan 2007; Kaveh and Javadi 2014), cuckoo search algorithm (Gandomi *et al.* 2013; Cuong-Le *et al.* 2021), artificial bee colony algorithm (Sonmez 2011; Jawad *et al.* 2021), biogeography-based optimization (Jalili and Hosseinzadeh 2018; Jalili, Hosseinzadeh, and Taghizadieh 2016; Aydogdu, Carbas, and Akin 2017) and a few newly developed metaheuristic techniques, including electrostatic discharge algorithms (Aydogdu, Ormecioglu, and Carbas 2021) and search group algorithm (Gonçalves, Lopez, and Miguel 2015). Truss optimization is also projected as a test problem in many novel metaheuristic algorithms. Therefore, this work aims to evaluate truss optimization in benchmark problems, and a novel contribution to the literature will be offered by developing a modified metaheuristic algorithm.

As stated previously, metaheuristic methods can effectively solve many engineering problems. The salp swarm algorithm (SSA) is a recently developed metaheuristic algorithm that has never been applied to solving truss structure optimization, which is one of the conventional structural optimization problems. Mirjalili *et al.* (2017) developed the SSA, a population-based metaheuristic optimization method inspired by swarm intelligence. Although the SSA was developed recently, numerous applications of SSA have been documented in various fields. For instance, the SSA has been used to optimize benchmarks and has addressed real-world problems. Abualigah *et al.* (2020) conducted a comprehensive SSA survey. They summarized the SSA applications as machine learning applications (feature selection and training neural networks), engineering applications (scheduling and control of power systems), image processing, wireless networking and other salp swarm applications. Nevertheless, regarding these solutions, it is concluded that the SSA has not been used to optimize truss systems. Therefore, in this work, the performance of the SSA in truss optimization problems was investigated. In addition, when assessing the SSA, it was discovered that the performance of the algorithm was not highly effective. Therefore, a modified version of the SSA was also developed by reinforcing the deficient and weak characteristics of the original algorithm. While this study presents a new algorithm to the swarm-based algorithm literature, both algorithms are applied for the first time to optimize truss systems, one of the most well-studied structural optimization systems with discrete and continuous variables. For this purpose, a novel algorithm called the modified salp swarm algorithm (MSSA), which optimizes truss systems, was developed. The performance of the MSSA was assessed on five different benchmark problems. Then, the performance of the SSA and the MSSA was compared with metaheuristic methods previously analysed in the literature.

2. Optimization methods: SSA and MSSA

In this section, the SSA and the newly developed MSSA methodology used in this work are introduced. First, a brief introduction to natural phenomena, including the swarm movement and the salp chain, is provided. Then, the SSA methodology, inspired by natural phenomena, is shared. Finally, the SSA and MSSA to be used for truss optimization problems are described in depth with a comprehensive description that includes the equations used by the algorithms and the definitions of each parameter. In addition, two novel improvements are made to enhance the performance of the SSA, which are vital aspects of this research that express the merits of this work. The flowchart and pseudo-code for SSA are provided, along with the necessary equation modifications for the MSSA.

2.1. Natural phenomena

More than 1.2 million marine species have already been catalogued in a centralized database, which serves as a resource for researchers (McCauley *et al*. 2015). Most of these species exhibit behaviours and features that are comparable to one another, such as communication strategies, locomotor performance and food-seeking behaviours. A salp is a kind of marine creature that is classified as a member of the Salpidae family and has a translucent, barrel-shaped body. The tissues, forms and movements of these creatures are very similar to those of jellyfish (Madin 1990). They have a cylindrical form with openings at the ends that allow water to flow through their gelatinous bodies, allowing them to move and feed via internal feeding filters. They often form a swarm in deep waters, known as a salp chain. This is one of the salps' fascinating behaviours, which may improve their mobility through quick, coordinated adjustments and foraging (Anderson and Bone 1980).

2.2. Methodology

The SSA was inspired and developed by mimicking the behaviours of salps in nature. To model the salp chains mathematically, the population is separated into two groups: a leader and followers. The salp at the head of the chain is called the leader, while the remaining salps are considered followers. As the name suggests, the leader of these salps directs the swarm while the followers follow one another. Since salps are defined in an n-dimensional search space, the positions of all salps are kept in a two-dimensional matrix. In addition, the SSA defines a food supply in the search space as the swarm's target. The leader's position is solely updated with the food supply. The position of the followers is updated by Newton's law of motion.

2.3. SSA for the truss optimization problem

In a truss optimization problem, the SSA attempts to determine the section area corresponding to the elements constituting the best-suited truss system to optimize it as much as possible. The terminology used in the SSA for the truss optimization problem is described as follows:

Algorithm memory ($[X]$): A matrix that contains design variables, fitness values and objective functions of the truss designs that are generated from the SSA.
Salps ($\vec{x}$): A candidate truss design/a truss design.
Number of salps (N_{ss}): The number of truss designs stored in the $[X]$.
Position of the salp swarm ($x_j; j \in [1, 2, \ldots ndv]$): Any design variable value of the truss.
Position change of the salp: Modification of the truss design.
Performance of the salp ($fit_i; i \in [1, 2, \ldots, N_{ss}]$): The fitness value of the truss design.
Salp having the best performance (fit_{best}): The global best solution in the optimization problem.

According to the descriptions and definitions, the main steps in the SSA for the truss optimization problem are detailed as follows.

Step 1. Initialization: The SSA creates initial truss designs randomly, formulated as follows:

$$[X]_{i,j} = lb_j + (ub_i - lb_j) * rnd; i = 1, 2, \ldots, N_{ss}; j = 1, 2, \ldots ndv \quad (1)$$

where rnd is a function that returns floating-point numbers between 0 and 1 that are drawn from a uniform distribution. ub_i and lb_j are the upper and lower boundaries of the jth design variable. The algorithm evaluates the initial designs and calculates their fitness values according to the criteria described in the online supplemental material (Altay, Çetindemir, and Aydoğdu 2022). Algorithm memory is created by recording the design variables, fitness and penalty values of the solutions.

Step 2. Movement of the leader: There are two movement models defined in the SSA: the movement of the leader and the movement of the followers. For leader movement, first, the algorithm sorts its solutions in descending order of their fitness values and assigns the first solution (truss design) in the ranking as the leader salp. Then, the SSA updates the position of the leader salp, as follows:

$$[X]^{new}{}_{1,j} = \begin{cases} [X]_{1,j} + c_1(rnd(ub_i - lb_j) + lb_j)rnd \geq 0.5 \\ [X]_{1,j} - c_1(rnd(ub_i - lb_j) + lb_j)rnd < 0.5 \end{cases}$$
$$j = 1, 2, \ldots ndv \tag{2}$$

where c_1 is the parameter that sets the salp swarm movement direction and size. The value of c_1 is particularly important to balance exploration and exploitation, and is determined as follows:

$$c_1 = 2e^{-\left(\frac{4 \cdot iter}{iter_{\max}}\right)^2} \tag{3}$$

where $iter$ and $iter_{\max}$ represent, respectively, the current iteration and maximum iteration number defined in the problem. The updated solution (truss design) is evaluated, and its fitness value is calculated. If the fitness value of the new truss design is greater than the fitness value of the old design, the new truss design takes the place of the old truss design. This procedure is named greedy selection.

Step 3. Movement of the followers: In this step, the follower salps (remaining truss designs) update their positions (change their design) relative to the leader salp. The movement is performed sequentially in order of Newton's law of motion. The SSA reformulates Newton's law of motion by considering the time as an iteration and setting the variety between iterations as 1 (Equation 4):

$$[X]^{new}{}_{i,j} = \frac{1}{2}([X]_{i,j} + [X]_{i-1,j}); i = 2, 3, \ldots, N_{ss}; j = 1, 2, \ldots ndv \tag{4}$$

Updated positions of the follower salps are evaluated, and their fitness value is calculated. The algorithm applies a greedy selection procedure for all follower salps.

After Step 3, one iteration is completed, and the SSA goes back to Step 2. The SSA repeats the procedures between Steps 2 and 3 until the maximum iteration number is reached.

2.4. MSSA for the truss optimization problem

As a result of the tests (Section 3), it was seen that SSA could not adjust the balance of exploration and exploitation at the beginning and end of the optimization process. The following two improvements were made to address this shortcoming of SSA.

2.4.1. Modification of parameter c_1

While the classical SSA has achieved satisfactory results regarding solving mathematical problems in the literature, it has been demonstrated in tests (see Section 3) that the SSA has significant convergence problems in solving structural optimization problems. Convergence problems occur particularly in the initial stages of the solution process, and the SSA cannot improve the solutions. Figure 1 illustrates the search histories for the lowest and average values acquired by the current cycle of SSA for the 10-bar truss example. According to Figure 1, the difference between the average and minimum values is quite large, demonstrating that the SSA produces highly dispersed solutions at the beginning of the optimization process. This problem persists for the first 600 cycles, and the SSA cannot improve the solutions sufficiently during this time. It was also stated in Section 2.3 that the value of c_1 is vital to prevent the generation of dispersed solutions. When the change in c_1 is investigated, c_1 variates close to linear, despite being formulated exponentially. To convert the change in c_1 into an exponential structure, the MSSA updates Equation (3) as follows. Changes in c_1 values calculated by the SSA and MSSA are shown in Figure 2.

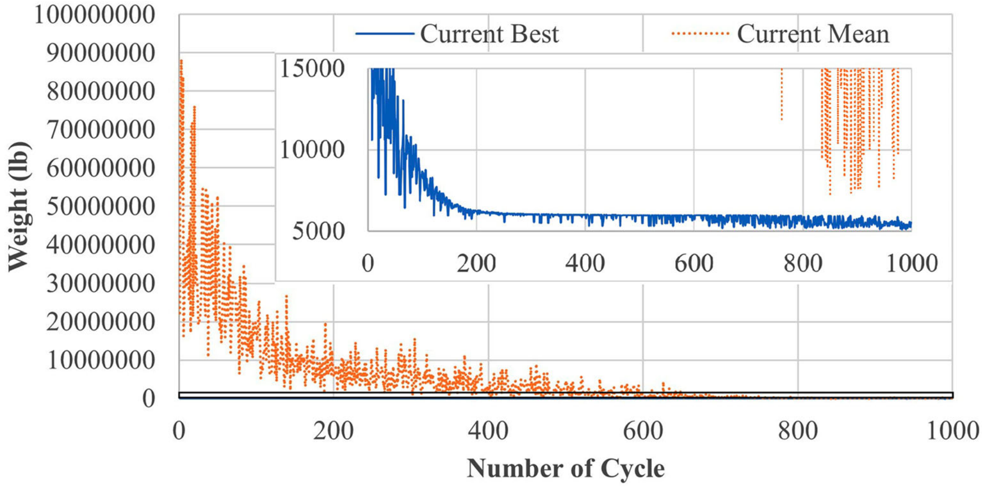

Figure 1. Comparison of the lowest and average values acquired by the current cycle of the salp swarm algorithm (SSA) for the 10-bar truss example.

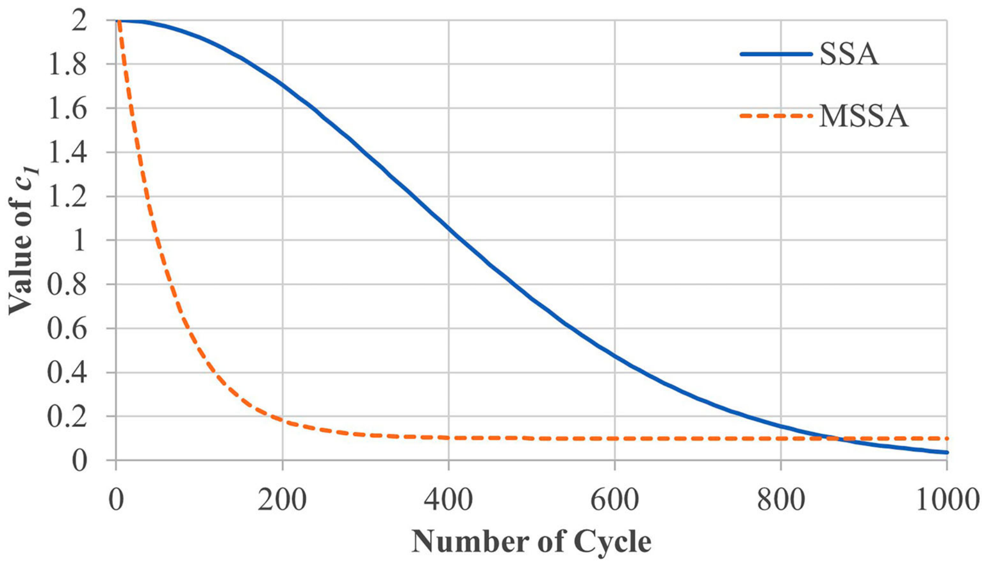

Figure 2. Changes in c_1 values calculated by the salp swarm algorithm (SSA) and modified salp swarm algorithm (MSSA).

$$c_1 = \beta - e^{-\left(\frac{16 \cdot iter}{iter_{\max}}\right)} \tag{5}$$

Here, β is a constant parameter to prevent near-zero c_1 values in progressive iterations (when $c_1 = 0$, the leader salp cannot move). Preliminary parameter work was conducted to determine the convenient β parameter, using a discrete optimization problem (47-bar truss). The convergence curves and optimum solutions are shown in Figures 3 and 4, respectively. According to the figures, the best results are obtained when the β value is between 0.01 and 0.1. Therefore, it is recommended to consider the β constant parameter between these values.

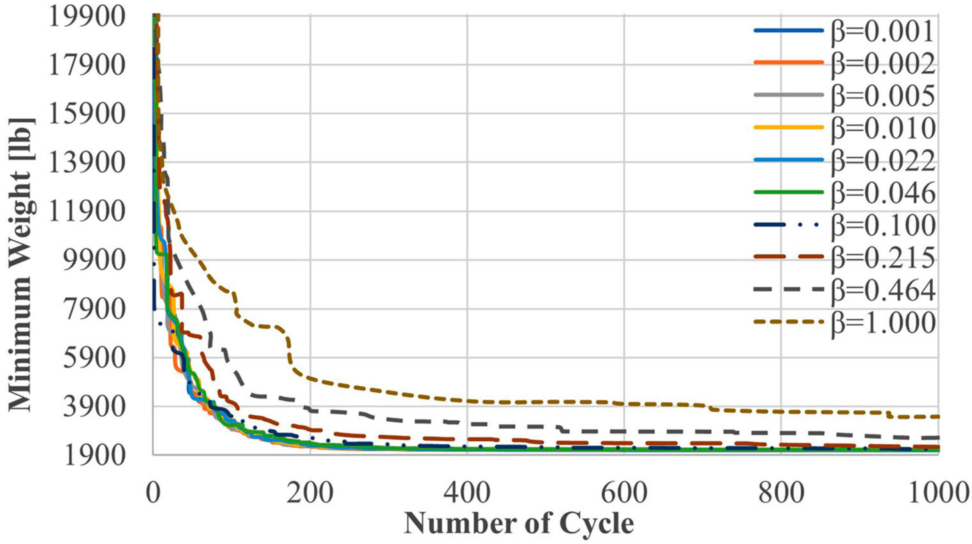

Figure 3. Convergence curves for the 47-bar truss based on various β parameters.

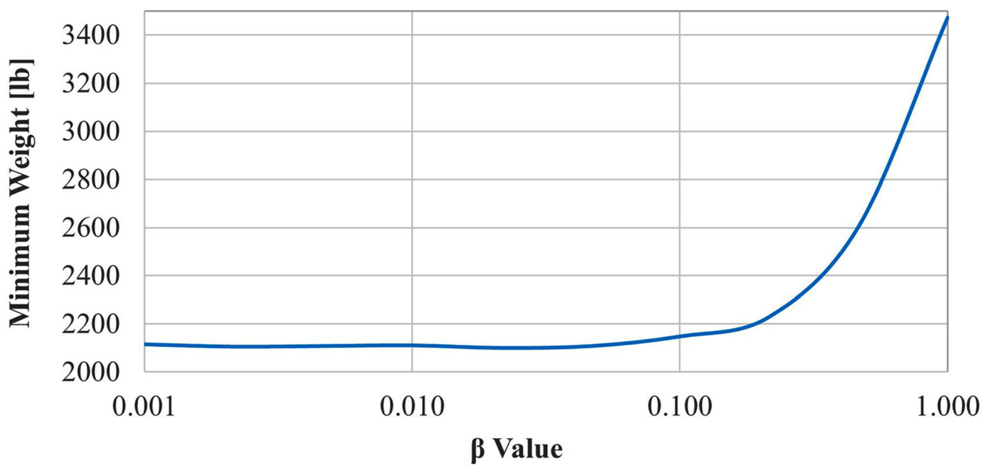

Figure 4. Minimum weight for the 47-bar truss based on the β parameter on a logarithmic scale.

2.4.2. Updating the movement of the followers' formula

Whereas the SSA method produces highly dispersed solutions at the beginning of the optimization process, exploitation becomes dominant in the last cycles. In this case, local convergences can be seen, especially in discrete problems. In the movement of the leader phase, parameter β in Equation (5) can help to overcome the stagnation problem. However, a new computation is needed to increase the exploration ability of SSA in the movement of the followers' phase. For this purpose, in the MSSA, the movement formula of followers is modified as follows:

$$[X]^{new}{}_{i,j} = \frac{1}{2}([X]_{i,j} + [X]_{k,j})$$
$$i = 1, 2, \ldots, N_{ss}; j = 1, 2, \ldots ndv; k \in [1, 2, \ldots, N_{ss}]; k \neq i \quad (6)$$

Define N_{ss} and $iter_{max}$

For i=1 to N_{ss}

    Generate initial i th swarm ($[X]_i$) using (Eq. 8)

    Evaluate $[X]_i$ and calculate its fitness value (fit_i) (Eq.7)

For iter=1 to $iter_{max}$

    Compute parameter c_1 using Eq. 10 (For MSSA use Eq. 13)

        Sort salp swarm in descending order of their fitness values.

        Modify leader salp ($[X]_1$) using Eq. 9

        For i=2 to N_{ss}

            Modify follower salps ($[X]_i$) using Eq. 12 (For MSSA use Eq. 14)

    Evaluate modified swarm and update global memory.

Figure 5. Pseudo-code of the salp swarm algorithm (SSA) and modified salp swarm algorithm (MSSA).

where subscript k represents a randomly selected salp from the algorithm memory. Figure 5 shows the pseudo-code of the SSA and MSSA. A detailed flowchart can be found in the online supplemental material.

3. Design examples

Since this article aims to assess the performance of the SSA for optimization problems of truss systems, five different benchmark problems are investigated in this research to compare the performance of the SSA and the MSSA. These are the 10-bar, 18-bar, 47-bar, 52-bar and 200-bar truss problems, which have previously been solved using various methodologies described in the literature. Two are selected as continuous (10-bar and 200-bar trusses) and the other two as discrete (47-bar and 52-bar trusses) for size optimization. Moreover, the 18-bar truss problem is analysed using discrete and continuous variables for size and shape optimization. Figures 6–9 give the graphical illustrations of these examples. Information about the design examples (load conditions, material and connection properties, and strength and displacement limitations) is given in the online supplemental material (Altay, Çetindemir, and Aydoğdu 2022).

Thirty independent tests were conducted using truss structures, the SSA and MSSA methodologies, and a variety of seed values. The area values of the designs with the lowest weight derived from 30 independent analyses, averages and standard deviation values are included in a detailed table as supplemental material, along with published data references (Altay, Çetindemir, and Aydoğdu 2022). The search histories, box-plot diagrams and tables for detailed optimum solutions can also be found in the supplemental material.

3.1. A 10-bar truss

As seen in Table 1, the results obtained with the SSA and MSSA remain between the reference results. The SSA's solution is about 0.4% heavier than the best result (Asl, Aslani, and Panahi 2013), which is more significant than the weight difference between the MSSA and Asl, Aslani, and Panahi's (2013) method, which is only 0.05%. Statistically, the MSSA has a lower average value and standard deviation than the SSA. Detailed search histories and convergence trends indicate that the MSSA has a higher

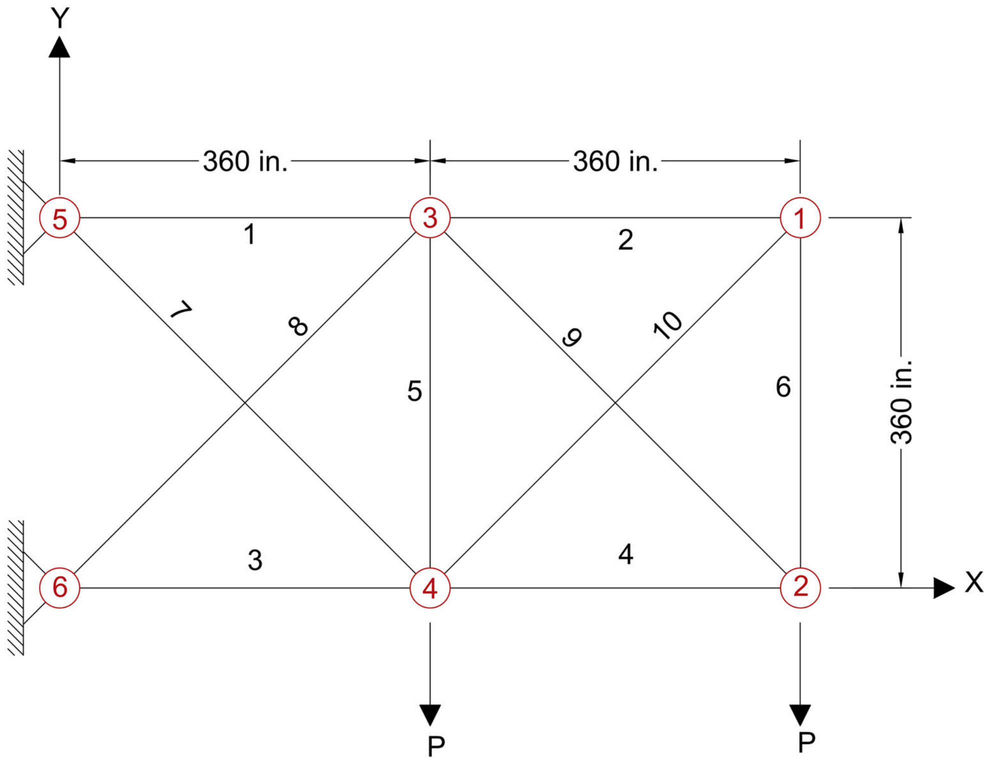

Figure 6. The 10-bar truss example.

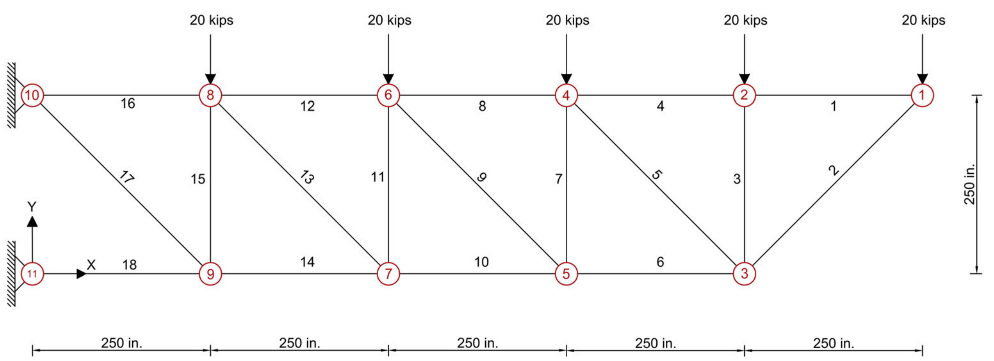

Figure 7. The 18-bar truss example.

convergence rate than the standard SSA. In addition, the MSSA reaches the optimal solution when it reaches 20% of the optimization process, whereas the standard SSA reaches the optimal solution when it reaches around 95% of the optimization process. Based on the SSA and MSSA box-plot diagrams, the range of optimum solutions for the classical SSA is wider than that for the MSSA in the test results, and the MSSA achieves comparably close optimum solutions within a much narrower range. This demonstrates that the MSSA performs more consistently than the classical SSA.

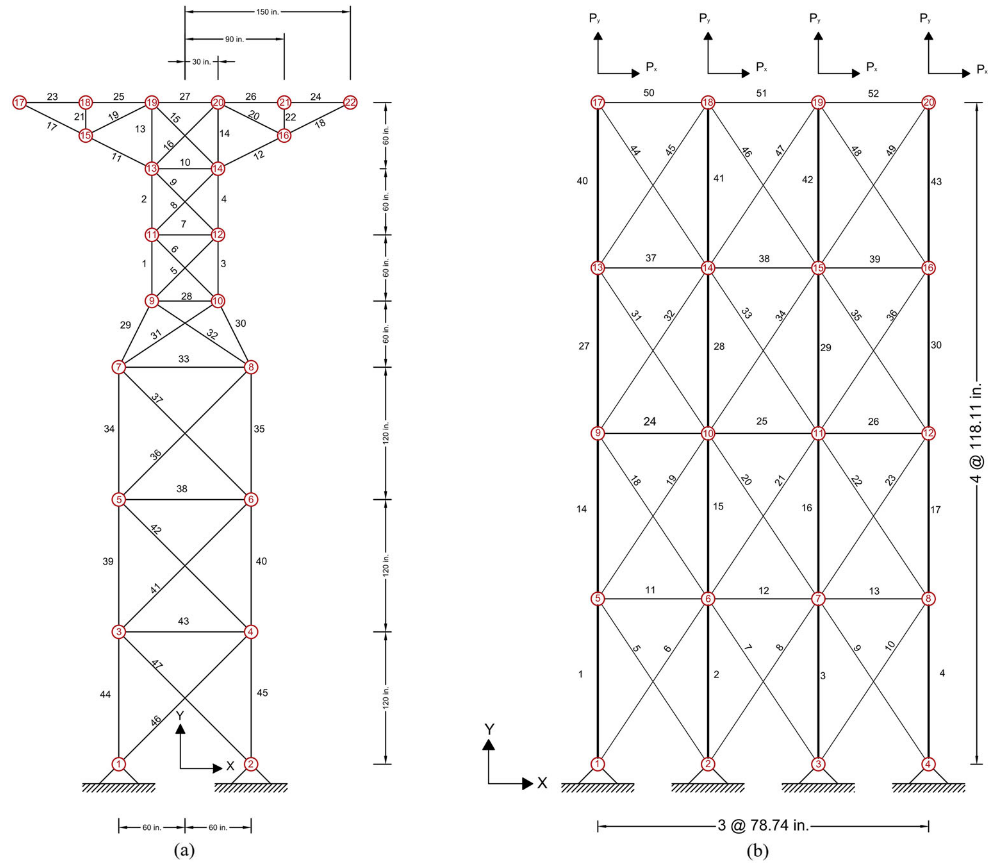

Figure 8. Discrete benchmark problems: (a) 47-bar and (b) 52-bar truss examples.

Table 1. Optimum solution details for the 10-bar truss example (weight in lb).

IHM	NEWSUMT	CA	HPSO	H-SAGA
5089.00	5076.85	5084.90	5060.92	5058.66
TLBO-MS	FBSFA	FBS-FPA	SSA	MSSA
5060.85	5060.86	5065.858	5077.50	5061.138

Note: IHM = inscribed hyperspheres method, Schmit and Farshi (1974); NEWSUMT = a FORTRAN program for inequality constrained function minimization, Schmit and Miura (1976); CA = combined approach (an energy criterion and a search procedure), Venkayya (1971); HPSO = heuristic particle swarm optimization, Li *et al*. (2007); H-SAGA = hybrid simulated annealing–genetic algorithm, Asl, Aslani, and Panahi (2013); TLBO-MS = teaching learning-based optimization (TLBO), hybridized with mapping strategy (MS), Baghlani, Makiabadi, and Maheri (2017); FBSFA = feasible boundary search (FBS) technique, hybridized with firefly algorithm (FA), Baghlani and Makiabadi (2014); FBS-FPA = feasible boundary search (FBS) technique, hybridized with flower pollination algorithm (FPA) , Anh (2016); SSA = salp swarm algorithm; MSSA = modified salp swarm algorithm.

3.2. An 18-bar truss

From the search histories and box-plot diagrams in the supplemental material, the MSSA has a higher convergence rate than the standard SSA. In addition, the MSSA reaches the optimal solution after 30% of the optimization process, whereas the standard SSA reaches the optimal solution just before 90%.

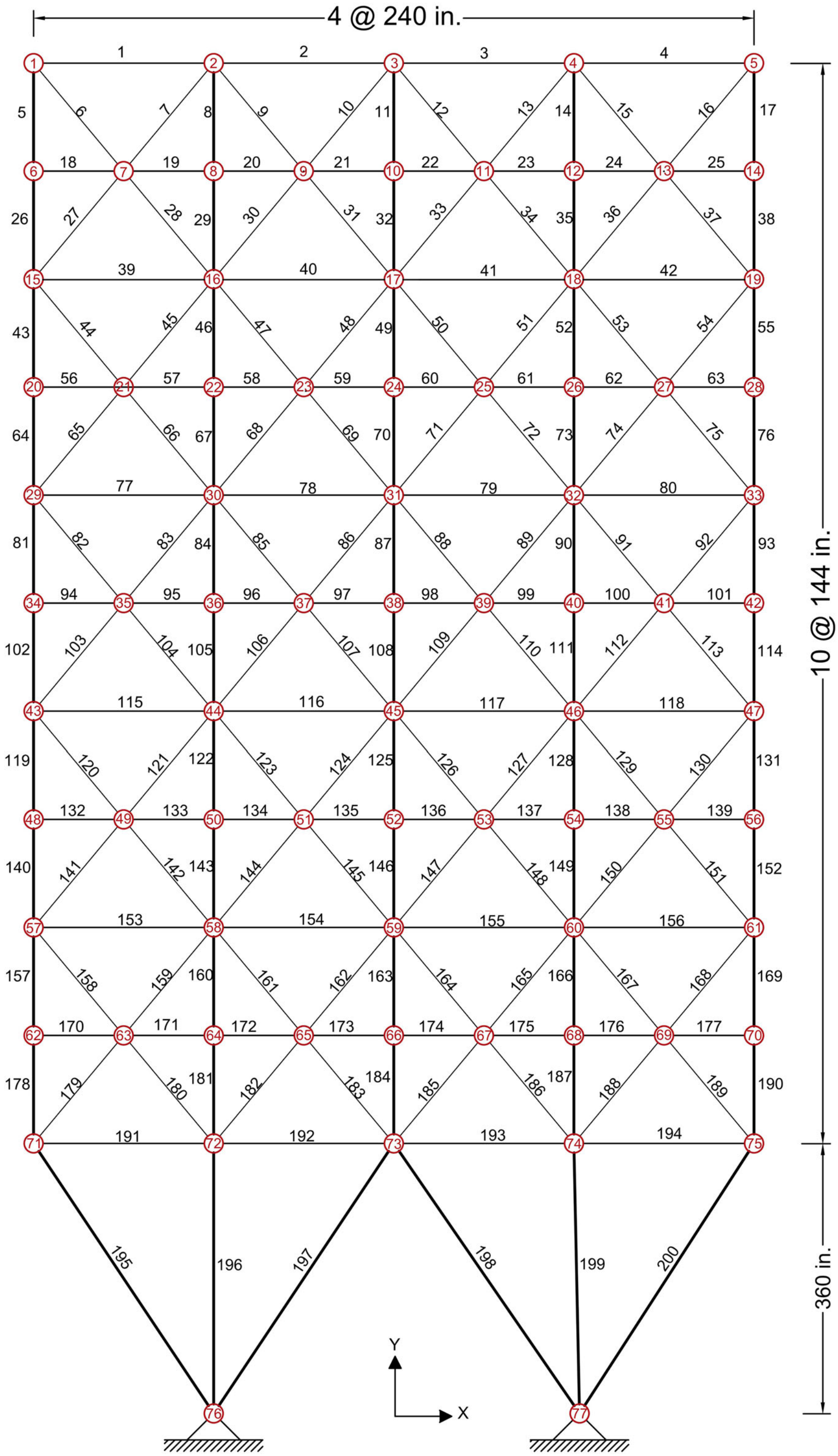

Figure 9. The 200-bar truss example.

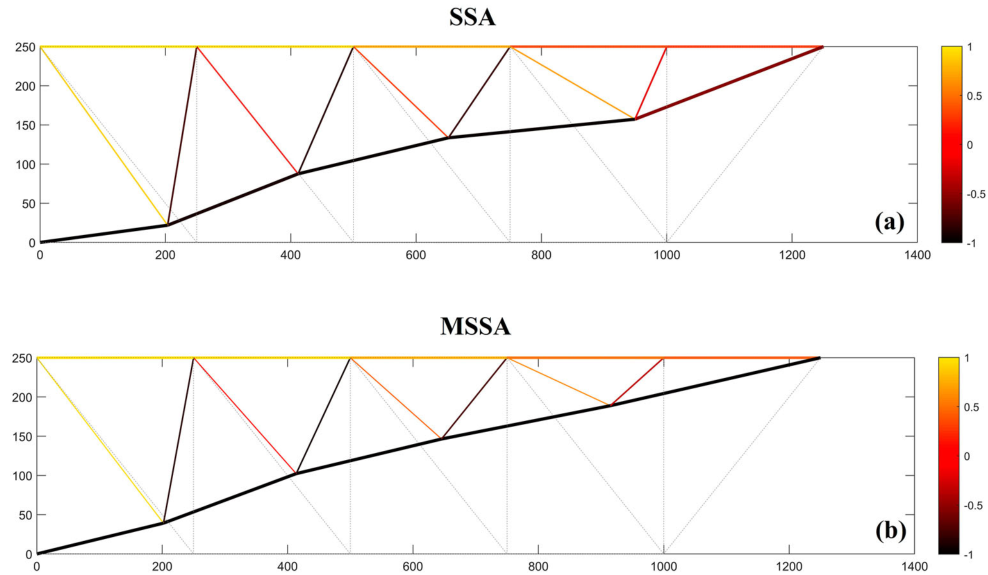

Figure 10. Optimum layout of the 18-bar truss: (a) salp swarm algorithm (SSA), (b) modified salp swarm algorithm (MSSA).

Table 2. Optimum solution details for the 18-bar truss example.

SA	IEVPS	FM-GA	ABC	MBRCGA
4533.240	4525.090	4530.700	4537.064	4520.200
SCPSO	CPSO	iPSO	SSA	MSSA
4512.365	4561.131	4520.990	4750.950	4565.310

Note: SA = simulated annealing, Hasancebi and Erbatur (2002); IEVPS = improved enhanced vibrating particles system, Kaveh and Khosravian (2022); FM-GA = force method and genetic algorithm, Rahami, Kaveh, and Gholipour (2008); ABC = artificial bee colony, Jawad *et al*. (2021); MBRCGA = mutation-based real-coded genetic algorithm, Kazemzadeh Azad and Jayant Kulkarni (2012); SCPSO = sequential cellular particle swarm optimization, Gholizadeh (2013); CPSO = cellular particle swarm optimization, Gholizadeh (2013); iPSO = integrated particle swarm optimizer, Mortazavi, Toğan, and Nuhoğlu (2017); SSA = salp swarm algorithm; MSSA = modified salp swarm algorithm.

Table 2 shows that the SSA and MSSA performed, respectively, about 5% and 1% worse (heavier) than one of the best-yielding methods in the literature, sequential cellular particle swarm optimization (SCPSO). The SSA and MSSA box-plot diagrams indicate that the range of optimum solutions for both the classical SSA and the MSSA have similar consistency behaviours. Figure 10 illustrates the optimum layout of the 18-bar truss for the SSA and MSSA. Demand/capacity (D/C) ratios are shown according to the elements' allowable stresses, and the elements' cross-sectional areas are illustrated through line thickness. In addition, the initial layout is shown as dashed lines.

3.3. A 47-bar truss

Table 3 shows the lowest structure weight acquired during 30 analyses. To the best of the authors' knowledge, the MSSA achieved the lowest result (2107.245 lb) among the reference studies presented. While SSA developed a more optimal (lower) solution (2218.483 lb) than harmony search (HS) (2396.8 lb), colliding bodies optimization (CBO) (2386.0 lb) and the discrete advanced Jaya algorithm (DAJA) (2376.019 lb), it performed worse than the switching teams algorithm (STA) approach (2172.49 lb). The difference between the results of the SSA and MSSA methods is about

Table 3. Optimum solution details for the 47-bar truss example (weight in lb).

HS	CBO	STA	DAJA	SSA	MSSA
2396.800	2386.000	2172.490	2376.019	2218.483	2107.245

Note: HS = harmony search, Lee *et al*. (2005); CBO = colliding bodies optimization, Kaveh and Mahdavi (2014); STA = switching teams algorithm, Shahrouzi (2020); DAJA = discrete advanced Jaya algorithm, Degertekin, Lamberti, and Ugur (2019); SSA = salp swarm algorithm; MSSA = modified salp swarm algorithm.

5%. On the other hand, the difference between the MSSA method and the method (STA) that performs closest to it is 3%. As in the first example, statistically, the average value of the MSSA is much lower than the SSA, and the standard deviation value is also lower than the SSA. In addition, the standard deviation of the SSA is significantly greater than the reference values.

As illustrated in the supplemental material, the MSSA has an exceptionally smooth convergence curve and achieves the optimal solution without encountering any issues during the optimization process. In contrast, the SSA has significant convergence problems; notably, the first 30–40% of the optimization process has difficulty developing optimal results, but it can make significant progress after 40%, and optimization convergence appears to be approaching a better solution after 60–70%.

As in the other representative examples, based on box-plot diagrams of the SSA and MSSA, it is observed that the range of optimum solutions obtained by the classical SSA is broader than that obtained by the MSSA. In addition, the MSSA provides extremely close optimum solutions for each cross-section within a relatively narrow range, indicating that the MSSA performs more reliably than the classical SSA. However, unlike the 10-bar truss, the difference between the average and best solution intervals and the average and worst solution intervals for the SSA was more uneven. This indicates that the solution has the potential to generate extremely poor solutions in the interval, reducing the reliability of the SSA approach. On the other hand, the MSSA's average, best and worst solutions are gathered in a balanced and very narrow range, which makes the MSSA a much more reliable and stable method than the SSA.

3.4. A 52-bar truss

The MSSA approach obtained the best (lowest) solution; the SSA method also performed well, outperforming seven of the eight reference solutions taken from the literature, with a difference of around 0.033% more than (lower performance) the hybrid particle swarm optimization and genetic algorithm (HPSOGA) method (Table 4). The difference between the two strategies (SSA and MSSA) is around 0.12%, and both methods perform exceptionally well in general. As illustrated in the figures of the supplemental material, convergence curves and trends for the current 52-bar truss problem reveal that the MSSA exhibits superior convergence behaviour to SSA. Initial solutions of the SSA have trouble advancing, resulting in a low convergence rate. Box-plot diagrams of the SSA and MSSA show that the range of optimum solutions obtained by the classical SSA is broader than that obtained by the MSSA, and that the MSSA provides extremely close optimum solutions for each cross-section within a relatively narrow range. This demonstrates that the MSSA performs more reliably than the classical SSA. In addition, as in the 42-bar truss problem, the difference between the average and best solution intervals and the average and worst solution intervals for the SSA is uneven. This indicates that the SSA has the potential to generate extremely poor solutions in the interval. In contrast, the MSSA's average, best and worst solutions are gathered in a balanced and very narrow range, making the MSSA a much more reliable and stable method than the SSA, and demonstrating good performance.

3.5. A 200-bar truss

Although the findings for the MSSA were comparable to the reference values, it was observed that the results for the SSA were significantly far away from reaching the optimum results (Table 5) (Lee

Table 4. Optimum solution details for the 52-bar truss example (weight in kg).

HPSO	IMBA	AEDE	NMA	GA
1905.490	1902.605	1902.605	1902.605	1970.142
HS	DHPSACO	HPSOGA	SSA	MSSA
1906.760	1904.830	1901.350	1901.974	1899.797

Note: HPSO = heuristic particle swarm optimization, Li, Huang, and Liu (2009); IMBA = improved mine blast algorithm, Sadollah *et al*. (2015); AEDE = adaptive elitist differential evolution, Ho-Huu *et al*. (2016); NMA = Newton meta-heuristic algorithm, Danesh and Jalilkhani (2020); GA = genetic algorithm, Wu and Chow (1995); HS = harmony search, Lee *et al*. (2005); DHPSACO = discrete heuristic particle swarm ant colony optimization, Kaveh and Talatahari (2009); HPSOGA = hybrid particle swarm optimization and genetic algorithm, Omidinasab and Goodarzimehr (2020); SSA = salp swarm algorithm; MSSA = modified salp swarm algorithm.

Table 5. Optimum solution details for the 200-bar truss example (weight in lb).

HS	SA	TLBO	CNNT-PSO	MCOA	SSA	MSSA
25,447.100	25,445.630	25,488.150	25,453.096	25,450.180	45,184.074	26,059.074

Note: HS = harmony search, Lee and Geem (2004); SA = simulated annealing, Lamberti (2008); TLBO = teaching–learning-based optimization, Degertekin and Hayalioglu (2013); CNNT-PSO = cyclic neighbourhood network topology particle swarm optimizer, Kim and Byun (2020); MCOA = modified coyote optimization algorithm, Pierezan *et al*. (2021); SSA = salp swarm algorithm; MSSA = modified salp swarm algorithm.

and Geem 2004; Lamberti 2008; Kim and Byun 2020; Pierezan *et al*. 2021; Degertekin and Hayalioglu 2013). According to the figures in the supplemental material, although the SSA has a decent convergence curve, the MSSA appears superior. The MSSA's solution interval is narrower than the SSA's solution interval, as seen in the box-plot diagrams. In addition, the MSSA's mean and standard deviation values are smaller than those for the SSA. Moreover, although the SSA appears to be more stable than the prior solutions, it was noticed that the solution intervals are larger, and the values obtained for the solution interval (average, maximum and minimum values) are nearly twice as large as those obtained for the MSSA.

3.6. Determination of search parameters

One of the biggest advantages of metaheuristic methods is the ease of adaptation to problems of several types and sizes. However, the search parameters of metaheuristic techniques need to be calibrated to enable them to be adapted to different problems. Since the SSA and MSSA have not been adapted to structural optimization problems in the literature, sensitivity analysis was performed, and the most suitable method parameters were determined in this study. In this context, all samples in the study were analysed according to different search agent values. Since all parameters are generated randomly in the formulae, the number of search agents is taken as the only variable for each benchmark problem. The minimum and standard deviation values were determined by performing 1000 iteration analyses 30 times, beginning with 10, the number of salps, and ending with the value 100 (including 100) at intervals of 10, to identify the optimal search agent values. A detailed table of the search parameters is provided in the supplemental material, with the minimum values (the best results) determined by these investigations highlighted in bold.

4. Discussion

The findings in Section 3 are discussed in terms of the differences in continuous and discrete problems regarding the best results for weight, average, standard deviation and convergence performance of the two optimization methods, the SSA and the MSSA.

The SSA and MSSA have been shown to perform better in discrete problems than in continuous problems. In both discrete problems (47-bar and 52-bar trusses), MSSA demonstrated superior performance and found the best solutions. The differences between the literature results and the MSSA's optimal solution are 3–12% and 0.08–3.6% for the 47-bar and 52-bar trusses, respectively. Although the SSA could not find the optimal solution, it obtained the third-best solution for discrete problems.

Even though the MSSA cannot produce the best solution for continuous problems, the differences between the best solutions are 0.05% and 2.36% for the 10-bar and 200-bar trusses, respectively. In contrast, the SSA did not perform well in continuous problems, and it found 0.37% and 47.78% heavier designs for the 10-bar and 200-bar trusses, respectively.

As the problem sizes increase, the decrease in the performance of the SSA becomes much more evident, and the more significant the difference between the SSA and MSSA.

It is seen that the SSA performs in a more dispersed manner, has a broader solution range, has higher standard deviation values and has a more unreliable structure in all solutions compared to the MSSA. In contrast, the MSSA seems to produce a more stable, more balanced and lower solution range than the SSA.

The convergence trends for all benchmark problems are defined by power functions, proving mathematically that the power coefficient (absolute value) for the MSSA is greater than that for the SSA, indicating a higher convergence rate.

The 18-bar truss is adopted as the second benchmark problem to assess whether the SSA and the modified methodology (MSSA) can simultaneously address continuous and discrete design variables and optimize the size and shape of truss structures subject to service loads. The MSSA performs comparably to one of the best solutions in the literature (SCPSO), but about 1% worse (heavier). In contrast, the SSA yields 5% worse than the SCPSO method. In addition, the MSSA obtains the best solution after 30% of the optimization process, whereas the standard SSA does so just before 90%.

5. Conclusions

The recently developed SSA and MSSA methods are evaluated on five different benchmark problems. The significant findings of this article are listed as follows:

(1) The SSA method performs poorly, particularly in continuous problems (10-bar and 200-bar trusses), and produces results that are inferior to those obtained in benchmark problems, whereas the MSSA method yields solutions that are superior to those obtained by the SSA and close to the best results presented in the literature, used as reference.
(2) Both methods produce acceptable results in discrete problems (47-bar and 52-bar trusses), although the MSSA outperforms the SSA.
(3) The MSSA shows a reasonable improvement in convergence rate, as seen in the convergence curves, whereas the SSA has significant convergence issues in discrete problems. This is due to the previously specified parameter (c_1), and the MSSA was developed because of changes in the parameter. Therefore, the beneficial effect of the changed parameter on the convergence rate of the MSSA was shown. Thus, the MSSA achieves a significantly higher rate of convergence when solving all benchmark problems.
(4) As the problem size increases, especially in continuous problems, the performance of the SSA decreases, whereas better and more stable solutions are obtained in the MSSA. In addition, the difference between the performance of the SSA and MSSA increases as the problem size increases. Statistically, the MSSA exhibits more consistent and more stable behaviours than the SSA, resulting in the MSSA being a safer approach.
(5) The SSA and MSSA are also tested on a problem that has both discrete and continuous variables (18-bar truss). The performance of the MSSA is remarkably close to the best results in the references. Like in continuous problems, the best solution of the SSA is much worse than the reference results.

In summary, it is demonstrated that the modifications to the MSSA yield outstanding results in terms of solution performance, convergence speed and solution security; it is also demonstrated that the MSSA is a robust and efficient technique that is effectively applicable to several similar problems. There may be issues in the performance of algorithms such as the MSSA in real-size problems, which could be considered in future studies. Because of certain drawbacks experienced in their solutions, optimization studies to improve the performance of the SSA and MSSA by incorporating well-known mathematical techniques such as Lévy flight and chaos theories may be considered in future studies.

Disclosure statement

No potential conflict of interest was reported by the authors.

Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Onur Altay http://orcid.org/0000-0003-0776-3580
Oguzhan Cetindemir http://orcid.org/0000-0002-1018-4515
Ibrahim Aydogdu http://orcid.org/0000-0002-8281-2365

References

Abualigah, Laith, Mohammad Shehab, Mohammad Alshinwan, and Hamzeh Alabool. 2020. "Salp Swarm Algorithm: A Comprehensive Survey." *Neural Computing and Applications* 32 (15): 11195–11215. doi:10.1007/s00521-019-04629-4

Altay, Onur, Oğuzhan Çetindemir, and İbrahim Aydoğdu. 2022. "Size Optimization of Planar Truss Systems Using the Modified Salp-Swarm Algorithm." *Figshare: Online Resource*, 26. doi:10.6084/m9.figshare.21280083.v1.

Anderson, P. A. V., and Quentin Bone. 1980. "Communication Between Individuals in Salp Chains. II. Physiology." *Proceedings of the Royal Society of London. Series B. Biological Sciences* 210 (1181): 559–574.

Anh, Phạm Hoàng. 2016. "Optimal Truss Sizing with Stress and Displacement Constraints." *Tạp chí Khoa học Công nghệ Xây dựng (KHCNXD)-ĐHXDHN* 10 (2): 69–77.

Asl, Reza Najian, Mohamad Aslani, and Masoud Shariat Panahi. 2013. "Sizing Optimization of Truss Structures Using a Hybridized Genetic Algorithm." *arXiv preprint arXiv:1306.1454*.

Assimi, Hirad, Ali Jamali, and Nader Nariman-Zadeh. 2017. "Sizing and Topology Optimization of Truss Structures Using Genetic Programming." *Swarm and Evolutionary Computation* 37: 90–103. doi:10.1016/j.swevo.2017.05.009

Aydogdu, Ibrahim, Serdar Carbas, and Alper Akin. 2017. "Effect of Levy Flight on the Discrete Optimum Design of Steel Skeletal Structures Using Metaheuristics." *Steel and Composite Structures, An International Journal* 24 (1): 93–112. doi:10.12989/scs.2017.24.1.093

Aydogdu, Ibrahim, Tevfik Oguz Ormecioglu, and Serdar Carbas. 2021. "Electrostatic Discharge Algorithm for Optimum Design of Real-Size Truss Structures." In *Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications*, edited by Serdar Carbas, Abdurrahim Toktas, and Deniz Ustun, 93–109. Singapore: Springer.

Baghlani, A., and M. H. Makiabadi. 2014. "Weight Optimization of Truss Structures by a New Feasible Boundary Search Technique Hybridized with Firefly Algorithm." *KSCE Journal of Civil Engineering* 18 (4): 1105–1118. doi:10.1007/s12205-014-0438-x

Baghlani, Abdolhossein, M. H. Makiabadi, and M. R. Maheri. 2017. "Sizing Optimization of Truss Structures by an Efficient Constraint-Handling Strategy in TLBO." *Journal of Computing in Civil Engineering* 31 (4): 04017004. doi:10.1061/(ASCE)CP.1943-5487.0000642

Bennage, W. A., and A. K. Dhingra. 1995. "Optimization of Truss Topology Using Tabu Search." *International Journal for Numerical Methods in Engineering* 38 (23): 4035–4052. doi:10.1002/nme.1620382308

Bureerat, Sujin, and Nantiwat Pholdee. 2016. "Optimal Truss Sizing Using an Adaptive Differential Evolution Algorithm." *Journal of Computing in Civil Engineering* 30 (2): 04015019. doi:10.1061/(ASCE)CP.1943-5487.0000487

Cuong-Le, Thanh, Hoang-Le Minh, Samir Khatir, Magd Abdel Wahab, Minh Thi Tran, and Seyedali Mirjalili. 2021. "A Novel Version of Cuckoo Search Algorithm for Solving Optimization Problems." *Expert Systems with Applications* 186: 115669. doi:10.1016/j.eswa.2021.115669

Danesh, M., and M. Jalilkhani. 2020. "SIzing Optimization of Truss Structures with Newton Meta-heuristic Algorithm." *Iran University of Science & Technology* 10 (3): 433–450.

Degertekin, S. O. 2012. "Improved Harmony Search Algorithms for Sizing Optimization of Truss Structures." *Computers & Structures* 92: 229–241. doi:10.1016/j.compstruc.2011.10.022

Degertekin, S. O., and M. S. Hayalioglu. 2013. "Sizing Truss Structures Using Teaching-Learning-Based Optimization." *Computers & Structures* 119: 177–188. doi:10.1016/j.compstruc.2012.12.011

Degertekin, S. O., L. Lamberti, and I. B. Ugur. 2019. "Discrete Sizing/Layout/Topology Optimization of Truss Structures with an Advanced Jaya Algorithm." *Applied Soft Computing* 79: 363–390. doi:10.1016/j.asoc.2019.03.058

Galante, Miguel. 1996. "Genetic Algorithms as an Approach to Optimize Real-World Trusses." *International Journal for Numerical Methods in Engineering* 39 (3): 361–382. doi:10.1002/(SICI)1097-0207(19960215)39:3 < 361::AID-NME854 > 3.0.CO;2-1

Gandomi, Amir H., and Amir H. Alavi. 2016. "An Introduction of Krill Herd Algorithm for Engineering Optimization." *Journal of Civil Engineering and Management* 22 (3): 302–310. doi:10.3846/13923730.2014.897986

Gandomi, Amir Hossein, Siamak Talatahari, Xin-She Yang, and Suash Deb. 2013. "Design Optimization of Truss Structures Using Cuckoo Search Algorithm." *The Structural Design of Tall and Special Buildings* 22 (17): 1330–1349. doi:10.1002/tal.1033

Gholizadeh, Saeed. 2013. "Layout Optimization of Truss Structures by Hybridizing Cellular Automata and Particle Swarm Optimization." *Computers & Structures* 125: 86–99. doi:10.1016/j.compstruc.2013.04.024

Gomes, Herbert Martins. 2011. "Truss Optimization with Dynamic Constraints Using a Particle Swarm Algorithm." *Expert Systems with Applications* 38 (1): 957–968. doi:10.1016/j.eswa.2010.07.086

Gonçalves, Matheus Silva, Rafael Holdorf Lopez, and Leandro Fleck Fadel Miguel. 2015. "Search Group Algorithm: A New Metaheuristic Method for the Optimization of Truss Structures." *Computers & Structures* 153: 165–184. doi:10.1016/j.compstruc.2015.03.003.

Hasancebi, Oğuzhan, and F. Erbatur. 2002. "On Efficient Use of Simulated Annealing in Complex Structural Optimization Problems." *Acta Mechanica* 157 (1): 27–50. doi:10.1007/BF01182153

Ho-Huu, V., T. Nguyen-Thoi, T. Vo-Duy, and T. Nguyen-Trang. 2016. "An Adaptive Elitist Differential Evolution for Optimization of Truss Structures with Discrete Design Variables." *Computers & Structures* 165: 59–75. doi:10.1016/j.compstruc.2015.11.014

Jalili, Shahin, and Yousef Hosseinzadeh. 2018. "Design Optimization of Truss Structures with Continuous and Discrete Variables by Hybrid of Biogeography-Based Optimization and Differential Evolution Methods." *The Structural Design of Tall and Special Buildings* 27 (14): e1495. doi:10.1002/tal.1495

Jalili, Shahin, Yousef Hosseinzadeh, and Nasser Taghizadieh. 2016. "A Biogeography-Based Optimization for Optimum Discrete Design of Skeletal Structures." *Engineering Optimization* 48 (9): 1491–1514. doi:10.1080/0305215X.2015.1115028

Jawad, Farqad K. J., Celal Ozturk, Wang Dansheng, Mohammed Mahmood, Osama Al-Azzawi, and Anas Al-Jemely. 2021. "Sizing and Layout Optimization of Truss Structures with Artificial Bee Colony Algorithm." Structures 30: 546–559.

Kaveh, A., and S. M. Javadi. 2014. "Shape and Size Optimization of Trusses with Multiple Frequency Constraints Using Harmony Search and Ray Optimizer for Enhancing the Particle Swarm Optimization Algorithm." *Acta Mechanica* 225 (6): 1595–1605. doi:10.1007/s00707-013-1006-z

Kaveh, Ali, and Masoud Khosravian. 2022. "Size/Layout Optimization of Truss Structures Using Vibrating Particles System Meta-heuristic Algorithm and Its Improved Version." *Periodica Polytechnica Civil Engineering* 66 (1): 1–17.

Kaveh, A., and V. R. Mahdavi. 2014. "Colliding Bodies Optimization Method for Optimum Discrete Design of Truss Structures." *Computers & Structures* 139: 43–53. doi:10.1016/j.compstruc.2014.04.006

Kaveh, A., and S. Talatahari. 2009. "A Particle Swarm Ant Colony Optimization for Truss Structures with Discrete Variables." *Journal of Constructional Steel Research* 65 (8-9): 1558–1568. doi:10.1016/j.jcsr.2009.04.021

Kazemzadeh Azad, S., and A. Jayant Kulkarni. 2012. "Structural Optimization Using a Mutation-Based Genetic Algorithm." *Iran University of Science & Technology* 2 (1): 81–101.

Kim, Tae-Hyoung, and Jung-In Byun. 2020. "Truss Sizing Optimization with a Diversity-Enhanced Cyclic Neighborhood Network Topology Particle Swarm Optimizer." *Mathematics* 8 (7): 1087. doi:10.3390/math8071087

Lamberti, L. 2008. "An Efficient Simulated Annealing Algorithm for Design Optimization of Truss Structures." *Computers & Structures* 86 (19-20): 1936–1953. doi:10.1016/j.compstruc.2008.02.004

Lee, Kang Seok, and Zong Woo Geem. 2004. "A New Structural Optimization Method Based on The Harmony Search Algorithm." *Computers & Structures* 82 (9-10): 781–798. doi:10.1016/j.compstruc.2004.01.002

Lee, Kang Seok, Zong Woo Geem, Sang-ho Lee, and Kyu-woong Bae. 2005. "The Harmony Search Heuristic Algorithm for Discrete Structural Optimization." *Engineering Optimization* 37 (7): 663–684. doi:10.1080/03052150500211895.

Li, L. J., Z. B. Huang, and F. Liu. 2009. "A Heuristic Particle Swarm Optimization Method for Truss Structures with Discrete Variables." *Computers & Structures* 87 (7-8): 435–443. doi:10.1016/j.compstruc.2009.01.004

Li, Li-Juan, Z. B. Huang, Feng Liu, and Q. H. Wu. 2007. "A Heuristic Particle Swarm Optimizer for Optimization of Pin Connected Structures." *Computers & Structures* 85 (7-8): 340–349. doi:10.1016/j.compstruc.2006.11.020

Li, Yancang, and Sida Lian. 2018. "Improved Fruit Fly Optimization Algorithm Incorporating Tabu Search for Optimizing the Selection of Elements in Trusses." *KSCE Journal of Civil Engineering* 22 (12): 4940–4954. doi:10.1007/s12205-017-2000-0

Lingyun, Wei, Zhao Mei, Wu Guangming, and Meng Guang. 2005. "Truss Optimization on Shape and Sizing with Frequency Constraints Based On Genetic Algorithm." *Computational Mechanics* 35 (5): 361–368. doi:10.1007/s00466-004-0623-8

Madin, Larry P. 1990. "Aspects of Jet Propulsion in Salps." *Canadian Journal of Zoology* 68 (4): 765–777. doi:10.1139/z90-111

McCauley, Douglas J., Malin L. Pinsky, Stephen R. Palumbi, James A. Estes, Francis H. Joyce, and Robert R. Warner. 2015. "Marine Defaunation: Animal Loss in The Global Ocean." *Science* 347 (6219): 1255641. doi:10.1126/science.1255641

Miguel, Letícia Fleck Fadel, and Leandro Fleck Fadel Miguel. 2012. "Shape and Size Optimization of Truss Structures Considering Dynamic Constraints Through Modern Metaheuristic Algorithms." *Expert Systems with Applications* 39 (10): 9458–9467. doi:10.1016/j.eswa.2012.02.113

Millan-Paramo, Carlos, and João Elias Abdalla Filho. 2020. "Size and Shape Optimization of Truss Structures with Natural Frequency Constraints Using Modified Simulated Annealing Algorithm." *Arabian Journal for Science and Engineering* 45 (5): 3511–3525. doi:10.1007/s13369-019-04138-5

Mirjalili, Seyedali, Amir H. Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. 2017. "Salp Swarm Algorithm: A Bio-inspired Optimizer for Engineering Design Problems." *Advances in Engineering Software* 114: 163–191. doi:10.1016/j.advengsoft.2017.07.002

Mortazavi, Ali, Vedat Toğan, and Ayhan Nuhoğlu. 2017. "Weight Minimization of Truss Structures with Sizing and Layout Variables Using Integrated Particle Swarm Optimizer." *Journal of Civil Engineering and Management* 23 (8): 985–1001. doi:10.3846/13923730.2017.1348982

Omidinasab, Fereydoon, and Vahid Goodarzimehr. 2020. "A Hybrid Particle Swarm Optimization and Genetic Algorithm for Truss Structures with Discrete Variables." *Journal of Applied and Computational Mechanics* 6 (3): 593–604.

Perez, Ruben E., and Kamran Behdinan. 2007. "Particle Swarm Approach for Structural Design Optimization." *Computers & Structures* 85 (19-20): 1579–1588. doi:10.1016/j.compstruc.2006.10.013

Pierezan, Juliano, Leandro dos Santos Coelho, Viviana Cocco Mariani, Emerson Hochsteiner de Vasconcelos Segundo, and Doddy Prayogo. 2021. "Chaotic Coyote Algorithm Applied to Truss Optimization Problems." *Computers & Structures* 242: 106353. doi:10.1016/j.compstruc.2020.106353

Rahami, H., A. Kaveh, and Y. Gholipour. 2008. "Sizing, Geometry and Topology Optimization of Trusses Via Force Method and Genetic Algorithm." *Engineering Structures* 30 (9): 2360–2369. doi:10.1016/j.engstruct.2008.01.012

Rajeev, S., and C. S. Krishnamoorthy. 1997. "Genetic Algorithms-Based Methodologies for Design Optimization of Trusses." *Journal of Structural Engineering* 123 (3): 350–358. doi:10.1061/(ASCE)0733-9445(1997)123:3(350)

Sadollah, Ali, Hadi Eskandar, Ardeshir Bahreininejad, and Joong Hoon Kim. 2015. "Water Cycle, Mine Blast and Improved Mine Blast Algorithms for Discrete Sizing Optimization of Truss Structures." *Computers & Structures* 149: 1–16. doi:10.1016/j.compstruc.2014.12.003

Schmit Jr., L. A., and B. Farshi. 1974. "Some Approximation Concepts for Structural Synthesis." *AIAA Journal* 12 (5): 692–699. doi:10.2514/3.49321

Schmit Jr., Lucien A., and Hirokazu Miura. 1976. "Approximation Concepts for Efficient Structural Synthesis." *NASA*.

Shahrouzi, M. 2020. "Switching Teams Algorithm for Sizing Optimization of Truss Structures." *International Journal of Optimization in Civil Engineering* 10 (3): 365–389.

Sonmez, Mustafa. 2011. "Discrete Optimum Design of Truss Structures Using Artificial Bee Colony Algorithm." *Structural and Multidisciplinary Optimization* 43 (1): 85–97. doi:10.1007/s00158-010-0551-5.

Venkayya, V. B. 1971. "Design of Optimum Structures." *Computers & Structures* 1 (1-2): 265–309. doi:10.1016/0045-7949(71)90013-7

Wu, Shyue-Jian, and Pei-Tse Chow. 1995. "Steady-State Genetic Algorithms for Discrete Optimization of Trusses." *Computers & Structures* 56 (6): 979–991. doi:10.1016/0045-7949(94)00551-D

Wu, Chun-Yin, and Ko-Ying Tseng. 2010. "Truss Structure Optimization Using Adaptive Multi-population Differential Evolution." *Structural and Multidisciplinary Optimization* 42 (4): 575–590. doi:10.1007/s00158-010-0507-9

Xu, Tao, Wenjie Zuo, Tianshuang Xu, Guangcai Song, and Ruichuan Li. 2010. "An Adaptive Reanalysis Method for Genetic Algorithm with Application to Fast Truss Optimization." *Acta Mechanica Sinica* 26 (2): 225–234. doi:10.1007/s10409-009-0323-x

Yang, Yaowen, and Chee Kiong Soh. 2002. "Automated Optimum Design of Structures Using Genetic Programming." *Computers & Structures* 80 (18-19): 1537–1546. doi:10.1016/S0045-7949(02)00108-6