Article

Diversity Teams in Soccer League Competition Algorithm for Wireless Sensor Network Deployment Problem

Yu Qiao [1], Thi-Kien Dao [2],*, Jeng-Shyang Pan [1,2],*, Shu-Chuan Chu [1] and Trong-The Nguyen [2,3]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China; amberjoe1214@163.com (Y.Q.); scchu0803@gmail.com (S.-C.C.)

[2] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350014, China; vnthe@hpu.edu.vn

[3] Department of Information Technology, Haiphong University of Manage and Technology, Haiphong 180000, Vietnam

* Correspondence: jvnkien@gmail.com (T.-K.D.); jengshyangpan@gmail.com (J.-S.P.)

check for updates

Abstract: The drawback of several metaheuristic algorithms is the dropped local optimal trap in the solution to complicated problems. The diversity team is one of the promising ways to enhance the exploration of searching solutions in algorithm to avoid the local optimum trap. This paper proposes a diversity-team soccer league competition algorithm (DSLC) based on updating team member strategies for global optimization and its applied optimization of Wireless sensor network (WSN) deployment. The updating team consists of trading, drafting, and combining strategies. The trading strategy considers player transactions between groups after the ending season. The drafting strategy takes advantage of draft principles in real leagues to bring new players to the association. The combining strategy is a hybrid policy of trading and drafting one. Twenty-one benchmark functions of CEC2017 are used to test the performance of the proposed algorithm. The experimental results of the proposed algorithm compared with the other algorithms in the literature show that the proposed algorithm outperforms the competitors in terms of having an excellent ability to achieve global optimization. Moreover, the proposed DSLC algorithm is applied to solve the problem of WSN deployment and achieved excellent results.

1. Introduction

Optimization is one of the most common problems found in life, e.g., engineering design, business planning, or even military applications. Optimization techniques are used to solve problems intelligently by choosing the optimal solution from a large number of solutions [1]. The meta-heuristic algorithm is viral for solving optimization problems as it is robust and straightforward [2–4], for example, reactive power planning problem in power systems [5], capacitated vehicle routing problem [6], and route planning of part process in flexible manufacturing systems [7]. Meta-heuristic algorithms have developed rapidly over the past few decades [8]. Metaheuristic algorithms are developed by taking inspiration from the natural phenomenon, e.g., physical, biological, or swarms moving as an advance seeks to generate or choose promising approximate solutions [9,10]. Examples of population-based algorithms of popular metaheuristics are as follows: genetic algorithm (GA) [11], firefly algorithms (FA) [12], adaptive differential evolution (JADE) [13], Cat swarm optimization

algorithm [14]. The optimization process of these meta-heuristic algorithms is usually used to generate a set of randomly initialized agents and to execute operations, e.g., selection, combination, and evaluation. The agents are iterated by a certain number of times through operations such as migration or evolution [15–17].

Soccer league competition (SLC) algorithm is a new metaheuristic algorithm that was inspired by the soccer league [18]. The competitive mechanism of football leagues is inspirited to model the algorithm for global optimization. The players in the football league are mimicked to simulate candidate solutions in the search space for the optimization algorithm. Improving the fitness of these players gradually as solutions in the algorithm is evaluated according to the objective function through iteration for optimization problems. The best fitness player is selected in the league as the optimal solution for a given optimization problem. Since it was proposed in 2014, in a short time, the SLC algorithm has been successfully applied to solve problems in several areas, e.g., the design of urban water supply networks [19,20], knapsack problems [21], and set covering problems [22]. The SLC algorithm has advantages such as fast convergence speed and more accurate results in comparison with the other metaheuristic algorithms [19]. However, the SLC algorithm also has a drawback, i.e., dropping in the optimal local trap when solving the complicated combinatorial problems.

This paper considers the improvement of the diversity teams to avoid the mentioned issue. We propose a novel improvement version of the SLC algorithm to enhance the diversity team's soccer league competition (namely DSLC) based on updating strategies. The strategies consist of trading transactions of the player between teams after ending each season, taking advantage of draft principles in real leagues to bring new players to the association, and combining the policies. Beside tested with the benchmark functions, we also apply the proposed method to solve the practical problem of deployment in Wireless Sensor Network (WSN).

Moreover, scholars has paid more attention to WSN because it is a network of many wireless sensors placed in the target space to monitor or capture environmental data, then transfer it to the base station. WSN has been widely applied in fields such as health care monitoring [23] and industrial monitoring [24]. However, its sensor nodes are limited in computation capability and storage capacity of the computing unit, in communication range and radio quality of communication unit, and the available energy of power units [25]. The implementation of WSN has faced many challenges related to determining node positions to ensure a trade-off between different metrics. Due to the differential density of WSN, finding an optimal deployment becomes an NP-Hard task [26,27]. To tackle this NP-Hardness of the deployment problem, meta-heuristic algorithms offer a good alternative method. The proposed method of enhanced diversity team to improve the metaheuristic SLC is a promising way for optimizing the design and deployment of WSN successfully. The reasonably arrange positioning nodes are modeled to maximize the coverage range in WSN as the optimization problem that would be solved by applying the proposed algorithm.

The paper structure is organized as follows. Section 2 reviews the related work, e.g., the original SLC algorithm, and the problem statement of the WSN deployment. Section 3 introduces a new proposed method of DSLC. Section 4 presents the simulation results of the proposed algorithm for selected testing functions. Section 5 applies the proposed algorithm to the coverage problems in WSN deployment. Finally, there is a concluding Section.

2. Related Work

2.1. Soccer League Competition Algorithm

Soccer league competition (SLC) [18] algorithm takes its inspiration from a soccer league that contains many teams. The teams are classified with two levels of strong and weak sides. The stronger teams will occupy the forefront of the league for a long time. The weaker teams will worry about relegation to the second league. During each season, there will be matches between teams. The number of matches in a season depends on the number of teams. In a season, a number of matches are held for

the league in competitions. After the end of each season, the worst record in the first league will be eliminated. The vacant seat is replaced by the two best teams in the second league. Assuming the league has several teams and the number of matches is calculated as follows.

$$M = (n \times (n-1))/2 \tag{1}$$

where M and n are the number of matches and teams, respectively. For each team, they play $n-1$ times per season. The players of a team are divided into leading players and reserve players. The fitness of the players determines the competence of the organization. The stronger is the club in the competition between groups, the higher is its probability of winning the game. For the internal competition of each side, the improvement of the team's fitness comes from the match between players. Each unit has the best player called Star Player (SP). For the league, there is one best player every season, named Super Star Player (SSP). Each team will make different adjustments after each game. The SP in each group will learn from the SSP in the league. SP wants to become an SSP by improving its fitness. The other players on the team learn from the SPs in each unit to enhance their fitness. At the end of each season, the team's fitness will increase. This strategy makes teams more competitive in the new season.

The player in the league considered a solution in the vector for optimization in which the locally optimal solution vectors and global optimal solution vectors are regarded as the star players in each team and superstar players in the league, respectively. The evaluation rate of the objective function for optimizing the problem is figured out based on the fitness of the player in the team.

$$PP(i, j) = \frac{1}{Func(i, j)}, \quad i \in \text{Team}, j \in \text{Players} \tag{2}$$

$$PT(i) = \frac{1}{n} \sum_{n=0}^{n} PP(i, j), \; i \in \text{Team}, j \in \text{Players} \tag{3}$$

where *Func* is the objective function; *PP* and *PT* are the power of the *j*th player in *team$_i$* and the power of *team$_i$* respectively; i, j are the number of the *i*-th team and the number of the *j*-th player on the team n is the number of players in a team. The high-power teams have a high probability of winning the game. The probability for two teams to win is as follows:

$$P_1 = \frac{PT_1}{PT_1 + PT_2} \tag{4}$$

$$P_2 = \frac{PT_2}{PT_1 + PT_2} \tag{5}$$

where P_1, P_2 are the probabilities for team$_1$ and team$_2$ to win. The sum of P_1 and P_2 equals to 1. PT_1 and PT_2 are the fitness of team1 and team2, respectively. The fitness of the players on the team (leading players and substitutes) is changed to improve forwarding to promising areas of searching space according to the fitness values of the players and the competitiveness of the team.

2.2. Deployment Problem for Wireless Sensor Networks

WSN deployment problem is to place sensor nodes in an accurate way that ensures high network performance [28]. Several terms can name this issue: placement, coverage, or deployment problem. Node deployment techniques are classified according to different criteria [29]. Depending on the application environment, deployment can be outdoor or indoor implementation. Based on the placement strategy, deployment can be random, deterministic, or scheduled. Random deployment, where nodes are placed randomly, is generally adopted in hostile environments or large-scale outdoor deployment areas [30]. Deterministic strategies are used for small-scale situations where we have a precise knowledge of the environment. In scheduled deployment, sensors are placed with higher density in areas where events are concentrated [31]. In indoor environments, random implementation

and scheduled deployment are infeasible due to existing of obstacles. The deterministic placement is the only option because it takes into account environmental characteristics [32]. The implementation must consider not only the environment characteristics but also should guarantee a high quality of services [33].

The optimization of WSN implementation has been facing many challenges related to determining node positions to ensure a trade-off between different metrics due to the differential density of WSN. Thus, to find an optimal deployment becomes an NP-Hard task [26]. For this optimization, the intelligent deterministic deployment algorithm is needed by applied metaheuristics [27]. The meta-heuristic offers an excellent alternative method to tackle the NP-Hardness of the deployment problem. These methods have received much attention in the literature. The deployment problem aims to optimize metrics simultaneously with different constraints of the optimization problem of deployment that have been introduced in several papers [34,35]. The mathematical techniques with assumptions, goals, and models were introduced to deal with deployment in WSN as optimization issues [36].

The coverage of the deployment problem is one of the exciting new studies in WSN, which is to improve the service quality of the network and to reduce the cost of deploying nodes. The node's deployment was studied in order to reduce power consumption with coverage maximization and is shown to prolong the WSN lifespan [37]. The problem is transformed into the essential question that studies how to maximize the coverage with the same node number. Assumed nodes of WSN deployed in two-dimensional space area can be modeled as a connected graph *G(V, E)*, where *V* and *E* are the finite set of vertices and edges, respectively, for sensor nodes. *V* is represented as $\{v_1, v_2, \ldots v_n\}$. The location of the node in the network v_i has coordinate (x_i, y_i) for $i = 1, \ldots n$. The connecting links of the vertices is a network represented as the set of edges *E* with $\{e_{1,2}, e_{1,3}, \ldots e_{i,j}, \ldots e_{n-1,n}\}$. The number of nodes and links of the network (*n* and *l*) are set to |V| and |E|, respectively. If an event is covered point (*x,y*) by sensor nodes vi. The probability that the point as the link can be defined as follows.

$$\alpha(e_i) = \begin{cases} 1, & \text{if } \mathrm{d}(v_i, \ p) < r, \\ 0, & \text{otherwise} \end{cases} , (\ i = 1, 2, \ldots n), \tag{6}$$

where e_i denoted an existed link of v_i, and $p(x, y)$ edge, *r* is a sensing radius of the nodes. A link equals $(v_i,)$ from $v_i \in$ V to $p \in$ V, and vice versa from $p(x, y)$ to v_i, and *d* is a Euclidean distance between the target point *p* and the sensor node. The calculation of Euclidean distance between target point p and a sensor node is expressed as follows.

$$\mathrm{d}(v_i, \ p) = \sqrt{(x_i - x)^2 + (y_i - y)^2} \tag{7}$$

The joint coverage rate for all nodes to monitor the pixel *p* in the monitoring area *S* as digital discrete of $m \times n$ pixel. The probability coverage in the monitoring area is computed as follows.

$$P(S, \ p) = 1 - \prod_{i=1}^{n} [1 - P(v_i, p) \times \alpha] \tag{8}$$

where *S* is the monitoring area, *P* is the coverage probability, and *n* is the number of nodes in the total size of a certain area. The coverage nodes layout of this deployment problem in WSN would be solved by applying the proposed method of diversity-team soccer league (DSLC) algorithm, as presented in Section 5.

3. Diversity Team Soccer League Competition Algorithm

This section presents a new proposed scheme of diversity-team soccer league competition optimization (DSLC) based on updating strategies to improve the SLC algorithm. The updating strategies are composed of trading, drafting, and combining strategies. The trading strategy considers trading transactions of the player between teams after the ending season. The drafting strategy takes

advantage of draft principles in real leagues to bring new players to the association. The combining approach is a combination of the policy of the trading and drafting ones. The principle of the DSLC algorithm is stated as follows.

Trading strategy:

The trading strategy is inspired by the players trading between teams in a real soccer league. While the offseason, the stronger teams are looking for potential star players to supplement the active players. The weaker teams are looking for deals to restructure the organization. This strategy can improve the step of league updating in the algorithm that is described as follows. The power of teams in the league is sorted, and then the top-ranked team is selected to trade with one leading player in a selected randomly team. The power of teams is evaluated according to the fitness rate function of the specific problem. The principle is expressed as equations below.

$$League = Sort([Team_1, \ Team_2, \ldots, Team_n]) \tag{9}$$

where n is a number of the team in the league, $Team_1.MainPlayer = [X_1, \ X_2, \ldots, X_a, \ldots, X_m]$,

$Team_n.MainPlayer = [Y_1, Y_2, \ldots, Y_b, \ldots, Y_m]$, m is a number of players, a and b represent any one of the $team_i$ and $team_j$, respectively. Then, we immigrate the top-ranked team with star players to a selected team randomly in the league and replace the weakened team with the top-ranked team.

$$Team_r MainPlayer = Team_1.MainPlayer \tag{10}$$

where r is a random index number.

Drafting strategy:

This strategy is inspired by the draft strategy of introducing new players in the real soccer league that was invented to bring young players into the league. The purpose of the draft strategy is to improve the inferior records of the competitiveness of teams by adding active players. First, we rank the players (who are about to enter the league) according to their strength from high to low, and then, we rate the teams based on the power from low to high. Here the strength or power is the fitness evaluated value of the local and global result. Second, we add the newly selected player in the team, and we also eliminate the weakest player from the group. The principle of the strategy is shown in Figure 1 below.
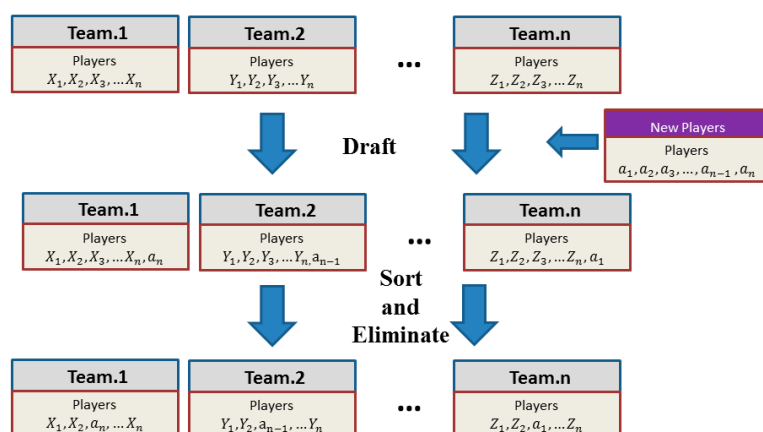
Figure 1. The principle of drafting strategy.

Figure 1 shows the principle of drafting strategy. The strength of teams is ranked from high to low, and set to $1, 2, \ldots, n$. The players on the group may be symbolled as X, Y, Z. Among new players, a_1 to a_n represent the rank of the strength of players from high to low. After sorting and eliminating, new players are added to the team, and the weakest players are removed. The advantage of the team could be improved by more diversity.

Combining Strategy

This strategy is a combination of strategies trading and drafting. Parameter random p is used to switch between the trading and drafting strategy. For every R_k season, if $p = random()$ is more significant than 1/2, the team will conduct (activate) the trading strategy, otherwise enable the drafting strategy, where R is a period of running iterations. The advantage of the trading strategy is that the variables of the population can be exchanged with each other, and the diversity of the community can be improved. The power of the drafting strategy can prevent the algorithm from falling into a locally optimal solution prematurely. Figure 2 shows a flowchart of the combining strategy (a combination of the trading and drafting strategy).
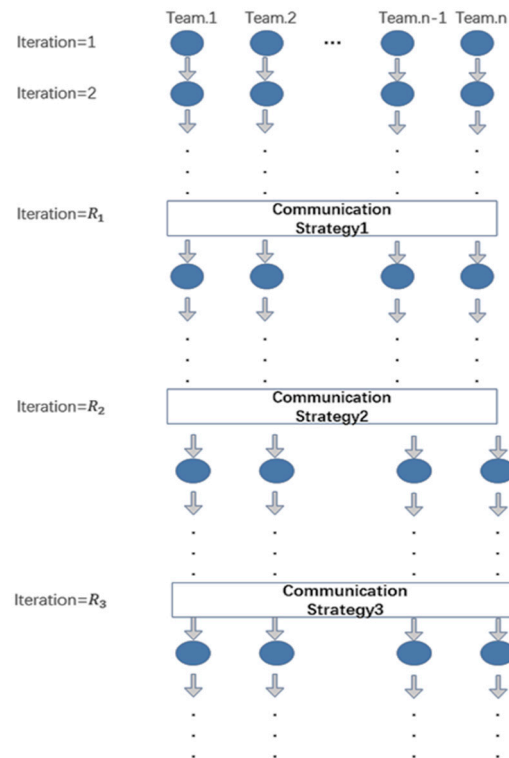
Figure 2. The combining strategy (a combination of the trading and drafting strategy).

The procedure steps of the proposed DSLC are presented as follows.

Step 1. Initialize: Generate N_P (population size) solutions randomly; calculate fitness function $F(x)$, where the fitness function is assumed to minimize $f(x)$, and the solution vector is $x = \{x_1, x_2, \ldots, x_n\} \in X$. X is the set of values for each variable.

Step 2. Generates sample: New solutions are generated in a league; the following equations, e.g., the number of players, is calculated as follows.

$$nPlayers = nT \times nMP + nRP \tag{11}$$

where nT is the number of teams in the league; nMP and nRP are the numbers of main player and reserve players.

The purpose of this step is to generate a random vector for each player as in following equations.

$$League = \begin{cases} Team1 \\ Team.2 \\ Team.3 \\ Team.4 \\ . \\ . \\ . \\ Team.n \end{cases} \tag{12}$$

$$Team = \begin{cases} MP.1 \\ MP.2 \\ . \\ . \\ . \\ MP.p \\ RP.1 \\ RP.2 \\ . \\ . \\ . \\ RP.q \end{cases} \tag{13}$$

$$MP = \left[X_{1,1}, X_{1,2}, \ldots, X_{1,k} \right] \tag{14}$$

$$RP = \left[X_{1,1}, X_{1,2}, \ldots, X_{1,k} \right] \tag{15}$$

where n is a number of teams per league; and p and q are main players and reserve players per team; k is a number of variables in a team.

Step 3. Team assessment: The strength of each player is calculated and ranked according to the evaluation of the fitness function. If a new solution of objective function value is better than the previous solution, it is replaced with the previous one and added to the team in order of player strength.

Step 4. League competes: The teams start to compete in matches. The operators of imitation and provocation are used to change the strength of the leading players and reserve players for the winner.

For imitation operator, as it is shown in Equation (16), the leading players of the winning team imitate the star player (*SP*) in their team and the superstar player (*SSP*) in the league to improve their strength. The following formulas are presented in the SLC algorithm as follows.

$$MP(i,j) = MP(i,j) + \alpha \cdot (SP - MP(i,j) + \beta \cdot (SSP - MP(i,j)) \tag{16}$$

In the equation, α and β are parameters for the main players to evolve to Super Player and Super star player, respectively. α and β are random numbers set between 0.7 and 1. For the provocation operator, as it is shown in Equation (17), the reserve players of the winning team imitate the average power of main players and provoke the main players. If the reserve players are better than the main players, the reserve players become the main players, and the main players become reserve players.

$$RP(i,j) = Average_{MP} + (\partial \cdot Average_{MP} - RP(i,j)) \tag{17}$$

In the equation, ∂ is the parameter for the reserve players to evolve to the average power of main players. ∂ is a random number set between 0.9 and 1.

Step 5. Updating league: Apply the proposed scheme above, updating strategies as communicating activation for enhancing diversity teams in the optimization algorithm.

Step 6. Relegation and promotion: The worst team in the first league is replaced by the best team in the second league.

Step 7. Check the stopping criterion: If the number of iterations is not reached, go to Step 3.

The pseudo-code of the process of the DSLC algorithm is shown in Algorithm 1.

Algorithm 1: Pseudo-code of the DSLC algorithm

1.　　**Input:**
2.　　　　Number of Teams: *nT*;
3.　　　　Fitness Function: $F(x)$;
4.　　　　Iteration: **T**
5.　　　　Winning factor: P_w
6.　　**Output:**
7.　　　　Best solution
8.　　**Initialization:**
9.　　　　Generate *nplayers* solutions vector (player) as *x*.
10.　　　Calculate solutions vector fitness F_p.
11.　　　Generate *nT* solutions (team) in a League.
12.　　　Sort players by F_p
13.　　　Assign players to the team in descending order.
14.　　**Main Loop:**
15.　　　t = 1;
16.　　　**While** (t < T) **do**
17.　　　　/*League Competes*/
18.　　　　Calculate the power of each team F_t.
19.　　　　Match between teams: (Take team A and team B as an example)
20.　　　　　If $F_{tA} > F_{tB}$
21.　　　　　　Rand p between 0 and 1
22.　　　　　　　If p > P_w
23.　　　　　　　Team A wins
24.　　　　　　Else
25.　　　　　　　Team B wins
26.　　　　　end If
27.　　　　　For the winning team:
28.　　　　　Main players: imitation operator
29.　　　　　Reserve players: provocation operator
30.　　　　/*Updating League*/
31.　　　　　Choose one of three strategies
32.　　　　　Strategy 1: Trading Strategy
33.　　　　　Strategy 2: Drafting Strategy
34.　　　　　Strategy 3: *Combining Strategy*
35.　　　　/*Relegation and promotion*/
36.　　　　　Initialize the last two teams in the last league
37.　　　　**End While**
38.　**Output:** The best player in League (best solution vector)

The next sections would present the results of testing the proposed algorithm of DSLC and its application for the node layout of the WSN to find the best node coverage.

4. Experimental Results of Testing Problems

In this section, for meeting test functions' diversity, twenty-one benchmark functions from the CEC2017 [38,39] are utilized to test the performance of the proposed DSLC. The selected benchmark problems consist of a broad range of test problems, e.g., uni-modal, multi-modal, and fixed-dimension functions. The detail presetting max iteration, dimensions, and boundaries for the function tests are listed in Table 1. The obtained optimal results of the proposed schemes (the DSLC-1, 2 and 3) are compared with the original SLC [18], genetic algorithm (GA) [11], firefly algorithm (FA) [12], and adaptive differential evolution (JADE) [13], respectively.

Different scenarios for testing the performance of the proposed DSLC are simulated, e.g., DSLC-Trading (DSLC-1), DSLC-Drafting (DSLC-2), and DSLC-Combining (DSLC-3) strategies, respectively, for the benchmark functions. The population size of the four algorithms, e.g., DSLC, SLC, GA, and FA in comparison is the same that N_p set to 80. The switch probability p is equal to 0.8. The parameter R equals to 10, as communication should be done every ten iterations, and the mutation rate equals to 0.05, crossover constant CR equals to 0.08 for GA. Because of the strong randomness of the meta-heuristic algorithms, the number of runs is set to 20, and their results are averaged.

The comparison results of the proposed DSLC-1, DSLC-2, and DSLC-3 with the SLC, FA, and GA algorithms are listed in Tables 2–4, respectively. In these tables, three symbols: "+", "-" and "~" are used to indicate the rate (r) of a pair of comparison results of the proposed schemes with other algorithms whose symbols can be "better", "worse", or "similar", respectively.

Table 2 shows a comparison of the obtained results of the proposed DSLC-Trading strategy (DSLC-1) with the SLC, FA, and GA algorithms for the test functions. As it can be seen in Table 2, the DSLC-Trading procedure (DSLC-1) performs well according to different rates that produce a higher number of "+" symbols than the other competitors, e.g., the SLC, FA, and GA algorithms.

Tables 3 and 4 show the comparison of the obtained results of the proposed DSLC-drafting strategy (DSLC-2) and DSLC-Combining strategy (DSLC-3) with the SLC, FA, and GA algorithms respectively for the test functions. As can be observed from the tables, the obtained results from the proposed schemes provide a better performance than that of the other competitors. It can be concluded that the proposed DSLC scheme provides a higher number of "winner +" than the FA, GA, and SLC methods for global optimization. It means that the proposed DSLC offers excellent performance in dealing with different test optimization functions.

Table 5 depicts a comparison of the JADE algorithm [13] with the proposed DSLC-strategies (DSLC-1, 2, and 3) for the test functions. As can be seen in the table, the DSLC-procedures also perform well, with a number of winners than the JADE algorithm. That means JADE obtained more "-" symbols than the DSLC scheme.

Figures 3–6 show the comparison results of the proposed schemes of DSLC-1, 2, and 3 with the SLC, FA, and GA algorithms listed in (a), (b), and (c), respectively. Panel (d) compares three schemes of strategies with the SLC algorithm for testing function 3.

Table 1. Twenty-one selected benchmark functions.

Number	Function Name	Function Expression	Range	Dimension	Iteration
1	Spherical	$f_1(x) = \sum_{i=1}^{N} x_i^2$	±100	30	100
2	High Conditioned Elliptic	$f_2(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	±100	30	100
3	Sum Square	$f_3(x) = \sum_{i=1}^{D} i x_i^2$	±100	30	100
4	Schwefel	$f_4(x) = 418.983n - \sum_{i=1}^{N} x_i \times sin\left(\sqrt{|x_i|}\right)$	±100	30	100

Table 1. *Cont.*

Number	Function Name	Function Expression	Range	Dimension	Iteration
5	Rotated Schwefel2	$f_5(x) = 418.983n - \sum_{i=1}^{N} x_i \times \sin\left(\sqrt{|x_i|}\right) + f_5^*$	±100	30	100
6	Quadric	$f_6(x) = \sum_{i=1}^{n} \sum_{k=1}^{i} x_i$	±10	30	100
7	Quartic Noisy	$f_7(x) = random[0,1) + \sum_{i=1}^{N} i \times x_i^4$	±10	30	100
8	Rosenbrock	$f_8(x) = \sum_{i=1}^{n-1} \left(100 \times \left(x_{i-1} - x_i^2\right)^2 + (1-x_i)^2\right)$	±600	30	100
9	Rastrigin	$f_9(x) = \sum_{i=1}^{D} [10 + x_i^2 - 10cos2\pi x_i$	±100	30	1000
10	Non-continuous Rotated Rastrigin's	$f_{10}(x) = \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right) + f_{10}^*$	±10	30	1000
11	Girewank	$f_{11}(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	±32	30	100
12	Ackley	$f_{12}(x)$ $= 20 + e - 20\exp\left(-\frac{1}{5}\sqrt{\left(\frac{1}{D}\sum_{i=1}^{D}\cos(i,j)\right)}\right)$ $- \exp\left(\frac{1}{D}\sum_{i=1}^{D} x_i^2\right)$	±10	2	100
13	Levy	$f_{13}(x) = (1 - w_d)^2\left[\sin^2(2\pi w_D) + 1\right] +$ $\sum_{i=1}^{D-1} (1 - w_i)^2\left[1 + 10\,\sin^2(1 + \pi w_i)\right] + \sin^2(\pi w_1)$	±512	2	100
14	Weierstrass	$f_{14}(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{kmax}\left[a^k\cos\left(2\pi b^k(x_i + 0.5)\right)\right]\right) -$ $D\sum_{k=0}^{kmax}\left[a^k\cos\left(2\pi b^k \cdot 0.5\right)\right]$	±10	30	100
15	Schaffer	$f_{15} = \frac{1}{2} + \frac{sin^2\left(x_1^2 - x_2^2\right) - 0.5}{\left[1 + 0.001 \times \left(x_1^2 - x_2^2\right)\right]^2}$	±100	2	100
16	Penalized1	$f_{16}(x) = \frac{\pi}{D}\sum_{i=1}^{D}\left(10\sin(\pi x_i)\right)^2 + \sum_{i=1}^{D}\left(0.25(x_i + 1)\right)^2$	±10	30	100
17	Penalized2	$f_{17}(x) = \sum_{i=1}^{D}\left(0.1(\sin(\pi x_i)\right)^2$	±10	30	100
18	Alpine	$f_{18}(x) = \sum_{i=1}^{D}\left|x_i * \sin(x_i) + 0.1x_i\right|$	±10	30	100
19	Himmelblau	$f_{19}(x) = \frac{1}{D}\sum_{i=1}^{D}\left(x_i^4 - 16 * x_i^2 + 5x_i\right)$	±10	30	100
20	Shifted rastrigin	$f_{20}(x) = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i)\right) + 10 + f22$	±10	30	100
21	Shifted griewank	$f_{21}(x) = \sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	±10	30	100

Table 2. Comparison of the obtained results of the proposed DSLC-Trading strategy (DSLC-1) with the SLC, FA, and GA algorithms for the test functions.

Test Functions	DSLC-1	SLC	r	FA	r	GA	r
1	**1.50 × 10^{-323}**	6.10 × 10^{-321}	+	2.54 × 10^{-06}	+	7.63 × 10^{-10}	+
2	**2.00 × 10^{-323}**	8.15 × 10^{-322}	+	1.48 × 10^{-11}	+	4.68 × 10^{-10}	+
3	**6.90 × 10^{-323}**	5.67 × 10^{-320}	+	4.92 × 10^{-08}	+	3.66 × 10^{-09}	+
4	**4.90 × 10^{-324}**	2.50 × 10^{-323}	+	3.10 × 10^{-03}	+	8.27 × 10^{-05}	+
5	4.90 × 10^{-324}	4.90 × 10^{-324}	~	2.80 × 10^{-04}	+	1.63 × 10^{-05}	+
6	9.80 × 10^{-322}	**2.50 × 10^{-323}**	−	2.35 × 10^{-13}	+	9.40 × 10^{-22}	+
7	**7.01 × 10^{-05}**	7.99 × 10^{-05}	+	6.11 × 10^{-04}	+	2.03 × 10^{-02}	+
8	1.96 × 10^{+01}	2.15 × 10^{+01}	+	2.32 × 10^{+01}	+	**1.01 × 10^{-00}**	−
9	3.20 × 10^{-14}	**1.60 × 10^{-14}**	+	2.89 × 10^{+01}	+	1.54 × 10^{-08}	+

Table 2. *Cont.*

Test Functions	DSLC-1	SLC	r	FA	r	GA	r
10	1.70×10^{-15}	1.78×10^{-15}	−	$5.30 \times 10^{+01}$	+	1.24×10^{-07}	+
11	5.50×10^{-15}	$\mathbf{4.44 \times 10^{-15}}$	+	9.90×10^{-03}	+	8.73×10^{-11}	+
12	$\mathbf{8.88 \times 10^{-16}}$	8.88×10^{-16}	~	5.23×10^{-04}	+	8.23×10^{-05}	+
13	$\mathbf{5.67 \times 10^{-30}}$	3.01×10^{-28}	+	9.81×10^{-06}	+	2.07×10^{-05}	+
14	$\mathbf{-1.50 \times 10^{+01}}$	$-1.50 \times 10^{+01}$	~	$\mathbf{-1.59 \times 10^{+01}}$	−	$\mathbf{-4.90 \times 10^{-01}}$	−
15	6.14×10^{-14}	$\mathbf{2.78 \times 10^{-16}}$	−	1.27×10^{-01}	+	9.72×10^{-03}	+
16	$\mathbf{3.68 \times 10^{-16}}$	4.99×10^{-13}	+	6.25×10^{-09}	+	4.01×10^{-06}	+
17	$\mathbf{2.02 \times 10^{-31}}$	2.27×10^{-13}	+	7.43×10^{-08}	+	9.45×10^{-12}	+
18	$\mathbf{4.90 \times 10^{-324}}$	1.50×10^{-323}	+	3.10×10^{-04}	+	4.58×10^{-06}	+
19	$-7.41 \times 10^{+01}$	$\mathbf{-7.54 \times 10^{+01}}$	−	$-7.46 \times 10^{+01}$	+	$-7.83 \times 10^{+01}$	−
20	$\mathbf{3.55 \times 10^{-15}}$	-7.54×10^{-15}	+	$3.08 \times 10^{+01}$	+	2.03×10^{-08}	+
21	7.55×10^{-15}	$\mathbf{5.55 \times 10^{-16}}$	−	2.30×10^{-08}	+	4.22×10^{-10}	+
Summary		13+ 3~ 5−		20+ 0~ 1−		18+ 0~ 3−	

Table 3. Comparison of the obtained results of the proposed DSLC-Drafting strategy (DSLC-2) with the SLC, FA, and GA algorithms for the test functions.

Test Functions	DSLC-2	SLC	r	FA	r	GA	r
1	$\mathbf{4.90 \times 10^{-324}}$	6.10×10^{-321}	+	2.54×10^{-06}	+	7.63×10^{-10}	+
2	$\mathbf{1.50 \times 10^{-323}}$	8.15×10^{-322}	+	1.48×10^{-11}	+	4.68×10^{-10}	+
3	$\mathbf{4.90 \times 10^{-324}}$	5.67×10^{-320}	+	4.92×10^{-08}	+	3.66×10^{-09}	+
4	$\mathbf{1.50 \times 10^{-323}}$	2.50×10^{-323}	+	3.10×10^{-03}	+	8.27×10^{-05}	+
5	$\mathbf{4.90 \times 10^{-324}}$	4.90×10^{-324}	~	2.80×10^{-04}	+	1.63×10^{-05}	+
6	2.96×10^{-322}	$\mathbf{2.50 \times 10^{-323}}$	−	2.35×10^{-13}	+	9.40×10^{-22}	+
7	$\mathbf{7.14 \times 10^{-05}}$	7.99×10^{-05}	+	6.11×10^{-04}	+	2.03×10^{-02}	+
8	$\mathbf{2.13 \times 10^{+01}}$	$2.15 \times 10^{+01}$	+	$2.32 \times 10^{+01}$	+	$\mathbf{1.01 \times 10^{-00}}$	−
9	$\mathbf{1.78 \times 10^{-15}}$	1.60×10^{-14}	+	$2.89 \times 10^{+01}$	+	1.54×10^{-08}	+
10	3.55×10^{-15}	$\mathbf{1.78 \times 10^{-15}}$	−	$5.30 \times 10^{+01}$	+	1.24×10^{-07}	+
11	8.99×10^{-15}	$\mathbf{4.44 \times 10^{-15}}$	+	9.90×10^{-03}	+	8.73×10^{-11}	+
12	8.88×10^{-16}	8.88×10^{-16}	~	5.23×10^{-04}	+	8.23×10^{-05}	+
13	3.90×10^{-12}	$\mathbf{3.01 \times 10^{-28}}$	−	$\mathbf{9.81 \times 10^{-06}}$	-	2.07×10^{-05}	−
14	$\mathbf{-1.50 \times 10^{+01}}$	$-1.50 \times 10^{+01}$	~	$-1.59 \times 10^{+01}$	+	$\mathbf{-4.90 \times 10^{-01}}$	+
15	$\mathbf{5.55 \times 10^{-17}}$	2.78×10^{-16}	+	1.27×10^{-01}	+	9.72×10^{-03}	+
16	$\mathbf{5.32 \times 10^{-14}}$	4.99×10^{-13}	+	6.25×10^{-09}	+	4.01×10^{-06}	+
17	$\mathbf{4.03 \times 10^{-19}}$	2.27×10^{-13}	+	7.43×10^{-08}	+	9.45×10^{-12}	+
18	3.50×10^{-323}	$\mathbf{1.50 \times 10^{-323}}$	−	$\mathbf{3.10 \times 10^{-04}}$	-	4.58×10^{-06}	−
19	$-7.17 \times 10^{+01}$	$\mathbf{-7.54 \times 10^{+01}}$	−	$\mathbf{-7.46 \times 10^{+01}}$	-	$\mathbf{-7.83 \times 10^{+01}}$	−
20	2.13×10^{-14}	$\mathbf{-7.54 \times 10^{-15}}$	−	$3.08 \times 10^{+01}$	+	2.03×10^{-08}	+
21	3.33×10^{-16}	5.55×10^{-16}	+	2.30×10^{-08}	-	4.22×10^{-10}	+
Summary		**12+** **3~** **6−**		20+ 0~ 1−		18+ 0~ 3−	

Table 4. Comparison of the obtained results of the proposed DSLC-Combining strategy (DSLC-3) with the SLC, FA, and GA algorithms for the test functions.

Test Functions	DSLC-3	SLC	r	FA	r	GA	r
1	$\mathbf{1.00 \times 10^{-323}}$	6.10×10^{-321}	+	2.54×10^{-06}	+	7.63×10^{-10}	+
2	$\mathbf{5.90 \times 10^{-323}}$	8.15×10^{-322}	+	1.48×10^{-11}	+	4.68×10^{-10}	+
3	$\mathbf{2.19 \times 10^{-321}}$	5.67×10^{-320}	+	4.92×10^{-08}	+	3.66×10^{-09}	+
4	$\mathbf{4.90 \times 10^{-324}}$	2.50×10^{-323}	+	3.10×10^{-03}	+	8.27×10^{-05}	+
5	$\mathbf{4.90 \times 10^{-324}}$	4.90×10^{-324}	~	2.80×10^{-04}	+	1.63×10^{-05}	+
6	3.10×10^{-322}	$\mathbf{2.50 \times 10^{-323}}$	−	2.35×10^{-13}	+	9.40×10^{-22}	+
7	1.22×10^{-04}	$\mathbf{7.99 \times 10^{-05}}$	−	6.11×10^{-04}	+	2.03×10^{-02}	+
8	$\mathbf{1.35 \times 10^{+01}}$	$2.15 \times 10^{+01}$	+	$2.32 \times 10^{+01}$	−	1.01×10^{-00}	+
9	1.15×10^{-13}	$\mathbf{1.60 \times 10^{-14}}$	−	$2.89 \times 10^{+01}$	+	1.54×10^{-08}	+
10	2.29×10^{-13}	$\mathbf{1.78 \times 10^{-15}}$	−	$5.30 \times 10^{+01}$	+	1.24×10^{-07}	+
11	3.33×10^{-16}	$\mathbf{4.44 \times 10^{-15}}$	−	9.90×10^{-03}	+	8.73×10^{-11}	+
12	$\mathbf{8.88 \times 10^{-16}}$	8.88×10^{-16}	~	5.23×10^{-04}	+	8.23×10^{-05}	+
13	$\mathbf{1.23 \times 10^{-29}}$	3.01×10^{-28}	+	9.81×10^{-06}	+	2.07×10^{-05}	+
14	$-1.50 \times 10^{+01}$	$-1.50 \times 10^{+01}$	~	$-1.59 \times 10^{+01}$	+	$\mathbf{-4.90 \times 10^{-01}}$	−
15	1.67×10^{-16}	2.78×10^{-16}	+	1.27×10^{-01}	+	9.72×10^{-03}	+
16	1.56×10^{-31}	4.99×10^{-13}	+	6.25×10^{-09}	+	4.01×10^{-06}	+
17	8.27×10^{-30}	2.27×10^{-13}	+	7.43×10^{-08}	+	9.45×10^{-12}	+
18	$\mathbf{1.00 \times 10^{-323}}$	1.50×10^{-323}	+	3.10×10^{-04}	+	4.58×10^{-06}	+
19	$\mathbf{-7.44 \times 10^{+01}}$	$-7.54 \times 10^{+01}$	+	$-7.46 \times 10^{+01}$	−	$-7.83 \times 10^{+01}$	−
20	4.09×10^{-14}	$\mathbf{-7.54 \times 10^{-15}}$	−	$3.08 \times 10^{+01}$	+	2.03×10^{-08}	+
21	2.66×10^{-15}	$\mathbf{5.55 \times 10^{-16}}$	−	2.30×10^{-08}	+	4.22×10^{-10}	+
Summary		11+ 3~ 7−		19+ 0~ 2−		19+ 0~ 2−	

Table 5. Comparison of the JADE algorithm with the proposed DSLC-strategies (DSLC-1, 2, and 3) for the test functions.

Test Functions	JADE	DSLC-1	r	DSLC-2	r	DSLC-3	r
1	6.10×10^{-323}	1.50×10^{-323}	−	4.90×10^{-324}	-	1.00×10^{-323}	-
2	8.15×10^{-322}	2.00×10^{-323}	−	1.50×10^{-323}	-	5.90×10^{-323}	-
3	5.67×10^{-320}	6.90×10^{-323}	−	4.90×10^{-324}	-	2.19×10^{-321}	-
4	2.50×10^{-323}	4.90×10^{-324}	−	1.50×10^{-323}	-	4.90×10^{-324}	-
5	4.90×10^{-324}	4.90×10^{-324}	−	4.90×10^{-324}	~	4.90×10^{-324}	~
6	$\mathbf{2.50 \times 10^{-323}}$	9.80×10^{-322}	+	2.96×10^{-322}	+	3.10×10^{-322}	+
7	$\mathbf{7.99 \times 10^{-05}}$	7.01×10^{-05}	+	7.14×10^{-05}	-	1.22×10^{-04}	+
8	$2.15 \times 10^{+01}$	$1.96 \times 10^{+01}$	−	$2.13 \times 10^{+01}$	-	$1.35 \times 10^{+01}$	
9	$\mathbf{1.60 \times 10^{-14}}$	3.20×10^{-14}	+	1.78×10^{-15}	-	1.15×10^{-13}	+
10	$\mathbf{1.78 \times 10^{-15}}$	1.70×10^{-15}	+	3.55×10^{-15}	+	2.29×10^{-13}	+
11	$\mathbf{4.44 \times 10^{-15}}$	5.50×10^{-15}	−	8.99×10^{-15}	+	3.33×10^{-16}	+

Table 5. *Cont.*

| Test Functions	JADE	DSLC-1	r	DSLC-2	r	DSLC-3	r
12	8.88×10^{-16}	8.88×10^{-16}	~	8.88×10^{-16}	~	8.88×10^{-16}	~
13	$\mathbf{3.01 \times 10^{-28}}$	5.67×10^{-30}	–	3.90×10^{-12}	+	1.23×10^{-29}	
14	$-1.50 \times 10^{+01}$	$-1.50 \times 10^{+01}$	~	$-1.50 \times 10^{+01}$	~	$-1.50 \times 10^{+01}$	~
15	$\mathbf{2.78 \times 10^{-16}}$	6.14×10^{-14}	+	5.55×10^{-17}	-	1.67×10^{-16}	-
18	4.99×10^{-13}	3.68×10^{-16}	–	5.32×10^{-14}	-	1.56×10^{-31}	-
19	2.27×10^{-13}	2.02×10^{-31}	–	4.03×10^{-19}	-	8.27×10^{-30}	-
20	$\mathbf{1.50 \times 10^{-323}}$	4.90×10^{-324}	+	3.50×10^{-323}	+	1.00×10^{-323}	-
21	$\mathbf{-7.24 \times 10^{+01}}$	$-7.41 \times 10^{+01}$	–	$-7.17 \times 10^{+01}$	+	$-7.44 \times 10^{+01}$	-
22	$\mathbf{7.11 \times 10^{-15}}$	3.55×10^{-15}	+	2.13×10^{-14}	+	4.09×10^{-14}	+
23	$\mathbf{5.55 \times 10^{-16}}$	7.55×10^{-15}	–	3.33×10^{-16}	-	2.66×10^{-15}	+
Summary		8+ 2~ 11–		5+ 3~ 13–		7+ 3~ 11–	
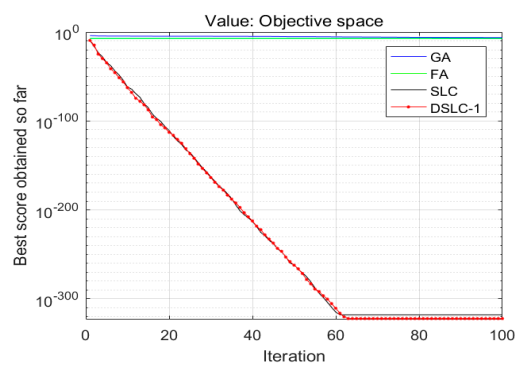
(a)
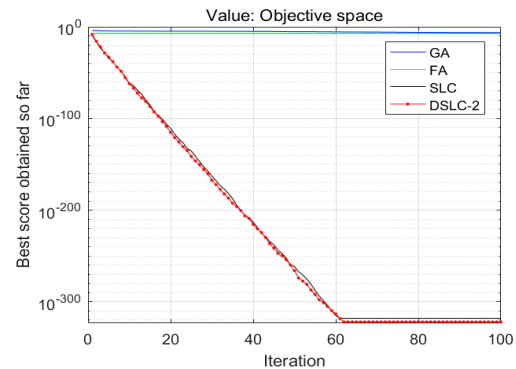
(b)
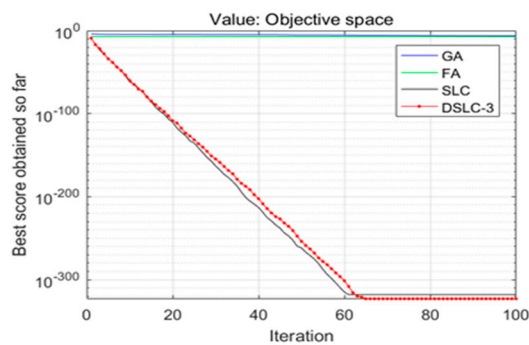
(c)
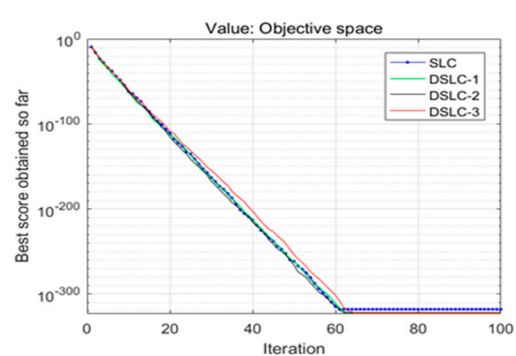
(d)

Figure 3. Comparison results of the proposed DSLC-1, 2, and 3 with the SLC, FA, and GA that listed (**a**), (**b**), and (**c**), respectively; (**d**) shows three schemes of strategies with the SLC algorithm for testing function 3.
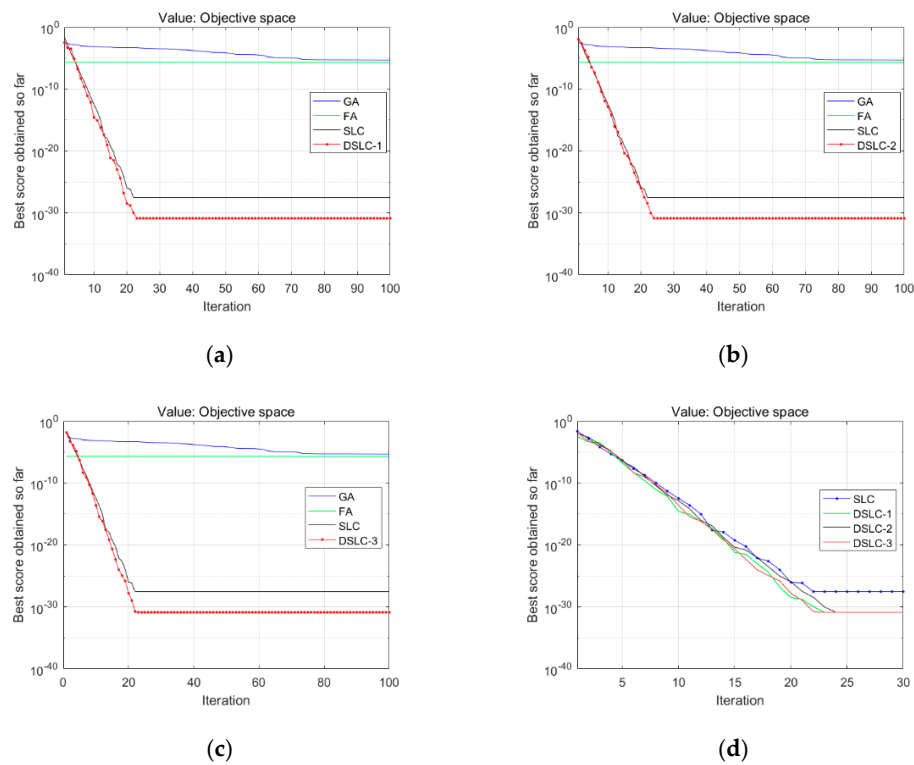
Figure 4. Comparison results of the proposed DSLC-1, 2, and 3 with the SLC, FA, and GA that listed (**a**), (**b**), and (**c**), respectively; (**d**) shows three schemes of strategies with the SLC algorithm for testing function 13.
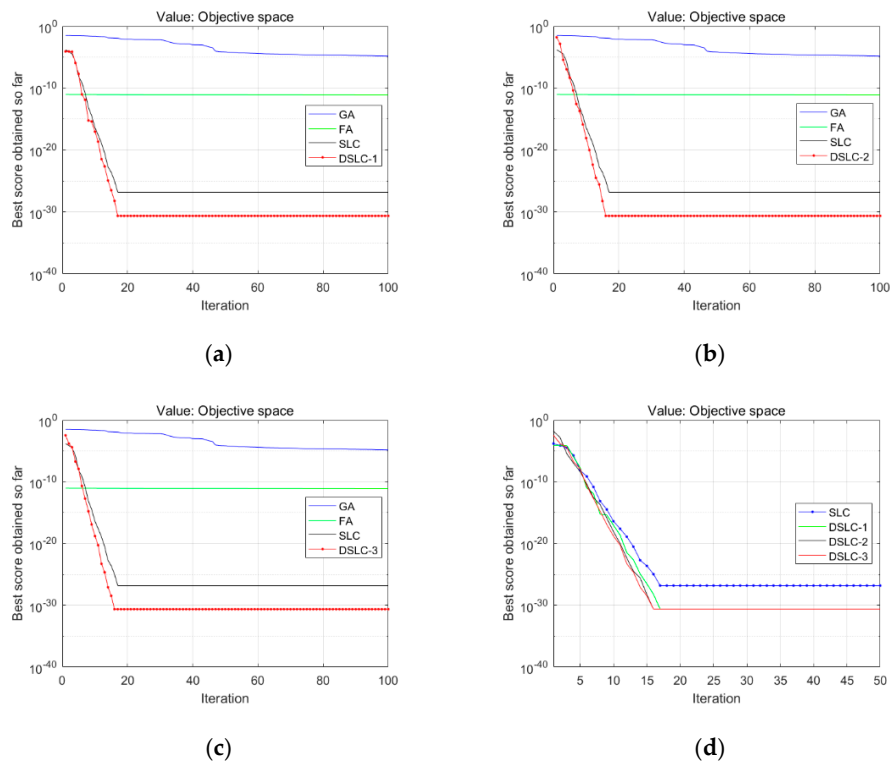
Figure 5. Comparison results of the proposed DSLC-1, 2, and 3 with the SLC, FA, and GA that listed (**a**), (**b**), and (**c**), respectively; (**d**) shows three schemes of strategies with the SLC algorithm for testing function 16.

(a)

(b)

(c)

(d)
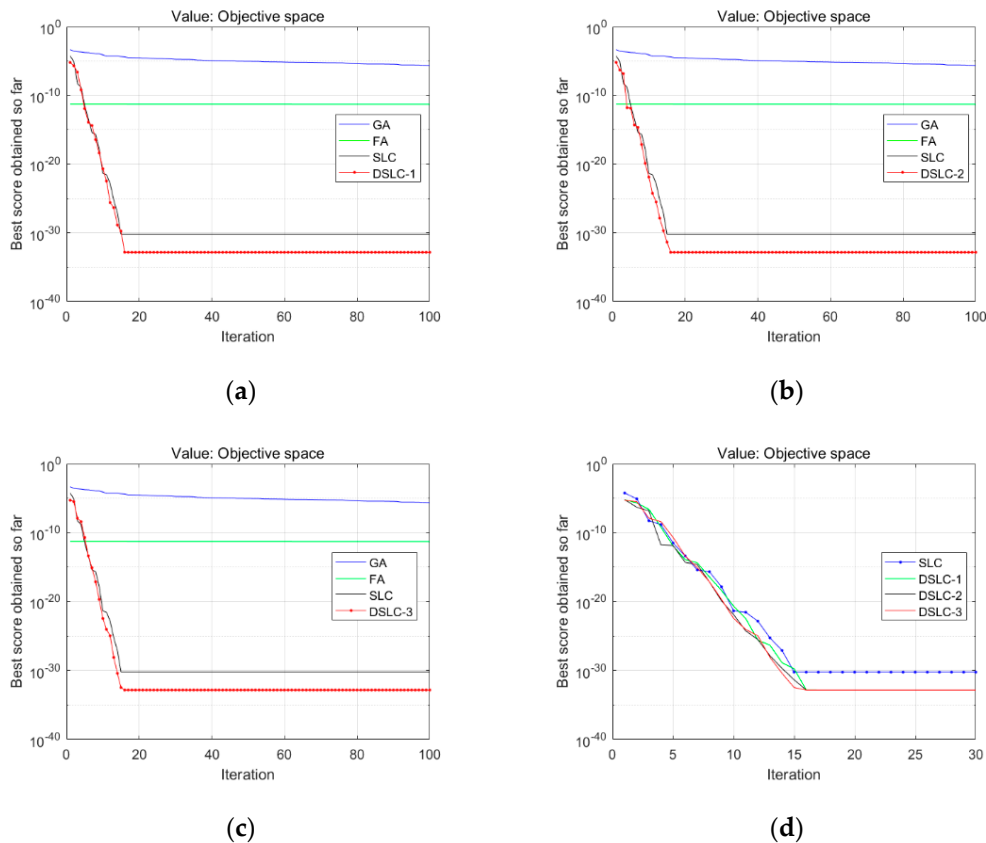
Figure 6. Comparison results of the proposed DSLC-1, 2, and 3 with the SLC, FA, and GA that listed (**a**), (**b**), and (**c**), respectively; (**d**) shows three schemes of strategies with the SLC algorithm for testing function 17.

Clearly, the cases of these testing functions on the proposed schemes of DSLC-1, 2, and 3 all provide comparatively faster convergence than other algorithms. It says the accuracy of the performance on testing functions of the proposed method of the DSLC procedures is improved significantly.

5. Applied DSLC for Deployment Optimization in WSN

WSN deployment still faces many challenges that are related to determining node positions to ensure a trade-off between different metrics. Due to the high density of WSN, finding an optimal deployment becomes an NP-Hard task. This section presents the applied DSLC algorithm to the deployment optimization of WSN based on node coverage technique. The experimental results of the DSLC algorithm are compared with the SLC, GA [40], and FA [41] algorithms for the coverage of node layout in the deployment of WSN.

5.1. Objective Function

The problem of determining the optimal spatial node coverage of implementation WSN is studied to model objective function mathematically. The formulation is figured out based on mathematical modeling of the coverage. In the node coverage problem, we assume sensor nodes as $s_{i,j}$ and the space node as $c_{i,j}$. Sensor coverage radius is r as $cov_{i',j'}$. The solution vector is $MP\{s_{i,j}\}$ of the position of main players in the team in DSLC algorithm for coverage deployed optimization. It means $MP_i\{(x_{i1}, y_{i1}), (x_{i1}, y_{i1}), \ldots, (x_{in}, y_{in})\}$, where (x, y) is the coordinate sensor or its location in the network. Because of the existing environmental surrounding noise in WSN, deployment causes

decreasing accuracy of the coverage area. In practice, the sensor node covering model as referred to in Equation (6) should be modified the probability model as follows.

$$
P(s_{ij}, p) = \begin{cases} 1, & d(s_{ij}, p) \leq r - r_e \\ e^{(\frac{-\omega_1 \alpha_1^{\tau_1}}{\alpha_2^{\tau_2} + \omega_2})}, & r - r_e < (s_{ij}, p) < r + r_e \\ 0, & otherwise \end{cases} \tag{18}
$$

where r_e is a measure of detecting uncertainty; $(r_e < r)$, r is the sensing radius of the node; τ_1, τ_2 and ω_1, ω_2 are parameters which depend on physical attributes of sensors; and $d(s_{ij}, p)$ is Euclidean distance point p to the sensor s_{ij}. Two input parameters are α_1 and α_2 that are presented as follows.

$$
\begin{aligned} \alpha_1 &= r_e - r + d(s_{ij}, p) \\ \alpha_2 &= r_e + r - d(s_{ij}, p) \end{aligned} \tag{19}
$$

The coverage probability ratio of the deploying area is constructed as the objective function of the entire network coverage optimization.

$$
Maximize \; F\{c\} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} P(s_{ij}, p) \times \alpha}{m \times n} \tag{20}
$$

where p is a testing coverage point that has coordinate (x, y), α is an event covered checking point (x, y), and whether covered or not, the node is replaced. The formula of α is expressed as follows:

$$
\alpha(c_{i,j}, s_{i',j'}) = \begin{cases} 1 & if \; d((i, j), (i', j')) < cov_{i',j'} \\ 0 & otherwise \end{cases} \tag{21}
$$

The node coverage in WSN deploying outline is processed as follows. The probability of coverage for each node is computed by applying DSLC for fitness function as in the Equation (20) to point pixels. The pixel points coverage is summarized as in the Equation (8) to join coverage of all pixels in the region for the entire network.

5.2. Parameter Setting

In the experiment, assumed N-nodes WSN was deployed in scattering the area of $M \times M$ randomly. N is a number of nodes that can be set to 100, 200, 300; M is a length measurement of the deployed area that can be 30, 50, 90 meters. The network has a base station (BS) that operates with an unlimited power supply. The characteristics of WSN operation were assumed to behave like scheduling periods of packet transmission time. The parameters are set for the metaheuristic algorithms: the DSLC, SLC, FA, and GA methods are the same condition to solve the same problem of the deployment WSN for comparing fairly. Table 6 lists the initial values for setting the parameters of the experiment.

Table 6. Initial values for setting parameters of the experiment.

Parameters Noticed	Denoted Symbols	Initial Values
Initial node energy	E_j	0.5 J
Receiving and transmitting energy	E_{fs}	10 pJ/bit/m^2
Number of nodes in WSN	N	100/200/300/nodes
Data aggregation energy	E_{DA}	5 pJ/bit/signal
Number bit of a data message	l	1024 bit
Radio electronics energy	E_{elec}	50 nJ/bit
Amplifier energy	E_{mp}	0.013 pJ/bit/m^4
Space distribution	M	30/50/90/300 m
Generations	$MaxIter$	200
Parameters of physical characteristics	τ_1, τ_2 and ω_1, ω_2	1, 0.95 and 0.9, 0.01
Number of runs	$runs$	20
Radius of the sensor reaching	$r; r_e$	3; 1.5 m
GA, FA	Initialize parameters	$N_P = 80$, $F_c = 0.8$, $P_m = 0.1$
DSCL, SCL	Initialize parameters	$N_P = 80$, $R_1 = 10$,

5.3. Experimental Results

Figures 7–9 show the results of the node distributed visually by optimizing the deployment node position layout of the proposed schemes of DSLC-1, 2, 3 and SLC, FA, GA methods, respectively. Observably, in these figures, it can be seen that the results obtained by the proposed schemes of DSLC produce a better node distributed layout of WSN implementation than the other competitors in the comparison. The obtained results are more realistic and more optimal in terms of deployment space shape.
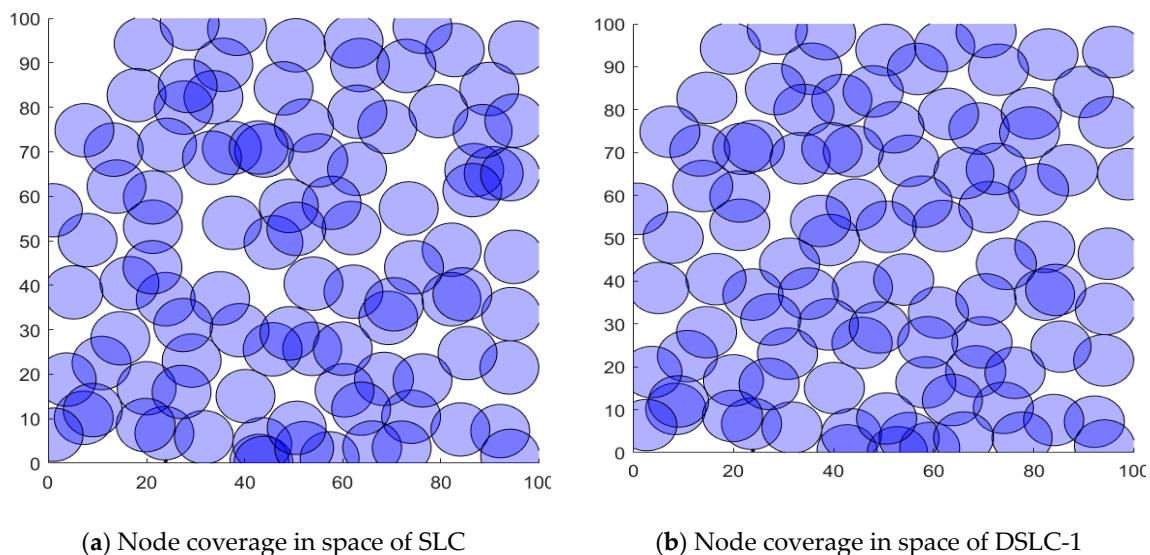

(**a**) Node coverage in space of SLC   (**b**) Node coverage in space of DSLC-1

Figure 7. Node distributed results of the optimized deploying node position layout visually, by the proposed scheme DSLC-1 and the SLC method, respectively.
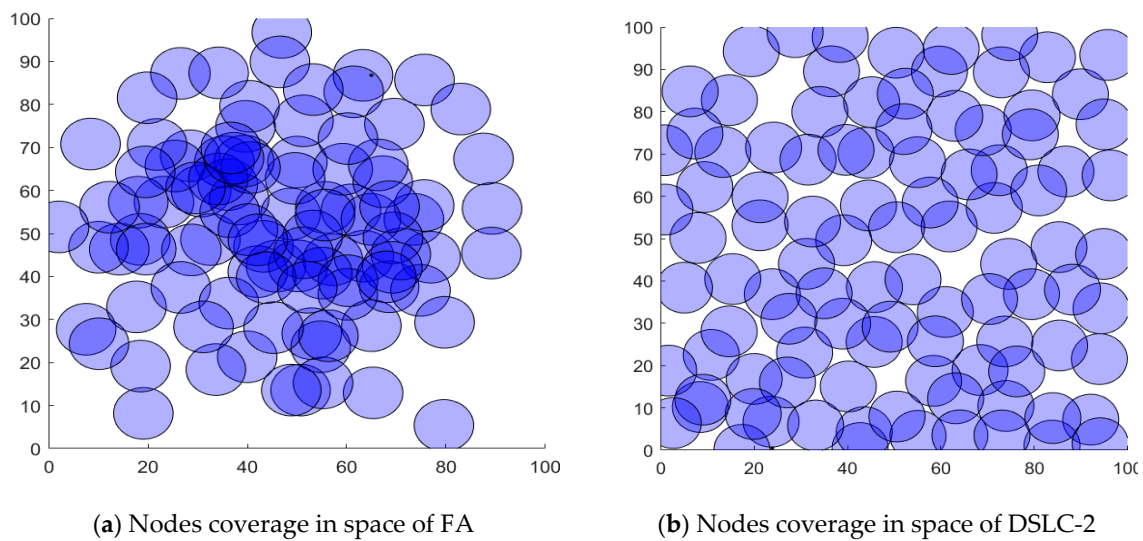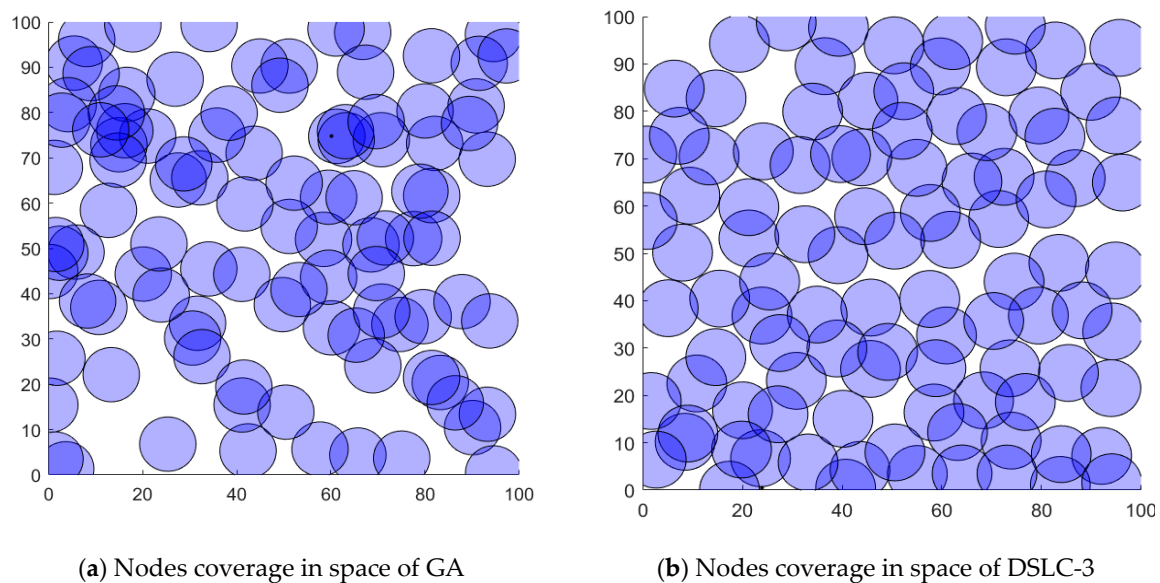
(**a**) Nodes coverage in space of FA

(**b**) Nodes coverage in space of DSLC-2

Figure 8. Node distributed results of the optimized deploying node position layout visually, by the proposed scheme DSLC-2 and the SLC method, respectively.

(**a**) Nodes coverage in space of GA

(**b**) Nodes coverage in space of DSLC-3

Figure 9. Node distributed results of the optimized deploying node position layout visually, by the proposed scheme DSLC-1 and the SLC method, respectively.

Table 7 indicates the Comparison of the coverage optimization performance of the proposed schemes of DSLC with SLC, FA, and GA for different deployed regions. As can be seen in Table 7, it can be said that the proposed approach provides an optimal solution to the deployment of WSN in that the nodes are included in connectivity evaluation and fully connected. In the proposed strategies, the combining strategy DSLC-3 offers more diverse groups than the other policies in the study, and it produces a better performance.
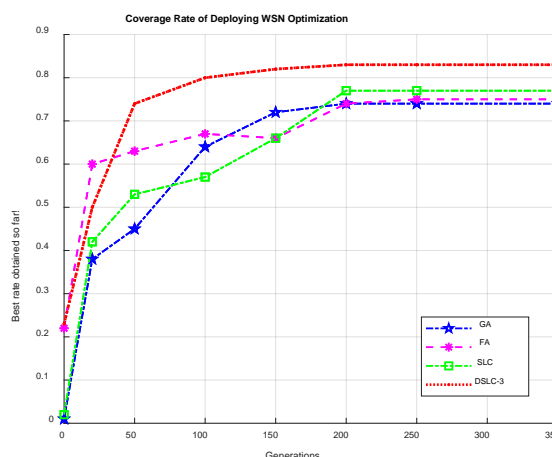
Figure 10 shows the comparison of the obtained coverage rate of the proposed DSLC-3 with the SLC, FA, and GA methods for the deployment WSN optimization. It can be seen from the figure that the proposed scheme of DSLC-3 can cover the entire deploying area with the best layout of the nodes. It can be concluded from the experimental results that the effect of the DSLC achieves the optimal global solution and generally produces better than the SLC, FA, and GA algorithms.

Table 7. Comparison of the coverage optimization performance of the proposed schemes of DSLC with SLC, FA, and GA for different deployed regions.

Deployed Area (m^2)	Moving Nodes	DSLC-1	DSLC-2	DSLC-3	SLC	FA	GA
40×40	20	83.49%	84.78%	86.12%	79.34%	69.11%	74.01%
70×70	30	82.21%	84.06%	85.10%	78.33%	71.33%	73.01%
90×90	50	84.19%	85.18%	87.13%	81.94%	72.93%	78.01%

Figure 10. Comparison of the obtained coverage rate of the proposed DSLC-3 with the SLC. FA, and GA methods for the deployment WSN optimization.

6. Conclusions

In this paper, we proposed an enhanced diversity team soccer league competition (namely DSLC) algorithm based on updating strategies of league team members for global optimization and deployment wireless sensor network (WSN) problems. The proposed approach aims to enhance the exploration of algorithm searching to avoid the local optimum trap in solving complex issues such as the deployment issue of WSN. The updating team strategies consist of trading, drafting, and combining schemes in the soccer league. We used twenty-one selected benchmark functions of CEC2017 and the node coverage problem of WSN deployment to test the performance of the proposed algorithm. The experimental results of the proposed method were compared with the other algorithms in the literature, e.g., soccer league competition (SLC), firefly algorithm (FA), genetic algorithm (GA), and adaptive differential evolution (JADE). The compared results show that the proposed DSLC algorithm provides increased accuracy optimization in dealing with complex optimization problems. In future work, the problem of optimal spatial node location of WSN in the indoor environment will be considered to deal with multi-objective optimization function that combines coverage, connection rate, node energy, and cost by applying the proposed DSLC algorithm.

Author Contributions: Conceptualization, S.-C.C. and J.-S.P.; formal analysis, T.-T.N.; methodology, Y.Q.; software, T.-T.N.; supervision, T.-K.D. and J.-S.P.; writing—original draft Y.Q.; writing —review and editing, K.D. All authors have read and agreed to the published version of the manuscript.

References

1. Hussain, K.; Mohd Salleh, M.N.; Cheng, S.; Shi, Y. Metaheuristic research: A comprehensive survey. *Artif. Intell. Rev.* **2019**, *52*, 219102233. [CrossRef]

2. Sedigheh, M.; Shiri, M.E. Metaheuristics in large-scale global continues optimization: A survey. *Inf. Sci.* **2015**, *295*, 407–428.

3. Nguyen, T.-T.; Pan, J.-S.; Dao, T.-K. A Compact Bat Algorithm for Unequal Clustering in Wireless Sensor Networks. *Appl. Sci.* **2019**, *9*, 1973. [CrossRef]

4. Du, Z.-G.; Pan, J.-S.; Chu, S.-C.; Luo, H.-J.; Hu, P. QUasi-Affine TRansformation Evolutionary Algorithm with Communication Schemes for Application of RSSI in Wireless Sensor Networks. *IEEE Access* **2020**, *8*, 8583–8594. [CrossRef]

5. Shaheen, A.M.; Spea, S.R.; Farrag, S.M.; Abido, M.A. A review of meta-heuristic algorithms for reactive power planning problem. *Ain Shams Eng. J.* **2018**, *9*, 215–231. [CrossRef]

6. Shanmugasundaram, G.; Thilagavathi, N.; Ramya, S.; Kanimozhi, K. An Investigation of Meta Heuristic Algorithms Applied on Capacitated Vehicle Routing Problem. In Proceedings of the 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 29–30 March 2019; pp. 1–6.

7. Souier, M.; Sari, Z.; Hassam, A. Real-time rescheduling metaheuristic algorithms applied to FMS with routing flexibility. *Int. J. Adv. Manuf. Technol.* **2013**, *64*, 145–164. [CrossRef]

8. Chu, S.-C.; Du, Z.-G.; Pan, J.-S. Symbiotic Organism Search Algorithm with Multi-Group Quantum-Behavior Communication Scheme Applied in Wireless Sensor Networks. *Appl. Sci.* **2020**, *10*, 930. [CrossRef]

9. Tian, A.Q.; Chu, S.C.; Pan, J.S.; Cui, H.; Zheng, W.M. A Compact Pigeon-Inspired Optimization for Maximum Short-Term Generation Mode in Cascade Hydroelectric Power Station. *Sustainability* **2020**, *12*, 767. [CrossRef]

10. Sörensen, K.; Sevaux, M.; Glover, F. A history of metaheuristics. In *Handbook of Heuristics*; Springer: Berlin, Germany, 2018; ISBN 9783319071244.

11. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1975; ISBN 978-0-262-08213-6.

12. Yang, X.-S. *Firefly Algorithms for Multimodal Optimization BT—Stochastic Algorithms: Foundations and Applications*; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin, Heidelberg, 2009; pp. 169–178.

13. Zhang, J.; Sanderson, A.C. JADE: Adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **2009**, *13*, 945–958. [CrossRef]

14. Chu, S.A.; Tsai, P.W.; Pan, J.S. Cat swarm optimization. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, Guilin, China, 7–11 August 2006.

15. Elbeltagi, E.; Hegazy, T.; Grierson, D. Comparison among five evolutionary-based optimization algorithms. *Adv. Eng. Informatics* **2005**, *19*, 43–53. [CrossRef]

16. Nguyen, T.-T.; Dao, T.-K.; Kao, H.-Y.; Horng, M.-F.; Shieh, C.-S. Hybrid Particle Swarm Optimization with Artificial Bee Colony Optimization for Topology Control Scheme in Wireless Sensor Networks. *J. Internet Technol.* **2017**, *18*, 743–752.

17. Dao, T.K.; Pan, T.S.; Nguyen, T.T.; Pan, J.S. Parallel bat algorithm for optimizing makespan in job shop scheduling problems. *J. Intell. Manuf.* **2018**, *29*, 451–462. [CrossRef]

18. Naser Moosavian, B.K.R. Soccer League Competition Algorithm, a New Method for Solving Systems of Nonlinear Equations. *Int. J. Intell. Sci.* **2014**, *4*, 7–16. [CrossRef]

19. Moosavian, N.; Roodsari, B.K. Soccer League Competition Algorithm: A Novel Meta-heuristic Algorithm For Optimal Design of Water Distribution Networks. *Swarm Evol. Comput.* **2014**, *17*, 14–24. [CrossRef]

20. Chagwiza, G.; Jaison, A.; Masamha, T. Parameter Improvement of the Soccer League Competition Algorithm by Introducing Stubborn Players: Application to Water Distribution Network. *Math. Probl. Eng.* **2016**, *2016*, 3425374. [CrossRef]

21. Hedieh Sajedi, S.K. Cognitive Soccer League Competition algorithm for solving knapsack problems. In Proceedings of the 1ST International Conference on Advances Research on Electrical and Computer Engineering, Tehran, Iran, 13 May 2016; p. 10240.

22. Jaramillo, A.; Gómez, A.; Mansilla, S.; Salas, J.; Crawford, B.; Soto, R.; Olguín, E. Using the Soccer League Competition algorithm to solve the set covering problem. In Proceedings of the 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), Gran Canaria, Spain, 15–18 June 2016; pp. 1–4.

23. O'Donovan, T.; O'Donoghue, J.; Sreenan, C.; Sammon, D.; O'Reilly, P.; O'Connor, K.A. A context aware wireless Body Area Network (BAN). In Proceedings of the 2009 3rd International Conference on Pervasive Computing Technologies for Healthcare—Pervasive Health 2009, London, UK, 1–3 April 2009.

24. Anastasi, G.; Farruggia, O.; Re, G.L.; Ortolani, M. Monitoring High-Quality Wine Production using Wireless Sensor Networks. In Proceedings of the 2009 42nd Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 5–8 January 2009; pp. 1–7.

25. Akyildiz, I.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Networks* **2002**, *38*, 393–422. [CrossRef]

26. Chakrabarty, K.; Iyengar, S.S.; Qi, H.; Cho, E. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans. Comput.* **2002**, *51*, 1448–1453. [CrossRef]

27. Pan, J.-S.; Nguyen, T.-T.; Dao, T.-K.; Pan, T.-S.; Chu, S.-C. Clustering Formation in Wireless Sensor Networks: A Survey. *J. Netw. Intell.* **2017**, *02*, 287–309.

28. Nguyen, T.-T.; Pan, J.-S.; Wu, T.-Y.; Dao, T.-K.; Nguyen, T.-D. Node Coverage Optimization Strategy Based on Ions Motion Optimization. *J. Netw. Intell.* **2019**, *4*, 1–9.

29. García-hernández, C.F.; Ibargüengoytia-gonzález, P.H.; García-hernández, J.; Pérez-díaz, J. A Wireless Sensor Networks and Applications: A Survey. *J. Comput. Sci.* **2007**, *7*, 264–273.

30. Nguyen, T.-T.; Pan, J.-S.; Lin, J.C.-W.; Dao, T.-K.; Nguyen, T.-X.-H. An Optimal Node Coverage in Wireless Sensor Network Based on Whale Optimization Algorithm. *Data Sci. Pattern Recognit.* **2018**, *2*, 11–21.

31. Fei, Z.; Li, B.; Yang, S.; Xing, C.; Chen, H.; Hanzo, L. A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms, and Open Problems. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 550–586. [CrossRef]

32. Jourdan, D.B.; Roy, N. Optimal sensor placement for agent localization. *ACM Trans. Sens. Networks* **2008**, *4*, 1–40. [CrossRef]

33. Chen, Y.; Chuah, C.-N.; Zhao, Q. Sensor placement for maximizing lifetime per unit cost in wireless sensor networks. In Proceedings of the MILCOM 2005—2005 IEEE Military Communications Conference, Atlantic City, NJ, USA, 17–20 October 2005; pp. 1097–1102.

34. Nguyen, T.-T.; Pan, J.-S.; Dao, T.-K. A Novel Improved Bat Algorithm Based on Hybrid Parallel and Compact for Balancing an Energy Consumption Problem. *Information* **2019**, *10*, 194. [CrossRef]

35. Nguyen, T.T.; Pan, J.S.; Dao, T.K. An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network. *IEEE Access* **2019**, *7*, 75985–75998. [CrossRef]

36. Nguyen, T.-T.; Dao, T.-K.; Horng, M.-F.; Shieh, C.-S. An Energy-based Cluster Head Selection Algorithm to Support Long-lifetime in Wireless Sensor Networks. *J. Netw. Intell.* **2016**, *1*, 23–37.

37. Dao, T.K.; Pan, T.S.; Nguyen, T.T.; Chu, S.C. A compact Articial bee colony optimization for topology control scheme in wireless sensor networks. *J. Inf. Hiding Multimed. Signal Process.* **2015**, *6*, 297–310.

38. Liang, J.J.; Qu, B.Y.; Suganthan, P.N. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. In *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*; Technical Report: Zhengzhou, China, 2013; p. 635.

39. Wu, G.; Mallipeddi, R.; Suganthan, P.N. Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. In *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*; Technical Report: Zhengzhou, China, 2017.

40. Kalayci, T.E.; Uğur, A. Genetic algorithm–based sensor deployment with area priority. *Cybern. Syst.* **2011**, *42*, 605–620. [CrossRef]

41. Tuba, E.; Tuba, M.; Beko, M. Mobile wireless sensor networks coverage maximization by firefly algorithm. In Proceedings of the 2017 27th International Conference Radioelektronika (RADIOELEKTRONIKA), Brno, Czech Republic, 19–20 April 2017; pp. 1–5.