Swarm Intelligence Algorithm for Microwave Filter Optimization

Erredir Chahrazad[(✉)], Emir Bouarroudj, and Mohamed Lahdi Riabi

Laboratory of Electromagnetic and Telecommunication,
University Brothers Mentouri Constantine 1, Constantine, Algeria
cerredir@yahoo.fr

Abstract. In this paper, three recent swarm intelligence algorithms (spider monkey optimization (SMO), social spider optimization (SSO) and teaching learning based optimization (TLBO)) are proposed to the optimization of microwave filter (H-plane three-cavity filter). The results of convergence and optimization use of these algorithms are compared with the results of the most popular swarm intelligences algorithm, namely particle swarm optimization (PSO) for different common parameters (population size and maximum number of iteration). The results showed validation of the proposed algorithms.

Keywords: Swarm intelligence · Spider monkey optimization
Social spider optimization · Teaching learning based optimization
Particle swarm optimization · Microwave filter

1 Introduction

Swarm intelligence (SI) algorithms are natural inspired techniques that involve the study of collective behavior of decentralized, self-organized systems [1, 2]. The swarm intelligence systems contain a set of particles (or agents) that interact locally with one another and with their environment. Swarm Intelligence techniques can be used in several engineering applications where the SI algorithms have been successfully applied to solve complex optimization problems including continuous optimization, constrained optimization and combinatorial optimization.

To date, several swarm intelligence models based on different natural swarm systems have been proposed in the literature, and successfully applied in many real-life applications. Examples of swarm intelligence models are: Ant Colony Optimization [3], Particle Swarm Optimization [4], Artificial Bee Colony [5], Bacterial Foraging [6], Cat Swarm Optimization [7], Artificial Immune System [8], and Glowworm Swarm Optimization [9].

In this paper, we will primarily focus on three of the recent swarm intelligences models, namely, spider monkey optimization (SMO), social spider optimization (SSO) and teaching learning based optimization (TLBO) to optimize the waveguide microwave filter (H-plane three-cavity filter). The results of optimization obtained are validated by comparing them with those obtained using the optimization algorithm available in the literature Particle swarm optimization (PSO). The details of each algorithm are presented in the next section.

2   Swarm Intelligence Algorithm

2.1   Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic method of optimization based on the reproduction of a social behavior. It was invented by Eberhart and Kennedy [4] in 1995. They tried to simulate the ability of animal societies that don't have any leader in their group or swarm (bird flocking and fish schooling) to move synchronously and their ability to change direction suddenly while remaining in an optimal formation (food source). PSO consists of a swarm of particles, where particle represent a potential solution. The particles of the swarm fly through hyperspace and have two essential reasoning capabilities: their memory of their own best position local best (LB) and knowledge of the global or their neighborhood's best global best (GB). The essential steps of particle swarm optimization are presented by the following algorithm:

Step 1. Initialize the optimization parameters (population size, number of generations, design variables of the optimization problem and the specific parameters of algorithm) and define the optimization problem (minimization or maximization of fitness function).

Step 2. Generate a random population (position and velocities), according to the population size and the limits of the design variables.

Step 3. Evaluate each initialized particle's fitness value, then calculate LB the positions of the current particles, and GB the best position of the particles.

Step 4. the best particle of the current particles is stored. The positions and velocities of all the particles are updated according to (1) and (2), then a group of new particles are generated.

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{1}$$

$$V_i(t+1) = w * V_i(t) + c_1 r_1 [LB_i(t) - X_i(t)] + c_2 r_2 [GB(t) - X_i(t)] \tag{2}$$

$V_i(t)$, $X_i(t)$ are the velocity and the position for particle i at time t. w is the inertia weight, at each iteration update with the following equation [10].

$$w(t) = w_{max} - \frac{(w_{max} - w_{max}) * t}{maxit} \tag{3}$$

The parameters w_{max}, w_{min}, $maxit$, c_1 and c_2 are constant coefficients determined by the user. r_1 and r_2 are random numbers between 0 and 1.

Step 5. Repeat the procedure from step 3 until the maximal iteration is met.

2.2   Spider Monkey Optimization Algorithm

Spider Monkey Optimization (SMO) algorithm is a new swarm intelligence algorithm based on the foraging behavior of spider monkeys, proposed by Bansal et al. in [11]. There are four important control parameters necessary for this algorithm: perturbation

rate (Pr), local leader limit (LLL), global leader limit (GLL) and maximum number of groups (MG). The SMO process consists of six phases:

Local Leader Phase (LLP). In this phase, Spider Monkey SM of each group updates its position based on the experience of the local leader and local group members as following expression:

$$SM_{ij} = SM_{ij} + r_1 \left(LL_{Kj} - SM_{ij}\right) + r_2 \left(SM_{rj} - SM_{ij}\right) \tag{4}$$

Where SM_{ij} is the jth dimension of the ith SM, LL_{kj} represents the jth dimension of the kth local group leader position. SM_{rj} is the jth dimension of the rth SM which is chosen randomly within k^{th} group such that $r \neq i$, r_1 is a random number between 0 and 1 and r_2 is a random number between -1 and 1.

Global Leader Phase (GLP). In GLP phase, all the SM's update their position using experience of global leader and local group member's experience. The position update equation for this phase is as follows:

$$SM_{ij} = SM_{ij} + r_1 \left(GL_j - SM_{ij}\right) + r_2 \left(SM_{rj} - SM_{ij}\right) \tag{5}$$

Where GL_j represents the jth dimension of the global leader position and $j \in \{1, 2 \ldots D\}$ is the randomly chosen index; D is the number of design variables. The positions are updated based upon some probability given by the following formula.

$$P_i = 0.9 \frac{F_i}{max\_F} + 0.1 \tag{6}$$

Where P_i is the probability, F_i is the fitness of ith SM, and max_F is the maximum fitness of the group.

Local Leader Learning Phase (LLLP). In this phase, the position of the local leader is updated by applying the greedy selection in that group. If the LL's position remains same as before, then the Local Limit Count is increased by 1.

Global Leader Learning Phase (GLLP). In this phase, the position of the global leader is updated by applying the greedy selection in the population. If the GL's position remains same as before, then the Global Limit Count is increased by 1.

Local Leader Decision Phase (LLDP). If a LL position is not updated for a predetermined number of iterations Local Leader Limit (LLL), then the positions of the spider monkeys are updated either by random initialization or by using information from both LL and GL through Eq. (7) based on the perturbation rate (pr)

$$SM_{ij} = SM_{ij} + r_1 \left(GL_j - SM_{ij}\right) + r_1 \left(SM_{ij} - LL_{Kj}\right) \tag{7}$$

Global Leader Decision Phase (GLDP). If the position of GL is not updated in predetermined number of iterations Global Leader Limit, then the population is split into subgroups. The groups are split till the number of groups reaches to maximum allowed groups (MG), then they are combined to form a single group again.

The details of each step of SMO implementation are explained below:

Step 1. Initialize the optimization parameters and define the optimization problem. Control parameters necessary for this phases; perturbation rate (Pr), local leader limit (LLL), global leader limit (GLL) and maximum number of groups (MG). Some settings of control parameters are suggested as follows:

– MG = N/10, i.e., it is chosen such that minimum number of SM's in a group should be 10.
– Global Leader Limit $\in$ [N/2, 2 × N].
– Local Leader Limit should be D × N.

$$\text{Pr}(t+1) = \text{Pr}(t) + \frac{(0.4 - 0.1)}{maxit}, \quad \text{Pr}(1) = 0.1 \tag{8}$$

Step 2. Initialize the population and evaluate the corresponding objective function value.

Step 3. Locate global and local leaders.

Step 4. The local leader phase starts by update the position of all group members' Eq. (4). Accept of a new solution if it gives better function value. All the accepted function values at the end of this phase are maintained and these values become the input to the global leader phase.

Step 5. Produce new positions for all the group members, selected by probability (P_i), by using self experience, global leader experience and group members' experiences Eq. (5).

Step 6. Update the position of local and global leaders, by applying the greedy selection process on all the groups (see. LLLP, GLLP).

Step 7. If any Local group leader is not updating her position after a specified number of times (Local Leader Limit) then redirect all members of that particular group for foraging (LLDP).

Step 8. If Global Leader is not updating her position for a specified number of times (Global Leader Limit) then she divides the group into smaller groups (see. GLDP).

Step 9. Repeat the procedure from step 3 to until the termination criterion is met.

2.3   Social Spider Optimization Algorithm

Social-spiders optimization (SSO) [12, 13] is a new proposed swarm optimization algorithm; it is based on the natural spider's colony behavior. An interesting characteristic of social-spiders is the highly female-biased populations, where the number of females N_f is randomly selected within the range of 65–90% of the entire population NP and the rest is the number of male Nm. Therefore, N_f and N_m are calculated by the following equations:

$$N_f = floor((0.9 - rand * 0.25) * NP) \tag{9}$$

$$N_m = NP - N_f \tag{10}$$

Where *floor* rounds each element to the nearest integer, and *rand* is a random number in the unitary range [0, 1].

   After the initialization process the algorithm starts the searching loop that only ends when the maximum number of function evaluations or the target function value is reached. The first step in the searching loop is to calculate the spider's weight. This calculation is done according to:

$$W_i = \frac{(Worst - f(x_i))}{(Worst - Best)} \tag{11}$$

Where W_i is the weight of the ith spider, $f(x_i)$ is the fitness value of the spider x_i. The values *Worst* and *Best* are defined as follows (considering a minimization problem):

$$Best = min_{i=1...NP} f(x_i) \tag{12}$$

$$Worst = max_{i=1...NP} f(x_i) \tag{13}$$

In the colony, the spiders communicate with each other directly by mating or indirect by a small vibration to determine the potential direction of a food source, this vibration depend on the weight and distance of the spider which has generated them.

$$Vib_{ij} = w_j * \exp\left(-(d_{ij})^2\right) \tag{14}$$

Where w_j indicates the weight of the jth spider, and d_{ij} is the euclidean distance between i^{th} and j^{th} spiders. Every spider is able to consider three vibrations from other spiders as:

- Vibrations Vib_{ci} are perceived by the individual i (X_i) as a result of the information transmitted by the member c (X_c) who is an individual that has two important characteristics: it is the nearest member to i and possesses a higher weight in comparison to i ($W_c > W_i$).
- The vibrations Vib_{bi} perceived by the individual i as a result of the information transmitted by the member b (X_b), with b being the individual holding the best weight (best fitness value) of the entire population *NP*, such that:

$$Wb = max_{k=1...N}(W_k). \tag{15}$$

- The vibrations Vib_{fi} perceived by the individual i (X_i) as a result of the information transmitted by the member f (X_f), with f being the nearest female individual to i.

   Depending on gender, each individual is updating their position according to three operations (female operation, male operation and mating operation). In female operation, the female individuals are updating as follow equation:

$$X_i = X_i + \alpha \, Vib_{ci}(X_c - X_i) + \beta \, Vib_{bi}(X_b - X_i)$$
$$+ \delta(rand - 0.5) \text{ with probability } PF \tag{16}$$

$$X_i = X_i + \alpha \, Vib_{ci}(X_c - X_i) + \beta \, Vib_{bi}(X_b - X_i) \\ + \delta(rand - 0.5) \text{ with probability } 1 - PF \tag{17}$$

Where PF is threshold parameter, α, β, δ and *rand* are random numbers between [0, 1].

The male spiders are divided into two different groups (dominant members D and non-dominant members ND) according to their position with regard to the median member. According to this, change of positions for the male spider can be modeled as follows:

$$X_i = X_i + \alpha \left(\frac{\sum_{h=1}^{N_m} X_h W_{N_f + h}}{\sum_{h=1}^{N_m} W_{N_f + h}} - X_i \right), \text{ Male D} \tag{18}$$

$$Xi = Xi + \alpha \, Vibfi(Xf - Xi) + \delta.(rand - 0.5), \text{ Male ND} \tag{19}$$

Where the individual (X_f) represent the nearest female individual to the male member.

After all males and females spiders are update, the last operator is representing the mating behavior where only dominant males will participate with females who are within a certain radius called mating radius given by

$$R = \frac{\sum_{d=1...n}(X_d^h - X_d^l)}{2D} \tag{20}$$

Where X^h and X^l are respectively the upper and lower bound for a given dimension and n is the problem dimension. Males and females which are under the mating radius generate new candidate spiders according to the roulette method. Each candidate spider is evaluated in the objective function and the result is tested against all the actual population members. If any member is worse than a new candidate, the new candidate will take the actual individual position assuming actual individual's gender.

2.4 Teaching Learning Based Optimization

Rao et al. [14–16] proposed an algorithm, called Teaching-Learning-Based Optimization (TLBO), based on the traditional teaching learning phenomenon of a classroom. TLBO is a population based algorithm, where a group of students (i.e. learner) is considered as population and the different subjects offered to the learners are analogous with the different design variables of the optimization problem. The results of the learner are analogous to the fitness value of the optimization problem. The best solution in the entire population is considered as the teacher. Teacher and learners are the two vital components of the algorithm, so there are two modes of learning; through the teacher (known as the teacher phase) and interacting with other learners (known as the learner phase).

Teacher Phase. In this part, learners take their knowledge directly through the teacher, where a teacher tries to increase the mean result value of the classroom to another value, which is better than, depending on his or her capability. This follows a random process depending on many factors. In this work, the value of solution is represented as

$X_{j,k,i}$, where j means the jth design variable (i.e. subject taken by the learners), $j = 1, 2,$ …, m; k represents the kth population member (i.e. learner), $k = 1, 2, …, N$; and i represents the ith iteration, $i = 1, 2, …, maxit$, where $maxit$ is the number of maximum generations (iterations). The existing solution is updated according to the following expression

$$X'_{j,k,i} = X_{j,k,i} + DM_{j,k,i} \tag{21}$$

$DM_{j,k,i}$ the difference between the existing mean and the new mean of each subject is given by

$$DM_{j,k,i} = r * \left(X_{j,kbest,i} - TF * M_{j,i} \right) \tag{22}$$

$M_{j,i}$ the mean result of the learners in a particular subject j, $X_{j,kbest,i}$ the new mean and is the result of the best learner (i.e. teacher) in subject j. r is the random number in the range [0, 1]. TF The teaching factor is generated randomly during the algorithm in the range of [1, 2], in which 1 corresponds to no increase in the knowledge level and 2 corresponds to complete transfer of knowledge. The in between values indicates the amount of transfer level of knowledge. The value of TF is not given as an input to the algorithm its value is randomly decided by the algorithm

$$TF = round[1 + rand(0,1)\{2 - 1\}] \tag{23}$$

Learner Phase. In this part, learners increase their knowledge by interaction among themselves. A learner interacts randomly with other learners for enhancing his or her knowledge. A learner learns new things if the other learner has more knowledge than him or her. At any iteration i, each learner is compared with the other learners randomly. For comparison, randomly select two learners P and Q such that $X'_{P,i} \neq X'_{Q,i}$ (where $X'_{P,i}$ and $X'_{Q,i}$ are the updated values at the end of the teacher phase).

$$X''_{j,P,i} = X'_{j,P,i} + r * \left(X'_{j,P,i} + X'_{j,Q,i} \right), f\left(X'_{P,i} \right) < f\left(X'_{Q,i} \right) \tag{24}$$

$$X''_{j,P,i} = X'_{j,P,i} + r * \left(X'_{j,Q,i} + X'_{j,P,i} \right), f\left(X'_{P,i} \right) > f\left(X'_{Q,i} \right) \tag{25}$$

Accept $X''_{j,P,i}$ if it gives a better function value.

3   Comparison of Optimization Techniques

The swarm intelligence algorithms have been widely used to solve complex optimization problems. These methods are more powerful than conventional methods based on formal logic or mathematical programming. In terms of comparison of intelligence algorithms of the swarm we have:

- The four algorithms studied in this paper are population-based techniques that implement a group of solutions to achieve the optimal solution.
- The PSO and TLBO algorithms use the best solution of the iteration to change the existing solution in the population, which increases the rate of convergence.
- TLBO and PSO do not divide the population unlike SSO and SMO.
- TLBO, SSO and SMO implement greed to accept the right solution.
- Each method requires parameters that affect the performance of the algorithm.
  - PSO requires coefficients of confidence and inertia.
  - SSO requires the threshold setting.
  - SMO requires the perturbation rate (Pr), the local leader limit (LLL), the global leader limit (LGL) and the maximum number of groups (MG).
  - In contrary, TLBO does not require any parameters, which simplifies the implementation of TLBO.

4 Application Example and Results

In this section, the application of a proposed algorithm is presented for the optimization of rectangular waveguide H-plane three-cavity filter [17] Fig. 1. When the main guide is WR28, four parameters are to be optimized W_1 and W_2 (the opening of the iris), l_1 and l_2 (the distance between the iris). The thicknesses of the iris are fixed to $t_1 = 1.45$ mm, $t_2 = 1.1$ mm. Table 1 contains the geometric variables of the structure and the corresponding ranges. As for the frequency range, it was chosen to be f ∈ (34, 35.5 GHz).

The objective is to minimize the fitness function in the frequency range, where the fitness functions is the mean value of the coefficient of reflection S11.

$$fitness = \frac{\sum_{f_1}^{f_2} S_{11}(f)}{PT} \qquad (26)$$

With PT is the number of points in the interval $[f_1, f_2]$.
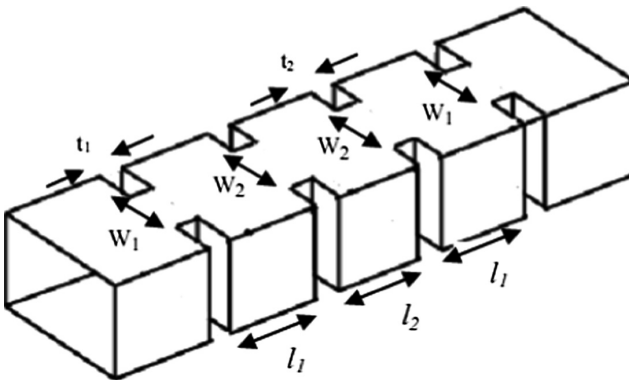

Fig. 1. Rectangular waveguide H-plane three-cavity filter

Table 1. Geometric variables of the structure and the corresponding ranges

Variables	Min	Max
W_1(mm)	1.80	5.40
W_2(mm)	1.20	3.60
l_1(mm)	2.07	6.22
l_2(mm)	2.35	7.05

The convergence of the fitness functions of each algorithm (Best, Worst, Mean) is presented in Table 2 with the population size of 50 and the maximum number of iterations take the following values (30, 50, 300). Table 3 shows the convergence of the fitness functions for the number of iterations is 50 and the population size (30, 70 and 100). Every algorithm is run 10 independent times. The other specific parameters of algorithms are given below:

**Table 2.** The fitness functions for population size = 50

Maxit	Algorithm	Best	Worst	Mean
30	PSO	0.0193	0.1588	0.0907
	SSO	0.0191	0.1405	0.0680
	TLBO	0.0082	0.0100	0.0089
	SMO	0.0082	0.0092	0.0088
50	PSO	0.0126	0.0945	0.0791
	SSO	0.0110	0.0353	0.0190
	TLBO	0.0081	0.0089	0.0082
	SMO	0.0081	0.0081	0.0081
300	PSO	0.0084	0.0210	0.0108
	SSO	0.0081	0.0121	0.0091
	TLBO	0.0081	0.0081	0.0081
	SMO	0.0081	0.0081	0.0081

- PSO Settings: c_1 and c_2 are constant coefficients $c_1 = c_2 = 2$, the inertia weight decreased linearly from 0.9 to 0.2.
- SSO Settings: the threshold parameter PF = 0.7.
- TLBO Settings: for TLBO there is no such constant to set.
- SMO Settings: the parameter of SMO depends on the population size where: (N = 30: MG = 3, GLL = 30 and LLL = 120); (N = 70: MG = 7, GLL = 70 and LLL = 280); (N = 100: MG = 10, GLL = 100 and LLL = 400).

Figure 2 shows the convergence of the fitness function of the best individual of each algorithm. The results of the optimization are presented in the Table 4.

Table 3. The fitness functions for Maxit = 50

Population	Algorithm	Best	Worst	Mean
30	PSO	0.0197	0.0849	0.0411
	SSO	0.0120	0.0536	0.0409
	TLBO	0.0081	0.0140	0.0089
	SMO	0.0081	0.0081	0.0081
70	PSO	0.0126	0.0634	0.0326
	SSO	0.0119	0.0501	0.0272
	TLBO	0.0081	0.0082	0.0081
	SMO	0.0081	0.0081	0.0081
100	PSO	0.0120	0.0352	0.0255
	SSO	0.0103	0.0311	0.0162
	TLBO	0.0081	0.0081	0.0081
	SMO	0.0081	0.0081	0.0081
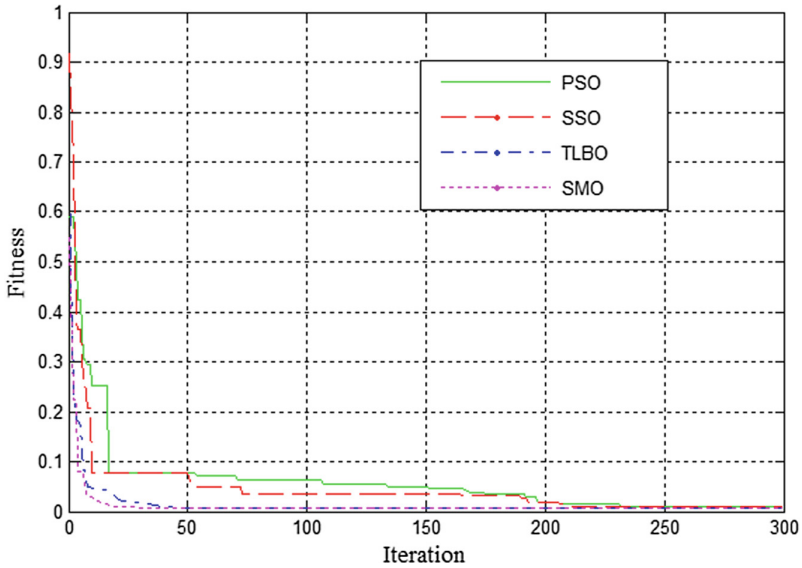
Fig. 2. The convergences of the fitness function of the best individual of each algorithm.

Table 4. The geometrical parameters optimized

Variables	[17]	PSO	SSO	TLBO	SMO
W_1(mm)	3.60	4.705	4.694	4.709	4.078
W_2(mm)	2.40	3.598	3.585	3.600	3.599
l_1(mm)	4.15	2.936	2.917	2.915	2.910
l_2(mm)	4.70	3.633	3.613	3.613	3.608

It is observed from Tables 2, 3 and Fig. 2 that, the SMO and TLBO algorithms performs better in terms of convergence than the PSO and SSO algorithms. In which SMO and TLBO algorithms converge to the minimum optimal for the first's iterations maxit = 30, on the other hand PSO and SSO algorithms converge to the minimum optimal in 300 iterations.

5   Conclusion

In this works, the study and application of the three recent swarm intelligence algorithms in literature (spider monkey optimization (SMO), social spider optimization (SSO) and teaching learning based optimization (TLBO)) are presented. These three algorithms are used for the optimization of microwave filter (H-plane three-cavity filter). The results of convergence and optimization are compared with the results of the most popular swarm intelligences algorithm, particle swarm optimization (PSO). The results showed validation of the proposed algorithms.

References

1. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press, Beckington (2008)
2. Lim, C.P., Jain, L.C., Dehuri, S.: Innovations in Swarm Intelligence. Springer, Berlin (2009). https://doi.org/10.1007/978-3-642-04225-6
3. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
4. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948 (1995)
5. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
6. Passino, K.M.: Biomimicry of bacteria foraging for distributed optimization and control. IEEE Control Syst. Mag. 22(3), 52–67 (2002)
7. Chu, S.-C., Tsai, P.-W., Pan, J.-S.: Cat swarm optimization. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 854–858. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-36668-3_94
8. Bakhouya, M., Gaber, J.: An immune inspired-based optimization algorithm: application to the Traveling Salesman problem. Adv. Model. Optim. 9(1), 105–116 (2007)
9. Krishnanand, K.N., Ghose, D.: Glowworm swarm optimization for searching higher dimensional spaces. In: Lim, C.P., Jain, L.C., Dehuri, S. (eds.) Innovations in Swarm Intelligence. SCI, vol. 248, pp. 61–75. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04225-6_4
10. Singh, N., Singh, S.B.: A new modified approach of mean particle swarm optimization algorithm. In: IEEE 5th International Conference on Computational Intelligence and Communication Networks, Mathura, India, pp. 296–300 (2013)
11. Bansal, J.C., Sharma, H., Jadon, S.S., Clerc, M.: Spider monkey optimization algorithm for numerical optimization. Memet. Comput. 6(1), 31–47 (2014)
12. Cuevas, E., Cienfuegos, M., Zaldívar, D., Cisneros, M.P.: A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst. Appl. 40(16), 6374–6384 (2013)

13. Cuevas, E., Cienfuegos, M.: A new algorithm inspired in the behavior of the social-spider for constrained optimization. Expert Syst. Appl. **41**(2), 412–425 (2014)
14. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput. Aided Des. **43**(3), 303–315 (2011)
15. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf. Sci. **183**(1), 1–15 (2012)
16. Rao, R.V., Rai, D.P.: Optimization of fused deposition modeling process using teaching-learning-based optimization algorithm. Eng. Sci. Technol. Int. J. **19**, 587–603 (2016)
17. Yang, R., Omar, A.S.: Investigation of multiple rectangular aperture irises in rectangular waveguide using TE modes. IEEE Trans. Microw. Theory Tech. **41**(8), 1369–1374 (1993)