SSMA: simplified slime mould algorithm for optimization wireless sensor network coverage problem

Yuanye Wei\textsuperscript{a}, Xiuxi Wei\textsuperscript{b}, Huajuan Huang\textsuperscript{a, b}, Jian Bi\textsuperscript{a, b}, Yongquan Zhou\textsuperscript{a, b} and Yanlian Du\textsuperscript{c, d}

\textsuperscript{a}College of Artificial Intelligence, Guangxi University for Nationalities, Nanning, People’s Republic of China; \textsuperscript{b}Guangxi Key Laboratories of Hybrid Computation and IC Design Analysis, Nanning, People’s Republic of China; \textsuperscript{c}College of Information and Communication Engineering, Hainan University, Haikou, People’s Republic of China; \textsuperscript{d}State Key Laboratory of Marine Resources Utilization in South China Sea, Hainan University, Haikou, People’s Republic of China

\begin{abstract}
Wireless sensor network (WSN) coverage problem is to think about how to maximize the network coverage to obtain reliable monitoring and tracking services with guaranteed quality of service. In this paper, a simplified slime mould algorithm (SSMA) for solving the WSN coverage problem is proposed. In SSMA, we mainly conducted 13 groups of WSNs coverage optimization experiments and compared them with six well-known meta-heuristic optimization algorithms. The experimental results and Wilcoxon rank-sum test show that the proposed SSMA is generally competitive, outstanding performance and effectiveness. We proposed SSMA algorithm could be helpful to effectively control the network nodes energy, improve the perceived quality of services and extend the network survival time.
\end{abstract}

\section{1. Introduction}

In general, the wireless sensor network (WSN) is composed of a large number of densely distributed sensor networks node, each node has limited computing, storage and wireless communication capabilities and can sense the surrounding environment at close range. It has the characteristics of small size, low cost and low power consumption. It can assist in sensing, collecting and processing the information of the monitored object in real time, so with the rapid development of science and technology in various fields, WSNs have been widely used in military affairs, environmental monitoring, medical and health care, safety monitoring and other fields (Singh et al., 2021). In various applications of WSNs, coverage has always been a crucial issue, which determines the monitoring ability of the target area. An appropriate node deployment strategy can not only improve the service quality of WSNs but also effectively promote its energy utilization. Generally, sensor nodes are randomly scattered in the workspace. However, this method will lead to a high concentration of nodes, which will lead to a low coverage rate and affect the monitoring quality (Zhang & Fok, 2017). Therefore, it is necessary to adaptively adjust and deploy the sensor nodes in WSNs, so that they are distributed more evenly in the detection area and have an effective coverage rate, so as to rationally allocate network space resources and better accomplish the tasks of environment awareness and information acquisition, which is of great significance to improve the network survivability, network reliability and network construction cost. Searching for the optimal node deployment scheme is a difficult task, especially for large-scale sensor networks (Adulyasas et al., 2015; Wang et al., 2010; Zhao et al., 2004). In this context, most scholars choose to focus on meta-heuristic algorithms (MAs), Because of its advantages of availability and simple concept and few parameters, some scholars have proposed new algorithms and used them to solve practical application problems in recent years. Such as, Deng et al. (2022) proposed an enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. Song et al. (2021) proposed a new algorithm MPPCEDE: multi-population parallel co-evolutionary differential evolution (DE) for parameter optimization; Deng et al. (2021) use the DE with wavelet basis function and optimal mutation strategy for complex optimization problems; Zhang et al. (2016) proposed a new Grey wolf optimizer for unmanned combat aerial vehicle path planning. Zhou et al. (2018) proposed a sensor deployment scheme based on a social spider optimization algorithm for WSNs.

\begin{articlehistory}
Received 13 December 2021
Accepted 28 May 2022
\end{articlehistory}

\begin{keywords}
Wireless sensor network (WSN); simplified slime mould algorithm (SSMA); effective coverage rate; meta-heuristic optimization algorithm
\end{keywords}

\section*{CONTACT} Yongquan Zhou yongquanzhou@126.com

© 2022 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
MAs have made a good contribution to the coverage optimization of WSNs. Mendis et al. (2006) used PSO optimized sink node path location in WSNs; a new method is proposed to deduce the optimal path of sink nodes in fixed sensor node networks with practical difficulties such as mobility constraints. The algorithm achieves ideal results, but it mainly solves the experiments of wireless sensor nodes with fixed dimensions, and the number of experiments is small. Song et al. (2018) propose a novel change step of the FOA algorithm was introduced to the coverage optimization, at the same time, the mathematical modelling of two network models is carried out respectively. In the experiment part, the number of experiments is less and the comparison algorithm is less. You can increase the number of experiments and comparison algorithms. Aziz et al. (2009) used PSO and Voronoi diagrams to optimize sensor coverage in WSNs. PSO is used to find the optimal deployment of the sensors that gives the best coverage while the Voronoi diagram is used to evaluate the fitness of the solution. But the comparison algorithm and the number of experiments are less. Kuila and Jana (2014) propose a novel DE-based clustering algorithm for WSNs to prolong the lifetime of the network by preventing faster death of the highly loaded CHs and incorporating a local improvement phase to the traditional DE for faster convergence and better performance of the proposed algorithm. Liao et al. (2011) used an ACO-based sensor deployment protocol for WSNs. Based on the ACO algorithm, the author proposed a deployment scheme to prolong the network lifetime, while ensuring complete coverage of the service region. The simulations show that the algorithm can prolong the lifetime of the network. Ambareesh and Madheswari (2021) proposed a hybrid red deer salp swarm algorithm for optimal routing protocol in wireless multimedia sensor networks. The work outlined in this paper is to minimize four different objectives namely packet loss, memory, delay and expected transmission cost. The main intention of the multi-objective function involves generating a diverse optimal solution set that is utilized to evaluate the trade-off among various objectives. Rajeswari et al. (2021) Powell Flower Pollination Algorithm (PFPA) propose to solve the MEB problem in WSNs. The proposed algorithm is compared with other heuristic approaches and the performance of the algorithm is assessed using benchmark instances with 50 and 100 nodes. Pakdel and Fotohi (2021) used FA for power management in WSNs. In this paper, a method is presented using the FA and using the four criteria of residual energy, noise rate, number of hops and distance. The proposed method called EM-FIREFLY is introduced which selects the best cluster head with high attractiveness and based on the fitness function and transfers the data packets through these cluster head to the sink. The proposed method is evaluated with an NS2 simulator and compared with the Algorithm-PSO and Optimal clustering methods. Karaboga et al. (2010) propose a novel hierarchical clustering approach for WSNs to maintain energy depletion of the network at a minimum using the Artificial Bee Colony Algorithm which is a new swarm-based heuristic algorithm. They present a protocol using Artificial Bee Colony Algorithm, which tries to provide optimum cluster organization in order to minimize energy consumption.

The slime mould algorithm (Li et al., 2020) is a novel meta-heuristic optimization algorithm, which was proposed by. It is to simulate the positive and negative feedback generated by the propagation wave of biological oscillator during the foraging process of Slime mould in nature and to guide their behaviour and morphological changes. In SMA, the oscillation factor in the process of foraging is very important, which reflects the perception of slime mould to the food concentration in the search space. And it will adjust their foraging patterns following the food concentration during foraging. Slime mould can obtain multiple information of food source, during searching food. When the food concentration is high, the wave source generated by the oscillator is relatively strong, and the feedback given by slime mould will approach the food source and surround it. Otherwise, when the food concentration is not perceived, Slime mould will randomly search in the search space. The algorithm uses the propagation wave of oscillator to simulate the feedback when sensing food concentration. Through this oscillation mode, feedback information can be given to other slime mould. Through this feedback information transmission mechanism, the foraging mode of slime mould is effectively simulated, and the superiority of the algorithm is improved. Because of its simple concept and strong searching ability, it has been paid attention to and studied by many scholars since it was put forward. Abdel-Basset et al. (2020) used SMAsWOA based on the thresholding technique to overcome ISP for COVID-19 chest X-ray images by integrating SMA and WOA to maximize the Kapur’s entropy; SMA was presented for the solar cell estimation (Kumar et al., 2020); Gao et al. (2020) proposed an improved SMA with cosine controlling parameters; Zhao and Gao (2020b) proposed a chaotic slime mould algorithm with Chebyshev map. Naik et al. (2022) used a leader slime mould algorithm (LSMA) introducing (NSD) based multilevel thresholding technique for multispectral images; Zhao and Gao (2020a) used hybridized the SMA and HHO algorithms for global optimization; Zhao et al. (2020) proposed an improved SMA; Agarwal and Bharti (2021) and Huang (1999) have implemented a modified NISI meta-heuristic approach known as a slime
mould optimization algorithm (SMOA) in this research for path planning and obstacle avoidance problem in mobile robots; Zubaidi et al. (2020) proposed hybrid ANN model with SMA for prediction of urban stochastic water demand; Mostafa et al. (2020) based on SMA proposed a new strategy to extract the optimal model parameters of solar PV panel; SMA for feature selection (Abdel-Basset et al., 2021); SMA-AGDE for solving various optimization problems (Houssein et al., 2021); mitigating the effects of magnetic coupling between HV transmission line and metallic pipeline using SMA (Djekidel et al., 2021), Wei et al. (2021) proposed an improved slime mould algorithm for optimal reactive power dispatch problem.

In this paper, to improve the search accuracy and population diversity of the con SMA in solving the coverage optimization problem of WSNs, we propose a simplified SMOA have four highlights as follows:

1. A simplified location updating formula is proposed, which improves the accuracy and speed of the algorithm in solving the coverage optimization problem of WSNs.
2. An improved adaptive oscillation factor is proposed to improve the exploration ability of SSMA in the early stage of search and the convergence speed of the algorithm.
3. SSMA main considers small, medium and large-scale wireless network sensor optimization problems.
4. Compare the performance of each algorithm in this optimization problem under different dimensions of small and medium scale.

The rest of this paper is structured as follows: Section 2 briefly introduces the WSN coverage problem. Section 3 introduces the SMA. The SSMA is proposed in Section 4. Section 5 carries out an experimental analysis and discussion of the SSMA for WSNs. Section 6 provides the conclusions and future work.

2. WSN coverage problem

Assumed that the monitoring area of WSNs is a two-dimensional plane, which is digitized into \( L \times M \) grids, and the size of each grid is set to unit 1. \( N \) homogeneous sensors are deployed in this area, and the node set can be expressed as \( Z = \{Z_1, Z_2, \ldots, Z_N\} \) having the same sensing radius \( R \). In this paper, the Boolean model is used as the node sensing model, so long as the target is within the node sensing range, it can be successfully sensed. Assuming that the coordinates of a node \( z_i \) in the detected area are \( (x_i, y_i) \) and the position coordinates \( D_j \) of the target point are \( (x_j, y_j) \), the distance between the node and the target point is

\[
d(d(z_i, D_j)) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2},
\]

where \( p(z_i, D_j) \) represents the perceptual quality of node \( z_i \) to \( D_j \). When the position \( D_j \) of node \( z_i \) is within the circle of perceived range, the perceived quality is 1, otherwise, it is 0, and the mathematical expression is

\[
p(z_i, D_j) = \begin{cases} 1 & \text{if } d(z_i, D_j) \leq R, \\ 0 & \text{otherwise}. \end{cases}
\]

Generally, the sensor’s perception probability of the target is less than 1. In order to improve the perception probability of the target, multiple sensors need to cooperate to detect, so the sensor’s perception probability of a certain target is

\[
p(Z, D_j) = 1 - \prod_{i=1}^{N} [1 - p(z_i, D_j)].
\]

The coverage rate of the node set \( Z \) over the entire monitoring area is

\[
f = \frac{1}{L \times M} \sum_{j=1}^{L \times M} p(Z, H_j).
\]

Equation (4) is the objective function of the WSN coverage optimization problem.

3. Slime mould algorithm

The slime mould algorithm (Li et al., 2020) mainly simulates the behaviour and morphological changes of slime mould in the foraging process in nature. The algorithm effectively simulates the information transmission mechanism of feeding back the food concentration in the search space through the propagation wave generated by the oscillator and simulates the threat to the existence in the foraging process with the weight factor, which improves the effectiveness of the algorithm.

Slime mould can judge the food source position according to the concentration of surrounding food during foraging, and adjust their foraging mode according to the influence of environment. It also can obtain multiple food sources and develop them at the same time. When obtaining information of multiple food sources, biological oscillators will generate different propagation waves according to the concentration of food sources. When the concentration is higher, the cytoplasm flow of veins connecting food will increase, the radius of veins will along with increase, and the channels will become more stable, thus forming the optimal path connecting food sources.
When slime mould obtains multiple food sources, they will adjust their foraging patterns according to food concentration, and when food concentration is high, slime mould will use local search; While the food concentration is low, Slime mould will use the multi-area search mode, more spontaneous to leave the low concentration area to search for food in another area.

3.1. Mathematical model

3.1.1. Search for food

When slime mould is foraging, they can sense the concentration and position of food through the smell in the air and then approach it constantly. This process can be expressed by the following equation:

\[
S(t + 1) = \begin{cases} 
S(t) + \frac{kb}{kc} \left( \frac{W \cdot S_{R1}(t) - S_{R2}(t)}{S(t)} \right), & r < q, \\
r \geq q,
\end{cases}
\]

(5)

where the parameter \( \overrightarrow{kb} \) is a number between \([-a, a]\) and \( \overrightarrow{kc} \) is linearly decreasing from 1 to 0, \( t \) represents the current time in iteration times, \( \overrightarrow{S} \) represents the current optimal individual, \( \overrightarrow{S} \) represents the current individual, \( S_{R1} \) and \( S_{R2} \) represent two individuals randomly selected in the slime mould population. \( W \) represents weight.

The \( q \) mathematical formula is as follows:

\[
q = \tanh |f(i) - DF|,
\]

(6)

where \( i \in 1, 2, \ldots, n, f(i) \) indicated fitness of individual \( \overrightarrow{S} \), \( DF \) represents the best fitness value obtained in all iterations.

The \( kb \) mathematical formula is as follows:

\[
kb = [-a, a],
\]

(7)

\[
a = \arctanh(-t/T_{max} + 1),
\]

(8)

\[
\overrightarrow{kc} = 1 - t/T_{max},
\]

(9)

where \( \overrightarrow{W} \) is the mathematical formula as follows:

\[
\overrightarrow{W}(smell(i)) = \begin{cases} 
1 + r \cdot \log \left( \frac{BF - f(i)}{BF - WF + e} + 1 \right), & i \leq \frac{N}{2}, \\
1 - r \cdot \log \left( \frac{BF - f(i)}{BF - WF + e} + 1 \right), & \text{other},
\end{cases}
\]

(10)

\[
smell = \text{sort}(f),
\]

(11)

where \( r \) is a random values interval of \([0, 1]\), \( T_{max} \) indicates the maximum number of iterations, \( BF \) indicates the best fitness value in the current iteration, \( WF \) indicates the worst fitness value in the current iteration, \( smell \) represents the corresponding values sorted by fitness values.

The search position of individual \( \overrightarrow{S} \) is updated by the best individual \( \overrightarrow{S}^* \) obtained at present and two random individuals \( \overrightarrow{S}_{R1} \) and \( \overrightarrow{S}_{R2} \) in the search space, and the position of individual \( \overrightarrow{S} \) is changed by fine tuning oscillation factor \( \overrightarrow{kb} \) and \( \overrightarrow{kc} \) weight factor \( \overrightarrow{W} \). Equation (5) simulates the uncertainty of venous contraction mode. Logarithmic log is used to reduce the numerical change rate reducing the rate of change of numerical value makes the numerical change of contraction frequency not too large, so that the searching individual can search in all possible directions near the optimal solution, thus simulating the fan-shaped structure of slime mould when it is close to food. This concept can also be applied to extend to hyper-dimensional space.

3.1.2. Wrapped food

In this section, we proposed a mathematical model for the second foraging mode of slime mould to wrapped food. In the process of approaching and then food encirclement, the organic matter in slime mould will secrete enzymes to digest food. If the concentration of food becomes higher, the cytoplasmic flow will also be greater, the greater the radius of the blood vessels, and therefore the greater the weight. On the contrary, when the food concentration in the current search area is low, the cytoplasmic flow becomes less, the vein radius becomes smaller, also the weight becomes smaller, and the slime mould will turn to other areas to search for food. According to the above principle, the second phase formula for updating the position of slime mould is as follows:

\[
\overrightarrow{S}(t + 1) = \begin{cases} 
\text{rand} \ast (ub - lb) + lb, & \text{rand} < pc, \\
\overrightarrow{S}(t) + \frac{kb}{kc} \left( \frac{W \cdot \overrightarrow{S}_{R1}(t) - \overrightarrow{S}_{R2}(t)}{\overrightarrow{S}(t)} \right), & r < q, \\
\frac{kb}{kc} \ast \overrightarrow{S}(t), & r \geq q,
\end{cases}
\]

(12)

where \( ub \) and \( lb \) represent the upper and lower boundaries of the search range, \( \text{rand} \) and \( r \) represent random values between \([0,1]\), and \( pc \) is a constant, which is taken as 0.03 in this paper.

3.1.3. Oscillation

During the foraging process, slime mould control the cytoplasmic flow on the vein network structure according to the propagation wave generated by its oscillator, and the width of veins can be adjusted by the flow. This mechanism can make slime mould find a better food concentration position. The algorithm has three variables: \( \overrightarrow{W} \), \( \overrightarrow{kb} \) and \( \overrightarrow{kc} \) which better simulates the foraging mode of slime mould under the action of oscillation mode. \( \overrightarrow{W} \) simulate the oscillation frequency of slime mould close to 1 under different food concentrations, which makes slime mould approach food more quickly when finding high-quality food and more slowly when the food concentration is lower in individual positions, thus improving
the efficiency of slime mould in selecting the best food source.

The value of $k_b$ oscillates randomly and approaches 0 with the increase of iteration times. The value of $k_c$ oscillates between $-1$ and $1$, and finally tends to 0. The synergistic effect between and simulated the selective behaviour of slime mould. In order to find a better food source, even if slime mould has found a better food source, it will still secrete some organic matter to explore other fields and try to find a higher quality food source instead of investing everything in one source.

In addition, the oscillation process of $k_b$ simulates the moving state of slime mould and decides whether to approach food sources or seek other food sources. At the same time, the process of exploring food is not smooth. During this period, there may be various obstacles, such as a light and dry environment, which limit the spread of slime mould. However, it also improves the possibility of slime moulds looking for higher quality food and avoids falling into local optimum. The pseudo-code of the SMA is shown in Algorithm 1.

### Algorithm 1. SMA pseudo-code

1. Initialize the parameter, population, $ub$ and $lb$, $Max\_iter$;
2. Initialize the position of slime mould $S_i(1, 2, \ldots, n)$;
3. While ($t < Max\_iter$)
4. Calculate the fitness of all slime mould;
5. Update $DF$, $BF$, $WF$, $S^*$;
6. For each search portion
7. Update $g$, $k_b$, $k_c$;
8. Update position
9. If ($rand < pc$),
10. Update position by Equation (12.1)
11. Else
12. If ($r < q$),
13. Update position by Equation (12.2)
14. Else
15. Update position by Equation (12.3)
16. End If
17. End If
18. End For
19. $t = t + 1$;
20. End While
21. Return $DF$, $S^*$

4. **Our proposed SSMA**

Slime mould algorithm (Li et al., 2020) is a newly swarm intelligence optimization algorithm, which is a stochastic optimizer to form the shortest path connecting food based on the oscillation mode generated by slime mould in nature. The algorithm has been concerned and studied by many scholars since it was put forward. The SMOA has the advantages of a simple concept and good results in solving various practical problems. However, like many meta-heuristic optimization algorithms have the disadvantages of premature convergence and easy to fall into local optimum. The stander SMA has obvious defects such as orientation local optimum, low coverage rate and insufficient convergence speed in solving the coverage optimization problem of WSNs. So, we propose an SSMA to solve the optimization problem of WSNs. The performance of the standard slime mould algorithm is not ideal when solving a coverage optimization in WSNs. The SSA for optimization problems mainly depends on the best leader (global best) and two randomly collected slime mould in the population, which leads to poor development and requires more iterations to converge this algorithm. Because slime mould will search in a limited area under the influence of food concentration in space, slime mould will search in the possible range around the best individual or the individual’s own range when the food concentration is high, and the probability of random search of slime mould is relatively small when the food concentration is not felt, so the probability of searching in a limited area becomes larger, which leads to premature convergence of stander SMA when solving some optimization problems.

It can be obviously from the above that due to the deficiency of the SMA updating formula, the algorithm cannot achieve good results in solving the coverage optimization problem of WSNs. The exploration and development of the algorithm are not well balanced, which leads to local optimum. In the experiment, we found that the optimized slime mould algorithm mainly depends on the best leader and two randomly collected slime mould in the population. Updating the formula leads to slime mould easily falling into local optimum, which leads to poor development, needs more iterations to converge, and leads to an imbalance between exploration and development. Therefore, we have further improved the updating and oscillation factor. The updated position (Equation (12)) is removed from the development stage of the improved algorithm. In the second part, the standard oscillation factor of SMA is changed into the adaptive oscillation factor of the cosine function, which improves the oscillation ability and convergence speed of the algorithm at the searching. The improvement of these two parts makes the exploration and development of SSMA get a good balance, which is also proved in the experimental part.

From the above analysis, a mathematical model of updating the formula can be obtained as follows:

$$
S^* (t+1) = \begin{cases} 
\text{rand} \times (ub - lb) + lb, & \text{rand} < pc, \\
S^* (t) + k_b \times (W - S_{R1}^*(t) - S_{R2}^*(t)), & \text{otherwise}, 
\end{cases}
$$

where $ub$ and $lb$ represent the upper and lower boundaries of the search range, $r$ and $rand$ represent random
values between, and \( pc \) is a constant, which is 0.03, in this paper as same the stander SMA (Li et al., 2020). Parameter \( \overrightarrow{kb} \) is a number between \([-a, a]\), \( \overrightarrow{k}c \) is a linearly decreasing number from 1 to 0, \( t \) represents the current iteration times, \( \overrightarrow{S^i}(t) \) represents the current best individual, \( \overrightarrow{S} \) represents the current individual, \( \overrightarrow{S_{\text{R1}}} \) and \( \overrightarrow{S_{\text{R2}}} \) represents two individuals randomly selected in the slime mould population. \( \overrightarrow{W} \) represents the weight.

The oscillation factor of SMA has an important relationship with the foraging speed and accuracy of slime mould. We find that the oscillation factor is larger and the search range is wider at the initial stage of iteration, which leads to the slow convergence of the algorithm and the local optimum at the initial stage of search; However, when the oscillation factor range is small at the initial stage of search; However, when the oscillation factor range is small at the initial stage of search, slime mould will accelerate its searching speed and precision. To solve this problem, we propose an improved oscillation factor, and the oscillation factor formula is as follows:

\[
\overrightarrow{kb} = [-a, a],
\]

\[
a = \gamma (\cos(\pi \cdot t/T_{\text{max}}) + \lambda),
\]

where \( a \) is the value of the change range of \( \gamma \), \( \gamma = 0.5 \); \( \lambda \) is the value step of the slime mould, \( \lambda = 1 \). The oscillation factor trend \( \overrightarrow{kb} \) of SMA and SSMA is shown in Figure 1. During the search, because the range of search elements is not large, the accuracy and speed of search can be improved, and the optimization speed of the algorithm can be improved.

The formula of weight \( \overrightarrow{W} \) is as follows:

\[
\overrightarrow{W}(\text{smell}(i)) = \begin{cases} 
1 + r \cdot \log \left( \frac{\overrightarrow{BF} - f(i)}{\overrightarrow{BF} - W_{\text{FF}+1}} + 1 \right), & \text{if } f(i) \leq N \frac{t}{T_{\text{max}}} , \\
1 - r \cdot \log \left( \frac{\overrightarrow{BF} - f(i)}{\overrightarrow{BF} - W_{\text{FF}+1}} + 1 \right), & \text{other},
\end{cases}
\]

\[
\text{smell} = \text{sort}(f),
\]

where \( r \) is a random number between \([0,1]\), \( T_{\text{max}} \) represents the maximum iteration times, \( \overrightarrow{BF} \) represents the best fitness value in the current iteration, \( \overrightarrow{WF} \) represents the worst fitness value obtained by the current iteration times, and \( \overrightarrow{smell} \) represents the corresponding values sorted by fitness values.

The SSMA mainly depends on the best leader and two random slime mould individuals in the population. Equation (13.2) is the core of the SSMA. When \( z \) less than 0.03, slime mould searches randomly in the whole search space, and Equation (13.1) is executed; otherwise, Equation (13.2) is executed. The best individual in Equation (13.2) can lead the development stage of slime mould, while two random individuals in the population mainly dominate the exploration stage of the SSMA. This formula can balance the exploration and development of the algorithm. The pseudo-code of the SMA is shown in Algorithm 2. The intuitive and detailed process of SSMA is shown in Figure 2.

### 5. Simulation result

#### 5.1. Problem definition

As mentioned in the second section, we will define it as follows: the coverage of sensor nodes is a circle with a fixed radius. Whether a certain position is in the coverage area of the sensor node can be determined by calculating the distance between the target point and the node. In order to evaluate the coverage of WSNs in a two-dimensional area, the whole monitoring area is divided into grids. If WSNs can cover points, then the coverage rate is 1. In addition, we assume that all sensors are the same monitoring area, and ignore the boundary effect on the sensor network.

During initialization, all sensor nodes are randomly scattered in a given monitoring area, and the initial
coordinates of these sensor nodes are the initial input values of the algorithm. Each search agent in the algorithm represents a placement scheme of sensor nodes. In the two-dimensional monitoring area, the dimension of search agent is twice the number of sensor nodes; in other words, the $2x-1$ dimensional data represents the abscissa of the $x$th sensor node, while the $2x$ dimensional data represents its ordinate. The algorithm takes WSN coverage as the fitness function that is Equation (4), as mentioned in the second section, and maximizes the fitness function as the optimization goal. Finally, the optimized coverage and the positions of all sensor nodes are output.

### 5.2. Parameter set

In this section, we compare the experimental results of SSMA with SMA (Li et al., 2020), PSO (Kennedy & Eberhart, 2002), GWO (Mirjalili et al., 2014), WOA (Mirjalili & Lewis, 2016), MPA (Mostafa et al., 2020) and FPA (Yang, 2012). Table 1 summarizes the experimental parameter settings for these algorithms. In order to eliminate the experimental error caused by chance, the average value of 20 independent runs is used as the comparison result. In this paper, we are mainly divided into two kinds of experiments: low dimension and high dimension. Other parameter settings: population size: $N$, maximum iteration: $T$; The monitoring radius is: $R$, the monitoring area is $L \times M$, the sensor of node is: $Node$, and the dimension is: $Dim$.

### 5.3. Comparative analysis of experiments

In this section, we compared the SSMA with some competitive MAs on WSN coverage optimization. Furthermore, all of the experimental series were performed on MATLAB.

---

**Table 1. Setting comparison algorithms parameter.**

| Algorithm          | Parameters and values |
|--------------------|-----------------------|
| PSO (Pakdel & Fotohi, 2021) | $C_1 = 1.5, C_2 = 2, W_{max} = 0.9, W_{min} = 0.4$ |
| FPA (Mirjalili & Lewis, 2016) | $p = 0.8$ |
| GWO (Agarwal & Bharti, 2021) | $a = [2,0]$ |
| WOA (Zubaidi et al., 2020) | $a = [2,0], b = 1, A = [2,0]$ |
| MPA (Mostafa et al., 2020) | $F_{ads} = 0.2, P = 0.5$ |
| SMA (Abdel-Baset et al., 2021) | $z = 0.03, \lambda = 1, y = 0.5$ |
| SSMA                | $z = 0.03, \lambda = 1, y = 0.5$ |
Table 2. \( T = 250, N = 40 \).

| Parameter | \( R \) | \( \text{Dim} \) | \( \text{Node} \) | \( L+M \) |
|-----------|--------|-----------|-----------|--------|
| Case 1    | 1m     | 20        | 10        | 5+5m   |
| Case 2    | 1m     | 40        | 40        | 10+10m |
| Case 3    | 1m     | 160       | 80        | 15+15m |

Table 3. \( T = 100, N = 30 \).

| Parameter | \( R \) | \( \text{Dim} \) | \( \text{Node} \) | \( L+M \) |
|-----------|--------|-----------|-----------|--------|
| C 1       | 10m    | 10        | 5         | 50+50  |
| C 2       | 10m    | 40        | 40        | 80+80  |
| C 3       | 10m    | 120       | 60        | 100+100|
| C 4       | 10m    | 160       | 80        | 150+150|

2017b and were run on a CPU Core i3-6100 v4 (3.70 GHz) with 8 GB RAM.

In the experimental part, we will set 13 groups of coverage optimization experiments, among which there are two groups of low-dimensional ones, which are divided into Case1–Case3 and C1–C4, the number of nodes is unequal between 5 and 80 nodes, besides, there are six groups of high-dimensional experiments, mainly from 100 to 1200 nodes.

5.3.1. Low-dimension experimental analysis

In this section, we mainly test and analyze the number of nodes in the low dimension. There are two types of cases in low dimension: Case1–Case3; C1–C4. The case Case1 to Case3 comes from the literature (Miao et al., 2020), C1–C4 are self-defined cases in this paper, and the parameter settings of the examples are shown in Tables 2 and 3. The experiments mainly compared the experimental results of SSMA with six state-of-the-art algorithms to analyze the robustness and convergence speed. In the previous section, we used the above two types with different ratios for experimental comparison. Case 1–Case 3 compares these three cases through Tables 4–7 and Figures 3–5, moreover, Tables 8–10 and Figures 6–8 represent the data tables and graphs of C1–C4 cases. Respectively, best values, worst values, average values and STD represent the best value, the worst value and STD of coverage. Black bold numbers represent the best value, dark grey bold represents the second-best value, and light grey bold represents the third best. Tables 4–7 respectively show the best value, the worst value, the average value and the standard deviation of independent operation for 20 times.

It can be clearly seen from Table 4 (best value) that PSO, SMA, FPA, GWO, WOA and MPA rank first in the coverage rate of Case1, which is 8% higher than PSO and SMA rank second in the coverage rate; In Case2, SSMA ranked first, with a coverage rate of 99%, and MPA ranked second was 1% higher than PSO, FPA, GWO, WOA and SMA by 21%, 13%, 5%, 11%, 1% and 19%. The coverage rate of SSMA in Case3 is 94.22%, which is 27.11%, 16%, 5.78%, 13.33%, 3.11% and 21.78% higher than PSO, FPA, GWO, WOA, MPA and SMA, respectively. In Case1 in Table 5 (the worst value), SSMA ranks second In Case2 and Case3, SSMA ranked first, MPA ranked second and FPA ranked third.

In Table 6 (average value) that SSMA is superior to other algorithms. In three examples, SSMA

| Table 4. Best values. |
|----------------------|
| Algorithm | PSO | FPA | GWO | WOA | MPA | SMA | SSMA |
| Case 1    | 0.9200 | 1   | 1   | 1   | 1   | 0.92 | 1    |
| Case 2    | 0.7800 | 0.86 | 0.94 | 0.88 | 0.98 | 0.8  | 0.99 |
| Case 3    | 0.6711 | 0.7822 | 0.8844 | 0.8089 | 0.9111 | 0.7244 | 0.9422 |

| Table 5. Worst values. |
|------------------------|
| Algorithm | PSO | FPA | GWO | WOA | MPA | SMA | SSMA |
| Case 1    | 0.6400 | 0.96 | 0.92 | 0.84 | 0.88 | 0.76 | 0.92 |
| Case 2    | 0.4900 | 0.84 | 0.8  | 0.8  | 0.8  | 0.75 | 0.94 |
| Case 3    | 0.4711 | 0.7511 | 0.6889 | 0.7022 | 0.8044 | 0.6889 | 0.8667 |

| Table 6. Average values. |
|--------------------------|
| Algorithm | PSO | FPA | GWO | WOA | MPA | SMA | SSMA |
| Case 1    | 0.7960 | 0.97 | 0.962 | 0.922 | 0.962 | 0.84 | 0.984 |
| Case 2    | 0.6815 | 0.8515 | 0.8925 | 0.8405 | 0.922 | 0.7685 | 0.9715 |
| Case 3    | 0.5842 | 0.7633 | 0.7909 | 0.7507 | 0.8589 | 0.7067 | 0.9044 |

| Table 7. Std values. |
|----------------------|
| Algorithm | PSO | FPA | GWO | WOA | MPA | SMA | SSMA |
| Case 1    | 0.0733 | 0.0178 | 0.0275 | 0.0378 | 0.0330 | 0.0486 | 0.0239 |
| Case 2    | 0.0613 | 0.0081 | 0.0481 | 0.0228 | 0.0337 | 0.0157 | 0.0139 |
| Case 3    | 0.0514 | 0.0076 | 0.0748 | 0.0254 | 0.0270 | 0.0107 | 0.0193 |
ranks first, while FPA ranks second, GWO and MPA rank third in Case1; MPA ranks second and GWO ranks third in Case2 and Case3. In Table 7 (standard deviation), SSMA still ranks in the top three, among which in Case1, SSMA ranks first with a standard deviation of 0.0239, FPA ranks second with 0.0178, and GWO ranks third with 0.0275; In Case2, FPA ranked first with a standard deviation of 0.0081, SSMA ranked second with a standard deviation of 0.0139 and SMA ranked third with a standard deviation of 0.0157. In Case3, SSMA ranks third, with a value of 0.0193, FPA ranks first, and SMA ranks third.

From Figures 3–5, it can be seen that the performance of SSMA is superior to several other algorithms compared with it. The convergence curve of Case1–Case3 in Figure 3 shows that the convergence curve of SSMA is much faster than the other six algorithms, and other algorithms are

Table 8. Best values.

| Algorithm | PSO | FPA | GWO | WOA | MPA | SMA | SSMA |
|-----------|-----|-----|-----|-----|-----|-----|------|
| C1        | 0.6344 | 0.6080 | 0.6332 | 0.6256 | 0.6308 | 0.5612 | 0.6356 |
| C2        | 0.7745 | 0.7416 | 0.8694 | 0.7838 | 0.8156 | 0.7017 | 0.8797 |
| C3        | 0.9331 | 0.9164 | 0.9855 | 0.9538 | 0.9804 | 0.8894 | 0.9977 |
| C4        | 0.6703 | 0.7284 | 0.8335 | 0.7814 | 0.8404 | 0.7181 | 0.8662 |

Table 9. Worst values.

| Algorithm | PSO | FPA | GWO | WOA | MPA | SMA | SSMA |
|-----------|-----|-----|-----|-----|-----|-----|------|
| C1        | 0.4980 | 0.5692 | 0.5888 | 0.5744 | 0.5984 | 0.5012 | 0.6232 |
| C2        | 0.6305 | 0.6855 | 0.7950 | 0.7028 | 0.7430 | 0.6427 | 0.8308 |
| C3        | 0.7852 | 0.8963 | 0.8701 | 0.9128 | 0.9365 | 0.8604 | 0.9870 |
| C4        | 0.5959 | 0.7115 | 0.6819 | 0.7272 | 0.7752 | 0.6769 | 0.8337 |
easy to fall into the local optimum or are not suitable for this problem, there is no way to optimize. In Figure 4, we can see that the standard deviation of SSMA is more stable and higher than other algorithms. As can be seen from the coverage in Figure 5, the coverage of SSMA is superior than other algorithms. As the dimensionality increases, the coverage performance is outstanding.

Tables 8–11 are comparative data of low-dimensional examples C1–C4, which respectively show the best value, the worst value, the average value and the standard deviation of independent operation for 20 times. It can be seen from Table 8 (Best values) that SSMA ranks first in C1–C4, with coverage rates of 0.6356, 0.8797, 0.9977 and 0.8662 respectively. The coverage rate of SSMA in C1 is 0.16%, 2.76%, 0.24%, 1%, 0.48% and 7.43% higher than that of PSO, FPA, GWO, WOA, MPA and SMA respectively. The coverage rate of SSMA in C2 is 8.36%, 13.37%, 1.03%, 9.59%, 6.41% and 17.8% higher than that of PSO, FPA, GWO, WOA, MPA and SMA respectively. In C3, the coverage rate is 7.46%, 8.13%, 1.22%, 4.39%, 1.73% and 10.83% higher than that of PSO, FPA, GWO, WOA, MPA and SMA respectively. The coverage rate of SSMA in C4 is 19.59%, 13.78%, 3.27%, 8.48%, 2.58% and 18.81% higher than that of PSO, FPA, GWO, WOA, MPA and SMA, respectively. The four examples of SSMA in Table 9 (Worst values) rank first, among which MPA ranks second in C1, GWO is ranked

Table 10. Average values.

| Algorithm | PSO | FPA | GWO | WOA | MPA | SMA | SSMA |
|-----------|-----|-----|-----|-----|-----|-----|------|
| C1        | 0.6144 | 0.5851 | 0.6249 | 0.5998 | 0.6139 | 0.5269 | 0.6305 |
| C2        | 0.7246 | 0.7136 | 0.8388 | 0.7499 | 0.7853 | 0.6708 | 0.8643 |
| C3        | 0.8601 | 0.9071 | 0.9258 | 0.9368 | 0.9627 | 0.8706 | 0.9947 |
| C4        | 0.6355 | 0.7213 | 0.7449 | 0.7549 | 0.8099 | 0.6963 | 0.8513 |

Figure 4. Box plot on case1 to case3.
second in C2 and MPA is ranked third. MPA is ranked second in C3 and C4, and GWO is ranked third. It can be seen from Table 10 (Average) that the average coverage rate of SSMA is significantly higher than that of the other six algorithms. The average coverage rate of SSMA in C1 reached 63.05%, ranking first, which was 1.61%, 4.54%, 0.56%, 3.07%, 1.66% and 10.36% higher than PSO, FPA, GWO, WOA, MPA and SMA respectively. The average

Figure 5. Nodes distribution optimization on case1 to case3.
coverage rate of SSMA in C2 reached 86.43%, which was 13.97%, 15.07%, 2.55%, 7.9% and 19.35% higher than PSO, FPA, GWO, WOA, MPA and SMA respectively. The average coverage rate of SSMA in C3 reaches 99.47%, which is 13.46%, 8.76%, 6.89%, 5.79%, 3.2% and 12.41% higher than PSO, FPA, GWO, WOA, MPA and SMA respectively. The average coverage rate of SSMA in C4 is 85.13%, which is 21.58%, 13%, 10.64%, 9.64%, 4.14% and
15.15% higher than PSO, FPA, GWO, WOA, MPA and SMA respectively. In Table 11 (Standard Deviation), SSMA ranks among C1-C3 three examples FPA ranks second and GWO ranks third in C2. FPA ranks second and SMA ranks third in C3. FPA ranks first and SMA ranks second in C4. From the above data display and analysis, it can be seen that SSMA has obvious advantages in solving the coverage optimization problem of WSNs, and has relatively strong robustness and fast convergence speed.

Figures 6–8 show the C1 to C4 of each algorithm, Figure 6 shows the convergence curve, Figure 7 shows the variance diagram, and Figure 8 shows the sensor coverage of each algorithm. From Figures 6–8, it can be concluded that SSMA has superior performance in coverage optimization in WSNs on the whole, and MPA has available performance in this optimization problem, which is generally second only to SSMA.

5.3.2. Large-scale experimental analysis

In this section, we test the proposed SSMA on a large-scale wireless sensor node and compare the results with other algorithms. The experimental parameters are the same as before. In experiments, the population size of the algorithm is set to 30, the maximum number of iterations is 100, and the average value of the parameters is taken after 20 independent operations of the algorithm. Table 12 has list the other parameters set.

Other algorithms may adapt in small-scale experiments, but they may lose performance in large-scale experiments. However, the SSMA is still applicable in the
Figure 6. Comparison coverage convergence curve.

### Table 11. Std values.

| Algorithm | PSO  | FPA  | GWO  | WOA  | MPA  | SMA  | SSMA |
|-----------|------|------|------|------|------|------|------|
| C1        | 0.0343 | 0.0106 | 0.0112 | 0.0135 | 0.0085 | 0.0176 | 0.0039 |
| C2        | 0.0383 | 0.0136 | 0.0165 | 0.0230 | 0.0199 | 0.0169 | 0.0124 |
| C3        | 0.0427 | 0.0056 | 0.0493 | 0.0115 | 0.0126 | 0.0083 | 0.0025 |
| C4        | 0.0196 | 0.0054 | 0.0614 | 0.0124 | 0.0130 | 0.0115 | 0.0068 |

### Table 12. \( T = 100, N = 30 \).

| Parameter | \( R \) | \( Dim \) | \( Node \) | \( L \times M \) |
|-----------|---------|---------|---------|---------|
| C5        | 45m     | 200     | 100     | 800×800 |
| C6        | 30m     | 600     | 300     | 800×800 |
| C7        | 25m     | 1000    | 500     | 800×800 |
| C8        | 20m     | 1400    | 700     | 800×800 |
| C9        | 18m     | 2000    | 1000    | 800×800 |
| C10       | 15m     | 2400    | 1200    | 800×800 |

large-scale experiment on the number of nodes. Table 13 shows the results of six large-scale experiments. From the table, it can be seen that the coverage SSMA of C5, C6, C7 and C9 are ranked first among the six large-scale experiments of C5–C10, and the SSMA of C8 and C10 are ranked second. In experiment C8, the coverage rate of SSMA was 0.18% less than that of MPA ranked first, and in experiment C10, the coverage rate of SSMA was 0.05% less than that of MPA ranked first. It can be seen that the SSMA ranked second in C8 and C10 experiments is not much different from the MPA ranked first. In experiment C5(200 dimensions), the coverage rate of SSMA reached 78.35%, which was 25.44%, 10.89%, 12.87%, 8.39%, 2.82% and 13.41% higher than that of PSO, FPA, GWO, WOA, MPA and SMA. In experiment C6(600 dimensions), the coverage rate of SSMA reached 83.32%, which was 29.67%, 6.94%, 8.54%, 5.41%, 0.49% and 8.68% higher than that of PSO, FPA, GWO, WOA, MPA and SMA. In experiment C7(1000 dimensions), the coverage rate of SSMA reached 86.29%, which was 30.44%, 5.46%, 6.95%, 4.32%, 0.21% and 6.92% higher than that of PSO, FPA, GWO, WOA, MPA and SMA. In experiment C8(1400 dimensions), the coverage rate of SSMA reached 81.02%, which was 29.58%, 4.45%, 5.47%, 3.43% and 5.54% higher than that of PSO, FPA, GWO, WOA and SMA. In experiment C9(2000 dimensions), the coverage rate of SSMA reached 85.15%, which was 31.64%, 3.94%, 5%, 0.1%, 3.22% and 4.83% higher than that of PSO, FPA, GWO, WOA, MPA and SMA. In experiment C10(2400 dimensions), the coverage rate of
SSMA reached 78.16%, which was 28.21%, 3.19%, 4.05%, 2.82% and 4.612% higher than that of PSO, FPA, GWO, WOA and SMA. From the above experimental results and experimental discussions, it shows that SSMA can solve the coverage optimization problem of WSNs. We can see that SSMA is still used in large-scale experiments, while other algorithms compared with SSMA have lost their performance. Experiments show that SSMA has superior performance and competitiveness.

Figure 9 shows the coverage convergence curve from C5 to C10. From the convergence graph, it can be seen that the convergence curve of SSMA is faster than other algorithms, and the optimization speed is stronger. With the increase of dimensions, the ability of optimization is less obvious and the difficulty of optimization increases. Figure 10 shows the graph of standard deviation, from which it can be seen that SSMA is the best, and Figure 11 shows the optimization of node distribution of randomly selected samples in each case. It can be observed that with the increase of iteration times, the network coverage gradually rises to the optimal value. However, the increase of the dimension of the optimization problem

---

**Figure 7.** Box plot on four cases.

**Table 13.** Average values.

| Algorithm | PSO | FPA | GWO | WOA | MPA | SMA | SSMA |
|-----------|-----|-----|-----|-----|-----|-----|------|
| C5        | 0.5291 | 0.6746 | 0.6548 | 0.6997 | 0.7553 | 0.6494 | 0.7835 |
| C6        | 0.5385 | 0.7638 | 0.7478 | 0.7791 | 0.8283 | 0.7464 | 0.8332 |
| C7        | 0.5585 | 0.8083 | 0.7934 | 0.8197 | 0.8608 | 0.7937 | 0.8629 |
| C8        | 0.5144 | 0.7657 | 0.7555 | 0.7759 | 0.8120 | 0.7548 | 0.8102 |
| C9        | 0.5351 | 0.8123 | 0.8015 | 0.8193 | 0.8505 | 0.8032 | 0.8515 |
| C10       | 0.4995 | **0.7497** | 0.7411 | 0.7534 | **0.7821** | 0.7404 | **0.7816** |
brings great challenges to the algorithm. It can be seen that with the increase of dimensions, the performance of optimization becomes a challenge, and the performance of optimization gradually weakens on high-dimensional tasks.

5.4. Analysis of statistical significance

Wilcoxon sum-rank test (WSRT), as a non-parameter test, can effectively evaluate the statistically significant difference between the two optimization algorithms.

Figure 8. Node distribution optimization on C1 and C4.
Tables 14–16 show the \( P \)-values of the Wilcoxon test (Hermann, 1984) obtained by different SSMA for 13 optimization questions, which are statistically significant when the significance level is 0.05. When we calculate the \( P \)-value higher than 0.05, it means that there is no significant difference between the two methods. On the contrary, when the \( P \)-value is less than 0.05, we can see that there are great differences between them. We use the Wilcoxon test to compare the performance difference between the two methods. Test results are shown in Table 12–14. ‘+’ indicates that the SSMA algorithm is superior to the comparison algorithm, ‘−’ indicates that the SSMA algorithm is second only to competitors, and ‘=’ indicates that there is no difference in performance between SSMA algorithm and comparison algorithm for each example in 20 runs. The results highlight the obvious advantages of SSMA over all other competitors in two different dimensional examples.

It can be seen from Tables 14–16 that the performance of our proposed SSMA is better than that of other algorithms. In addition, in Table 17, we use the Friedman test to rank several algorithms (Ashcroft & Pereira, 2002). According to the mean rank obtained by the Friedman test, the maximum mean rank of SSMA variables is 6.95, which indicates that the overall performance of the proposed SSMA in WSN coverage optimization is the best. The rank of other test algorithms is MPA > GWO > FPA > WOA > SMA > PSO.
Figure 9. Coverage convergence curve on C5–C 10.
Figure 10. Box plot on C5–C10.
Table 14. \( p \)-values results of Wilcoxon rank-sum test (SSMA for comparative samples).

| Algorithm | PSO       | FPA       | GWO       | WOA       | MPA       | SMA       |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Case1     | 8.1993e−05 | 0.1185    | 0.0239    | 3.4338e−04 | 0.0437    | 7.8386e−05 |
| Case2     | 8.4627e−05 | 7.6461e−05 | 8.6955e−05 | 8.6584e−05 | 2.8314e−04 | 8.4506e−05 |
| Case3     | 8.7949e−05 | 8.6461e−05 | 8.7949e−05 | 8.7699e−05 | 1.0078e−04 | 8.7699e−05 |

\(+/- = +/-\)

6. Conclusions and future direction

In this paper, an SSMA is proposed. It is based on the standard SMA to solve the coverage optimization problem of WSNs. For the SSMA, we have mainly made two improvements. The first was to the improved position-updating equation, mainly to improve the optimization performance of the SSMA and improve the coverage rate; the second is the improvement of the oscillation factor. The second part of the improvement was mainly improved the optimization speed in the early stage of search and the convergence speed. In the experiment section, we mainly conducted 13 groups of experiments: low-dimensional experiments and high-dimensional experiments. Low-dimensional experiments are mainly divided into two categories: small radius and large radius; There were mainly six groups in high-dimensional experiments. The experimental part is mainly compared with six state-of-the-art algorithms, namely PSO, FPA, GWO, WOA, MPA and SMA. In addition, Wilcoxon rank-sum test and Friedman test are used to determine the significant difference between the results of SSMA and other competitors. Experimental results show that these improvements to

![Figure 11. Node distribution optimization on C5 and C7.](image-url)
SSMA can improve search efficiency and speed up convergence. On this optimization problem, the proposed SSMA was superior to most test algorithms. The application shows that the SSMA is an efficient and reliable optimization algorithm. For future work, first, the proposed SSMA can be applied to other real-world problems, such as route path planning, transportation safety management, job shop scheduling, neural networks (Huang, 1999; Huang & Du, 2008; Zhang et al., 2006), clustering analysis (Du et al., 2007), and function approximate (Han & Huang, 2006), graph colouring problem. Second, other versions of SSMA can be extended, such as
multi-objective version, complex version, binary version, quantum coding version, etc. Finally, combining SSMA with other algorithms may be a promising aspect. Fourth, the updating formula and parameters of SSMA can be improved, and the correctness can be verified by experiments on benchmark problems.

**Disclosure statement**

No potential conflict of interest was reported by the authors.

**Funding**

This work is supported by the National Science Foundation of China [grant number 62066005], Project of Innovation Project of Guangxi Graduate [grant number YCSW2021157]. Project supported by the Hainan Provincial Natural Science Foundation of China [grant number 20QN237] and Education Department of Hainan Province, No. Hnky2020-5.

**References**

Abdel-Basset, M., Chang, V., & Mohamed, R. (2020). HSMA_WOA: A hybrid novel slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest x-ray images. *Applied Soft Computing*, 95, 106642. https://doi.org/10.1016/j.asoc.2020.106642

Abdel-Basset, M., Mohamed, R., Chakrabortty, R. K., Ryan, M. J., & Mirjalili, S. (2021). An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection. *Computers & Industrial Engineering*, 153, 107078. https://doi.org/10.1016/j.cie.2020.107078

Adulyasas, A., Sun, Z., & Wang, N. (2015). Connected coverage optimization for sensor scheduling in wireless sensor networks. *IEEE Sensors Journal*, 15(7), 3877–3892. https://doi.org/10.1109/JSEN.2015.2395958

Agarwal, D., & Bharti, P. S. (2021). Implementing modified swarm intelligence algorithm based on slime moulds for path planning and obstacle avoidance problem in mobile robots. *Applied Soft Computing*, 107, 107372. https://doi.org/10.1016/j.asoc.2021.107372

Ambareesh, S., & Madheswari, A. N. (2021). HRDSS-WMSN: A multi-objective function for optimal routing protocol in wireless multimedia sensor networks using hybrid red deer salp swarm algorithm. *Wireless Personal Communications*, 119(1), 117–146. https://doi.org/10.1007/s11277-021-08201-z

Ashcroft, S., & Pereira, C. (2002). *Practical Statistics for the Biological Sciences: Simple Pathways to Statistical Analyses* (pp. 105–108). Palgrave Macmillan.

Aziz, N. A., Alias, M. Y., & Mohammed, A. W. (2009). A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram. International Conference on Networking. IEEE.

Deng, W., Xu, J., Gao, X.-Z., & Zhao, H. (2022). An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(3), 1578–1587. https://doi.org/10.1109/TSMC.2020.3030792

Deng, W., Xu, J., Song, Y., & Zhao, H. (2021). Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem. *Applied Soft Computing*, 106, 106724. https://doi.org/10.1016/j.asoc.2020.106724

Djekidel, R., Bentouati, B., Javaid, M. S., Bouchekara, H. R. E., Bayoumi, A. S., & El-Sehiemy, R. A. (2021). Mitigating the effects of magnetic coupling between HV transmission line and metallic pipeline using slime mould algorithm. *Journal of Magnetism and Magnetic Materials*, 529, 167865. https://doi.org/10.1016/j.jmmm.2021.167865

Du, J.-X., Huang, D. S., Wang, X.-F., & Gu, X. (2007). Shape recognition based on neural networks trained by differential evolution algorithm. *Neurocomputing*, 70(4–6), 896–903. https://doi.org/10.1016/j.neucom.2006.10.026

**Tables**

| Table 15. *p*-Values results of Wilcoxon rank-sum test (SSMA for comparative samples). |
|-----------------|--------|--------|--------|--------|--------|--------|
| Algorithm       | PSO    | FPA    | GWO    | WOA    | MPA    | SMA    |
| C 1             | 0.0645 | 8.8074e−05 | 8.8575e−05 | 8.849e−05 | 1.1112e−04 | 8.8324e−05 |
| C 2             | 8.8575e−05 | 8.849e−05 | 4.4934e−04 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 |
| C 3             | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.849e−05 | 8.8575e−05 | 8.8575e−05 |
| C 4             | 8.849e−05 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 |
| +/- = /−        | 3/1/0  | 4/0/0  | 4/0/0  | 4/0/0  | 4/0/0/ | 4/0/0  |

| Table 16. *p*-Values results of Wilcoxon rank-sum test (SSMA for comparative samples). |
|-----------------|--------|--------|--------|--------|--------|--------|
| Algorithm       | PSO    | FPA    | GWO    | WOA    | MPA    | SMA    |
| C 5             | 8.8575e−05 | 8.8074e−05 | 8.8575e−05 | 8.849e−05 | 1.1112e−04 | 8.8324e−05 |
| C 6             | 8.8575e−05 | 8.849e−05 | 4.4934e−04 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 |
| C 7             | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.849e−05 | 8.8575e−05 | 8.8575e−05 |
| C 8             | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 |
| C 9             | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 0.390   | 8.8575e−05 |
| C 10            | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 8.8575e−05 | 0.6813  | 8.8575e−05 |
| +/- = /−        | 6/0/0  | 6/0/0  | 6/0/0  | 6/0/0  | 4/2/0  | 6/0/0  |

| Table 17. Overall Wilcoxon rank-sum test results and rank results. |
|-----------------|--------|--------|--------|--------|--------|--------|
| Results         | PSO    | FPA    | GWO    | WOA    | MPA    | SMA    |
| +/- = /−        | 12/1/0 | 13/0/0 | 12/1/0 | 13/0/0 | 11/2/0 | 13/0/0 |
| Mean rank       | 1.20   | 4.08   | 4.25   | 3.65   | 5.85   | 2.23   |
| Over rank       | 7      | 4      | 3      | 5      | 2      | 6      |
| Sigma           | 6.95   |        |        |        |        | 1      |
Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152, 113377. https://doi.org/10.1016/j.eswa.2020.113377

Gao, Z. M., Zhao, J., & Li, S. R. (2020). The improved slime mould algorithm with cosine controlling parameters (Vol. 1631, pp. 012083). IOP Publishing.

Han, F., & Huang, D. S. (2006). Improved extreme learning machine for function approximation by encoding a priori information. *Neurocomputing*, 69(16–18), 2369–2373. https://doi.org/10.1016/j.neucom.2006.02.013

Herrmann, D. (1984). *Wahrscheinlichkeitsrechnung und Statistik 30 BASIC-Programme*. Vieweg+Teubner Verlag.

Houssein, E. H., Mahdy, M. A., Blondin, M. J., Shebl, D., & Mohamed, W. M. (2021). Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Systems With Applications*, 174, 114689. https://doi.org/10.1016/j.eswa.2021.114689

Huang, D. S. (1999). Radial basis probabilistic neural networks: Model and application. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(7), 1083–1101. https://doi.org/10.1142/S0218001499000604

Huang, D. S., & Du, J.-X. (2008). A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. *IEEE Transactions on Neural Networks*, 19(2), 2099–2115. https://doi.org/10.1109/TNN.2008.2004370

Karaboga, D., Oktay, S., & Ozturk, C. (2010). Cluster based wireless sensor network routings using *Artificial Bee Colony Algorithm*. International Conference on Autonomous & Intelligent Systems. IEEE.

Kennedy, J., & Eberhart, R. (2002). *Particle swarm optimization*. ICNN95—international Conference on Neural Networks. IEEE.

Kuila, P., & Jana, P. K. (2014). A novel differential evolution based clustering algorithm for wireless sensor networks. *Applied Soft Computing*, 25, 414–425. https://doi.org/10.1016/j.asoc.2014.08.064

Kumar, C., Raj, T. D., Premkumar, M., & Raj, T. D. (2020). A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters. *Optik*, 223, 165277. https://doi.org/10.1016/j.ijleo.2020.165277

Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 111, 300–323. https://doi.org/10.1016/j.future.2020.03.055

Liao, W. H., Yao, Y., & Wu, R. T. (2011). Ant colony optimization based sensor deployment protocol for wireless sensor networks. *Expert Systems with Applications*, 38(6), 6599–6605. https://doi.org/10.1016/j.eswa.2010.11.079

Mendis, C., Guru, S. M., Halgamuge, S., & Fernando, S. (2006). Optimized sink node path using particle *swarm optimization*. Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on. IEEE Computer Society.

Miao, Z., Yuan, X., Zhou, F., Qiu, X., Song, Y., & Chen, K. (2020). Grey wolf optimizer with an enhanced hierarchy and its application to the wireless sensor network coverage optimization problem. *Applied Soft Computing*, 96, 106602. https://doi.org/10.1016/j.asoc.2020.106602

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69(3), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mostafa, M., Rezk, H., Aly, M., & Ahmed, E. M. (2020). A new strategy based on slime mould algorithm to extract the optimal model parameters of solar PV panel. *Sustainable Energy Technologies and Assessments*, 42, 100849. https://doi.org/10.1016/j.seta.2020.100849

Naik, M. K., Panda, R., & Abraham, A. (2022). Normalized square difference based multilevel thresholding technique for multispectral images using leader slime mould algorithm. *Journal of King Saud University – Computer and Information Sciences*, 34(1), 4524–4536. https://doi.org/10.1016/j.jsu.2020.10.030

Pakdel, H., & Fotohi, R. (2021). A firefly algorithm for power management in wireless sensor networks (WSNs). *The Journal of Supercomputing*, 77(9), 9411–9432. https://doi.org/10.1007/s11227-021-03639-1

Rajeswari, M., Thirugnanasambandam, K., Raghav, R. S., Prabu, U., Saravanan, D., & Anguraj, D. K. (2021). Flower pollination algorithm with Powell’s method for the minimum energy broadcast problem in wireless sensor network. *Wireless Personal Communications*, 119(2), 1111–1135. https://doi.org/10.1007/s11227-021-08253-1

Singh, A., Sharma, S., & Singh, J. (2021). Nature-inspired algorithms for wireless sensor networks: A comprehensive survey. *Computer Science Review*, 39, 100342. https://doi.org/10.1016/j.cosrev.2020.100342

Song, R., Xu, Z., & Liu, Y. (2018). Wireless sensor network coverage optimization based on fruit fly algorithm. *International Journal of Online Engineering (IJOE)*, 14(6), 58–70. https://doi.org/10.3991/jiioe.v14i06.8698

Song, Y., Wu, D., Deng, W., Gao, X.-Z., Li, T., Zhang, B., & Li, Y. (2021). MPPCEDE: Multi-population parallel co-evolutionary differential evolution for parameter optimization. *Energy Conversion and Management*, 228(2), 113661. https://doi.org/10.1016/j.enconman.2020.113661

Wang, X.-F., Huang, D. S., & Xu, H. (2010). An efficient local Chan-Vese model for image segmentation. *Pattern Recognition*, 43(3), 603–618. https://doi.org/10.1016/j.patcog.2009.08.002

Wei, Y., Zhou, Y., Luo, Q., & Deng, W. (2021). Optimal reactive power dispatch using an improved slime mould algorithm. *Energy Reports*, 7, 8742–8759. https://doi.org/10.1016/j.cergy.2021.11.138

Yang, X. S. (2012). Flower pollination algorithm for global optimization. International Conference on Unconventional Computing and Natural Computation. *Lecture Notes in Computer Science*, 7445, 240–249. https://doi.org/10.1007/978-3-642-32894-7_27

Zhang, J., Huang, D. S., Lok, T.-M., & Lyu, M. R. (2006). A novel adaptive sequential niche technique for multimodal function optimization. *Neurocomputing*, 69(16–18), 2396–2401. https://doi.org/10.1016/j.neucom.2006.02.016

Zhang, Q., & Fok, M. (2017). A two-phase coverage-enhancing algorithm for hybrid wireless sensor networks. *Sensors*, 17(12), 117. https://doi.org/10.3390/s17101117

Zhang, S., Zhou, Y., Li, Z., & Pan, W. (2016). Grey wolf optimizer for unmanned combat aerial vehicle path planning. *Advances in Engineering Software*, 99, 121–136. https://doi.org/10.1016/j.advsens.2016.05.015

Zhao, J., & Gao, Z.-M. (2020a). The hybridized Harris hawk optimization and slime mould algorithm. *Journal of Physics:*
Zhao, J., & Gao, Z. M. (2020b). The chaotic slime mould algorithm with Chebyshev map. *Journal of Physics: Conference Series, 1631*(1), 012071. https://doi.org/10.1088/1742-6596/1631/1/012071

Zhao, J., Gao, Z.-M., & Sun, W. (2020). The improved slime mould algorithm with Levy flight. *Journal of Physics: Conference Series, 1617*(1), 012033. https://doi.org/10.1088/1742-6596/1617/1/012033

Zhao, Z.-Q., Huang, D. S., & Sun, B.-Y. (2004). Human face recognition based on multi-features using neural networks committee. *Pattern Recognition Letters, 25*(12), 1351–1358. https://doi.org/10.1016/j.patrec.2004.05.008

Zhou, Y., Zhao, R., Luo, Q., & Wen, C. (2018). Sensor deployment scheme based on social spider optimization algorithm for wireless sensor networks. *Neural Processing Letters, 48*(1), 71–94. https://doi.org/10.1007/s11063-017-9701-8

Zubaidi, S. L., Abdulkareem, I. H., Hashim, K. S., Al-Bugharbee, H., Ridha, H. M., Gharghan, S. K., Al-Qaim, F. F., Muradov, M., Kot, P., & Al-Khaddar, R. (2020). Hybridised artificial neural network model with slime mould algorithm: A novel methodology for prediction of urban stochastic water demand. *Water, 12*(10), 2692. https://doi.org/10.3390/w12102692