

Chapter 2

A Swarm Global Optimization Algorithm Inspired in the Behavior of the Social-Spider

2.1 Introduction

The collective intelligent behavior of insect or animal groups in nature such as flocks of birds, colonies of ants, schools of fish, swarms of bees, and termites have attracted the attention of researchers. The aggregate behavior of insects or animals is called swarm behavior. Entomologists have studied this collective phenomenon to model biological swarms, and engineers applied these models as a framework for solving complex real-world problems. This branch of artificial intelligence which deals with the collective behavior of swarms through complex interaction of individuals without supervision is referred to as swarm intelligence. Bonabeau defined swarm intelligence as “any attempt to design algorithms or distributed problem solving devices inspired by the collective behavior of the social insect colonies and other animal societies” [1]. Swarm intelligence has some advantages such as scalability, fault tolerance, adaptation, speed, modularity, autonomy, and parallelism [2].

The key components of swarm intelligence are self-organization and division of labors. In a self-organizing system, each of the covered units may respond to local stimuli individually and act together to accomplish a global task via division of labors without a centralized supervision. The entire system can adapt to internal and external changes efficiently.

Several swarm algorithms have been developed by a combination of deterministic rules and randomness, mimicking the behavior of insect or animal groups in nature. Such methods include the social behavior of bird flocking and fish schooling such as the Particle Swarm Optimization (PSO) algorithm [3], the cooperative behavior of bee colonies such as the Artificial Bee Colony (ABC) technique [4], the social foraging behavior of bacteria such as the Bacterial Foraging Optimization Algorithm (BFOA) [5], the simulation of the herding behavior of krill individuals such as the Krill Herd (KH) method [6], the mating behavior of firefly insects such as the Firefly (FF) method [7] and the emulation of

the lifestyle of cuckoo birds such as the Cuckoo Optimization Algorithm (COA) [8].

Insect colonies and animal groups provide a rich set of metaphors for designing swarm optimization algorithms. Such cooperative entities are complex system composed by individuals with different cooperative-tasks where each member tends to reproduce specialized behaviors depending generally on its gender [9]. However, most of the swarm algorithms model individuals as unisex performing virtually the same behavior. Under these circumstances, algorithms waste the possibility to add new and selective operators as a result of considering individuals with different characteristics such as sex, task-responsibility, etc. These operators could incorporate computational mechanisms to improve several important algorithm characteristics such as population diversity or searching capacities.

Although PSO and ABC are the most popular swarm algorithms for solving complex optimization problems, they present serious flaws such as premature convergence and difficulty to overcome local minima [10, 11]. The reason of these problems is the operators used for modifying the individual positions. In such algorithms, during their evolution, the position of each agent in the next iteration is updated yielding an attraction towards the position of the best particle seen so-far (in case of PSO) or of other randomly chosen individual (in case of ABC). Such behaviors produce that the entire population, as the algorithm evolves, concentrates around the best particle or diverges without control, favoring the premature convergence or damaging the exploration-exploitation balance [12, 13].

The interesting and exotic collective behaviors of social insects have fascinated and attracted the interest of researchers for many years. The collaborative swarming behavior that we observe in these groups provides survival advantages, where insect aggregations of relatively simple and “unintelligent” individuals can accomplish very complex tasks using only limited local information and simple rules of behavior [14]. Social-spiders are a representative example of social insects [15]. A social-spider is a spider species whose members maintain a set of complex cooperative behaviors [16]. Whereas most spiders are solitary and even aggressive toward other members of their own species, social-spiders show a tendency to live in groups, forming long-lasting aggregations, often referred to as colonies [17]. In a social-spider colony, each member, depending on its sex, executes a variety of tasks, such as predation, mating, web design, and social interaction [17, 18]. The web, as an important part of the colony, is not only used as a common environment for all members, but also as a communication channel among them [19]. Therefore, important information (such as trapped preys or mating possibilities) is transmitted through the web in form of small vibrations. Such information, considered as a local knowledge, is employed by each member to conduct its own cooperative behavior, influencing simultaneously the social regulation of the colony [20].

In this chapter, a novel swarm algorithm, namely the Social Spider Optimization (SSO) is presented for solving optimization tasks. The SSO algorithm is based on the simulation of the cooperative behavior of social-spiders. In the presented algorithm, individuals emulate a group of spiders which interact to each other based on the biological laws of the cooperative colony. The algorithm considers two

different search agents (spiders): males and females. Depending on the sex, each individual is conducted by a set of different evolutionary operators which mimic the different cooperative behaviors assumed in the colony. Different to most of the existent swarm algorithms, in the presented approach, each individual is modeled considering two different genders. Such fact allows not only to emulate in a better realistic way the cooperative behavior of the colony, but also to incorporate computational mechanisms to avoid critical flaws present in the popular PSO and ABC algorithms, such as premature convergence and incorrect exploration-exploitation balance. To illustrate the proficiency and robustness of the presented approach, it is compared to other well-known evolutionary methods. The comparison examines several standard benchmark functions which are commonly considered within the literature of evolutionary algorithms. The results show a high performance of the presented method when searching for a global optimum of several benchmark functions.

This chapter is organized as follows. In Sect. 2.2, are introduced the basic biological aspects of the algorithm. In Sect. 2.3, the novel SSO algorithm and its characteristics are both described. Section 2.4 presents the experimental results and the comparative study. Finally, in Sect. 2.5, conclusions are drawn.

2.2 Biologic Fundamentals

Social insect societies are complex cooperative systems that self-organize within a set of constraints. Cooperative groups are better at manipulating and exploiting their environment, defending resources and brood, and allow for task specialization among group members [21, 22]. A social insect colony functions as an integrated unit that not only possesses the ability to operate in a distributed manner, but also undertake enormous construction of global projects [23]. It is important to acknowledge that global order in social insects can arise as a result of internal interactions among insects.

A few species of spiders have been documented exhibiting a degree of social behavior [15]. One can generalize the behavior of these species in two basic forms, solitary spiders and social spiders [17]. This classification is made based on the level of cooperative behavior that they exhibit [18]. In general, solitary spiders create and maintain their own web while live in scarce contact with other individuals of the same species. In contrast, social spiders form colonies that remain together on a communal web, where a close spatial separation is presented between group members [19].

A social spider colony is composed of two fundamental components: members and a communal web. Members are divided in two different categories, males and females. An interesting characteristic of social-spiders is the highly female-biased populations. Some studies suggest that the number of male spiders barely reaches the 30 % of the total colony members [17, 24]. In the colony, each member, depending on its gender, cooperate in different activities such as build and maintain

the communal web, prey capture, mating and social contact [20]. Interactions among members are either direct or indirect [25]. Direct interactions imply body contact, or the exchange of fluid, such as mating. For indirect interactions, it is used the communal web as a “medium of communication”. Through the communal web, it is transmitted important information available for each colony member [19]. This information, encoded in form of small vibrations, is a critical aspect for the collective coordination among the members [20]. Since the vibrations depend on the weight and distance of the elements which provoke them, they are employed by the colony members to decode several messages, such as size of the trapped preys, characteristics of the neighboring members, etc.

In spite of the complexity, all the cooperative global patterns, presented in a colony level, are generated as a result of internal interactions among colony members [26]. Such internal iterations involve a set of simple behavioral rules followed by each spider in the colony. Behavioral rules are divided in two different classes: social interaction (cooperative behavior) and mating [27].

As a social insect, spiders perform cooperative interaction over other colony members. The way in which this behavior takes place depends on the spider gender. Female spiders which show a major tendency to socialize present an attraction or dislike over other spiders irrespective of the gender [17]. For a particular female spider, such attraction or dislike is commonly developed over other spiders that according to their vibrations (emitted over the communal web) represent strong colony members [20]. Since the vibrations depend on the weight and distance of the members which provoke them, strong vibrations are produced either by big spiders or neighboring members [19]. The bigger a spider is, the better it is considered as a colony member. The final decision of attraction or dislike over a determined member is taken according to an internal state which is influenced by several factors such as reproduction cycle, curiosity, and other random phenomena [20].

Different to female spiders, the behavior of male members is reproductive oriented [28]. Male spiders recognize themselves a subgroup of alpha males which dominate the colony resources. Therefore, the male population is divided in two classes: dominant and non-dominant male spiders [28]. Dominant male spiders have better fitness characteristics (normally size) in comparison with non-dominant. As a main behavior, dominant males are attracted to the closest female spider in the communal web. In contrast, non-dominant male spiders tend to concentrate in the center of the male population, as a strategy to take advantage of the resources wasted by the dominant males [29].

Mating is an important operation that not only assures the colony survival, but also allows the information exchange among members. Mating in a social-spider colony is performed by dominant males and the female members [30]. Under such circumstances, when a dominant male spider locates to one or more female members within a specific range, it mates with all the females in order to produce offspring [31].

2.3 The Social Spider Optimization (SSO) Algorithm

In this chapter, the operational principles from the social-spider colony have been used as guidelines for developing a new swarm optimization algorithm. The SSO assumes that entire search space is a communal web, where all the social-spiders interact. In the presented approach, each solution within the search space represents a spider position in the communal web. Every spider receives a weight according to the fitness value of the solution that the social-spider symbolizes. The algorithm models two different search agents (spiders): males and females. Depending on the gender, each individual is conducted by a set of different evolutionary operators which mimic the different cooperative behaviors assumed in the colony.

An interesting characteristic of social-spiders is the highly female-biased populations. In order to emulate this fact, the algorithm starts by defining the number of female and male spiders that will be characterized as individuals in the search space. The number of females N_f is randomly selected within the range of 65–90 % of the entire population N , previously chosen. Therefore, N_f is calculated by the following equation:

$$N_f = \text{floor}[(0.9 - \text{rand} \cdot 0.25) \cdot N] \quad (2.1)$$

where rand is a random number between $[0, 1]$ whereas $\text{floor}(\cdot)$ maps a real number to an integer number. The number of male spiders N_m is computed as the complement between N and N_f . It is calculated as follows:

$$N_m = N - N_f \quad (2.2)$$

Therefore, the complete population S , composed by N elements, is divided in two sub-groups F and M . The Group F assembles the set of female individuals ($F = \{f_1, f_2, \dots, f_{N_f}\}$) whereas M groups the male members ($M = \{m_1, m_2, \dots, m_{N_m}\}$), where $S = F \cup M$ ($S = \{s_1, s_2, \dots, s_N\}$), such that $S = \{s_1 = f_1, s_2 = f_2, \dots, s_{N_f} = f_{N_f}, s_{N_f+1} = m_1, s_{N_f+2} = m_2, \dots, s_N = m_{N_m}\}$.

2.3.1 Fitness Assignment

In the biological metaphor, the spider size is the characteristic that evaluates the individual capacity to perform better its assigned tasks. In the presented approach, every individual (spider) receive a weight w_i which represents the solution quality that corresponds to the spider i (irrespective of the gender) of the population S . In order to calculate the mass of every spider the next equations are used:

$$w_i = \frac{J(s_i) - \text{worst}_S}{\text{best}_S - \text{worst}_S} \quad (2.3)$$

where $J(\mathbf{s}_i)$ is the fitness value obtained by the evaluation of the spider position \mathbf{s}_i with regard to the objective function $J(\cdot)$. The values $worst_{\mathbf{S}}$ and $best_{\mathbf{S}}$ are defined as follows (considering a maximization problem):

$$best_{\mathbf{S}} = \max_{k \in \{1, 2, \dots, N\}} (J(\mathbf{s}_k)) \text{ and } worst_{\mathbf{S}} = \min_{k \in \{1, 2, \dots, N\}} (J(\mathbf{s}_k)) \quad (2.4)$$

2.3.2 Modeling of the Vibrations Through the Communal Web

The communal web is used as a mechanism to transmit information among the colony members. This information, encoded in form of small vibrations, is a critical aspect for the collective coordination for all individuals in the population. The vibrations depend on the weight and distance of the spider which provoke them. Since the distance is relative to the individual that provokes the vibrations and the member who detects them, members near to the individual that provokes the vibrations perceive stronger vibrations in comparison with members located in distant positions. In order to reproduce this process, the vibrations perceived by the individual i as a result of the information transmitted by the member j are modeled according to the following equation:

$$Vib_{i,j} = w_j \cdot e^{-d_{i,j}^2} \quad (2.5)$$

where the $d_{i,j}$ is the Euclidian distance between the spiders i and j , such that $d_{i,j} = \|\mathbf{s}_i - \mathbf{s}_j\|$.

Although it is virtually possible to compute the perceived-vibrations considering any pair of individuals, three special relations are considered in the SSO approach:

1. The vibrations $Vibc_i$ perceived by the individual i (\mathbf{s}_i) as a result of the information transmitted by the member c (\mathbf{s}_c). Where c is an individual that has two important characteristics, it is the nearest member to i and posses a higher weight in comparison to i ($w_c > w_i$).

$$Vibc_i = w_c \cdot e^{-d_{i,c}^2} \quad (2.6)$$

2. The vibrations $Vibb_i$ perceived by the individual i as a result of the information transmitted by the member b (\mathbf{s}_b). Where b is the individual with the best weight (best fitness value) of the entire population \mathbf{S} , such that $w_b = \max_{k \in \{1, 2, \dots, N\}} (w_k)$.

$$Vibb_i = w_b \cdot e^{-d_{i,b}^2} \quad (2.7)$$

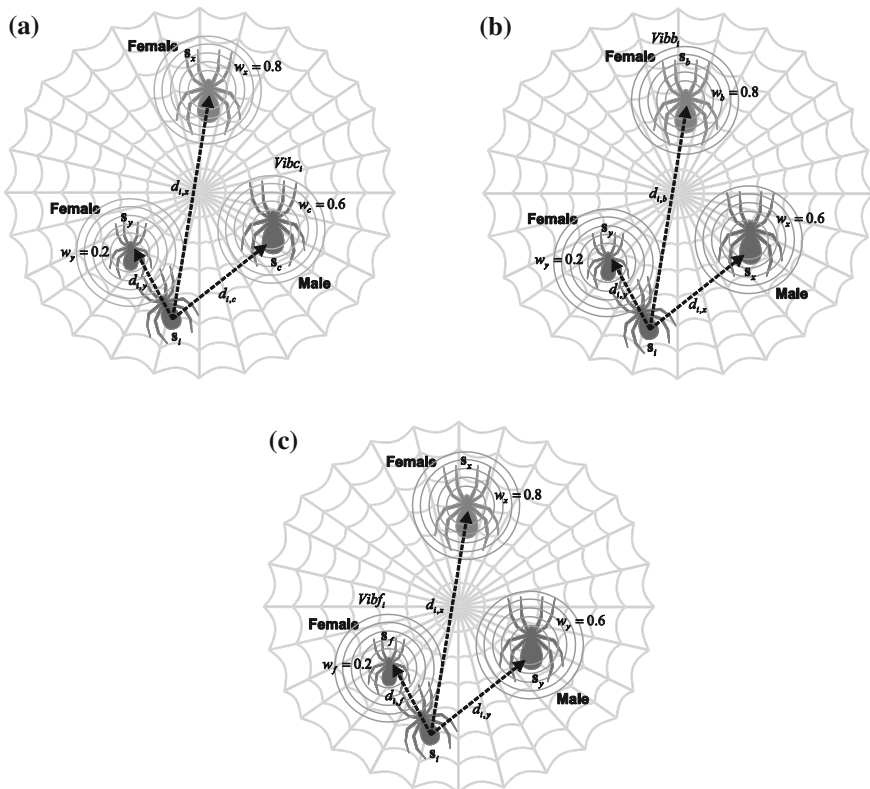


Fig. 2.1 Configuration of each special relation: **a** $Vibc_i$, **b** $Vibb_i$ and **c** $Vibf_i$

3. The vibrations $Vibf_i$ perceived by the individual i (s_i) as a result of the information transmitted by the member f (s_f). Where f is the nearest female individual to i .

$$Vibf_i = w_f \cdot e^{-d_{i,f}^2} \quad (2.8)$$

Figure 2.1 shows the configuration of each special relation: (a) $Vibc_i$, (b) $Vibb_i$ and (c) $Vibf_i$.

2.3.3 Initializing the Population

Like other evolutionary algorithms, the SSO is an iterative process; where the first step is to randomly initialize the entire population (females and males). The algorithm begins by initializing the set S of N spider positions. Each spider position, \mathbf{f}_i or \mathbf{m}_i , is a n -dimensional vector containing the parameter values to be

optimized. Such values are randomly and uniformly distributed between the pre-specified lower initial parameter bound p_j^{low} and the upper initial parameter bound p_j^{high} , just as it described by the following expressions:

$$\begin{aligned} f_{i,j}^0 &= p_j^{low} + \text{rand}(0, 1) \cdot (p_j^{high} - p_j^{low}) & m_{k,j}^0 &= p_j^{low} + \text{rand}(0, 1) \cdot (p_j^{high} - p_j^{low}) \\ i &= 1, 2, \dots, N_f; j = 1, 2, \dots, n & k &= 1, 2, \dots, N_m; j = 1, 2, \dots, n \end{aligned} \quad (2.8)$$

where j , i and k are the parameter and individual indexes respectively whereas zero indicates the initial population. Hence, $f_{i,j}$ is the j th parameter of the i th female spider position.

2.3.4 Cooperative Operators

2.3.4.1 Female Cooperative Operator

Social-spiders perform cooperative interaction over other colony members. The way in which this behavior takes place depends on the spider gender. Female spiders present an attraction or dislike over other spiders irrespective of the gender. For a particular female spider, such attraction or dislike is commonly developed over other spiders that according to their vibrations (emitted over the communal web) represent strong colony members. Since the vibrations depend on the weight and distance of the members which provoke them, strong vibrations are produced either by big spiders or neighboring members relative to the individual which perceives them. The final decision of attraction or dislike over a determined member is taken according to an internal state which is influenced by several factors such as reproduction cycle, curiosity, and other random phenomena.

In order to emulate the cooperative behavior of the female spider, a new operator is defined. The operator considers the position change of the female spider i at each iteration. Such position change (which can be of attraction or repulsion) is computed as a combination of three different elements. The first one involves the change in regard to the nearest member to i with a higher weight (this member produces the vibration $Vibc_i$). The second one considers the change regarding the best individual of the entire population \mathbf{S} (such individual produces the vibration $Vibb_i$). Finally, the third one implements the incorporation of a random movement. Since the final movement of attraction or repulsion depends on several random phenomena, this election is modeled as a stochastic decision. For this operation, a uniform random number r_m is generated within the range $[0, 1]$. If r_m is smaller than a threshold PF , an attraction movement is generated; otherwise, a repulsion movement is produced. Therefore, such operator can be modeled as follows:

$$\mathbf{f}_i^{k+1} = \begin{cases} \mathbf{f}_i^k + \alpha \cdot Vibc_i \cdot (\mathbf{s}_c - \mathbf{f}_i^k) + \beta \cdot Vibb_i \cdot (\mathbf{s}_b - \mathbf{f}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2}) & \text{with probability } PF \\ \mathbf{f}_i^k - \alpha \cdot Vibc_i \cdot (\mathbf{s}_c - \mathbf{f}_i^k) - \beta \cdot Vibb_i \cdot (\mathbf{s}_b - \mathbf{f}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2}) & \text{with probability } 1 - PF \end{cases} \quad (2.9)$$

where α , β , δ and rand are random numbers between $[0, 1]$ whereas k represents the iteration number. The individual \mathbf{s}_c and \mathbf{s}_b represent the nearest member to i with a higher weight and the best individual of the entire population \mathbf{S} , respectively.

Under this operation, each particle presents a movement which combines the past position, with the attraction or repulsion vector over the local best element \mathbf{s}_c and the global best individual seen \mathbf{s}_b so-far. This particular type of interaction avoids the quick concentration of particles in only one point and encourages each particle to search around a local candidate region in its neighborhood (\mathbf{s}_c), rather than interacting with a particle (\mathbf{s}_b) in a distant region of the domain. The use of this scheme has two advantages. First, it prevents the particles from moving toward the global best position making the algorithm less susceptible to premature convergence. Second, it encourages the particles to explore their own neighborhood thoroughly before converging toward global best position. Therefore, it provides the algorithm with global search ability and enhances the exploitative behavior of the presented approach.

2.3.4.2 Male Cooperative Operator

According to the biological behavior of the social-spider, male population is divided in two classes: dominant and non-dominant male spiders. Dominant male spiders have better fitness characteristics (normally size) in comparison with non-dominant. Dominant males are attracted to the closest female spider in the communal web. In contrast, non-dominant male spiders tend to concentrate in the center of the male population, as a strategy to take advantage of the resources wasted by the dominant males.

For emulating such cooperative behavior, the male members are divided in two different groups (dominant members D and non-dominant members ND) according to their position with regard to the median member. Male members, with a weight value above the median value within the male population, are considered the dominant individuals D. On the other hand, the individuals under the median value are labeled as non-dominant ND males. In order to implement such computation, the male population \mathbf{M} ($\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{N_m}\}$) is arranged according to their weight value, in a decreasing order. Thus, the individual whose weight $w_{N_f + m}$ is located in the middle is considered the median male member. Since the indexes of the male population \mathbf{M} in regard to the entire population \mathbf{S} are incremented by the number of female members N_f , the median weight is indexed by $N_f + m$. According to this, change of positions for the male spider can be modeled as follows:

$$\mathbf{m}_i^{k+1} = \begin{cases} \mathbf{m}_i^k + \alpha \cdot Vibf_i \cdot (\mathbf{s}_f - \mathbf{m}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2}) & \text{if } w_{N_f+i} > w_{N_f+m} \\ \mathbf{m}_i^k + \alpha \cdot \left(\frac{\sum_{h=1}^{N_m} \mathbf{m}_h^k \cdot w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} - \mathbf{m}_i^k \right) & \text{if } w_{N_f+i} \leq w_{N_f+m} \end{cases} \quad (2.10)$$

where the individual \mathbf{s}_f represents the nearest female individual to the male member i whereas $(\sum_{h=1}^{N_m} \mathbf{m}_h^k \cdot w_{N_f+h} / \sum_{h=1}^{N_m} w_{N_f+h})$ correspond to the weighted mean of the male population \mathbf{M} .

By using this operator, two different behaviors are produced. First, the set \mathbf{D} of particles is attracted to others in order to provoke mating. Such behavior allows to incorporate diversity in the population. Second, the set \mathbf{ND} of particles are attracted to weighted mean of the male population \mathbf{M} . This fact is used to partially control the search process according to the average performance of a sub-group of the population. Such mechanism acts as a filter which avoids that very good individuals or extremely bad individuals influence the search process.

2.3.5 Mating Operator

Mating in a social-spider colony is performed by dominant males and the female members. Under such circumstances, when a dominant male \mathbf{m}_g spider ($g \in \mathbf{D}$) locates a set \mathbf{E}^g of female members within a specific range r (range of mating), it mates, forming a new brood \mathbf{s}_{new} which is generated considering all the elements of the set \mathbf{T}^g that, in turn, has been generated by the union $\mathbf{E}^g \cup \mathbf{m}_g$. It is important to emphasize that if the set \mathbf{E}^g is empty, the mating operation is canceled. The range r is defined as a radius which depends on the size of the search space. Such radius r is computed according to the following model:

$$r = \frac{\sum_{j=1}^n (p_j^{high} - p_j^{low})}{2 \cdot n} \quad (2.11)$$

In the mating process, the weight of each involved spider (elements of \mathbf{T}^g) defines the probability of influence of each individual into the new brood. The spiders with heavier weight are more probable to influence the new product, while elements with lighter weight have a lower probability. The influence probability Ps_i of each member is assigned using the roulette method, which is defined as follows:

$$Ps_i = \frac{w_i}{\sum_{j \in \mathbf{T}^k} w_j}, \quad (2.12)$$

where $i \in \mathbf{T}^g$.

Once the new spider is formed, it is compared, the new spider candidate s_{new} with the worst spider s_{wo} of the colony, according to their weight values (where $w_{wo} = \min_{l \in \{1,2,\dots,N\}} (w_l)$). If the new spider is better than the worst spider, the worst spider is replaced by the new one. Otherwise, the new spider is discarded and the population does not suffer changes. In case of replacement, the new spider assumes the sex and the same index of the replaced spider. Such fact assures that the entire population S maintains the original rate between female and male members.

In order to illustrate the mating operation, it is considered an example, where Fig. 2.2a is used as optimization problem. It is also assumed a population S of seven different 2-dimensional members ($N = 8$), five females ($N_f = 5$) and two males ($N_m = 3$). Figure 2.2b shows the initial configuration of the presented example. In the example, three different female members $f_2(s_2)$, $f_3(s_3)$ and $f_4(s_4)$ constitute the set E^2 located inside of the influence range r of a dominant male $m_2(s_7)$. Then, the new candidate spider s_{new} is generated from the elements f_2 , f_3 , f_4 and m_2 which constitute the set T^2 . Therefore, the value of the first decision variable $s_{new,1}$ for the new spider is chosen by means of the roulette mechanism considering the values already existing from the set $\{f_{2,1}, f_{3,1}, f_{4,1}, m_{2,1}\}$. The value of the second decision variable $s_{new,2}$ is also chosen in the same manner. Table 2.1 shows the data for

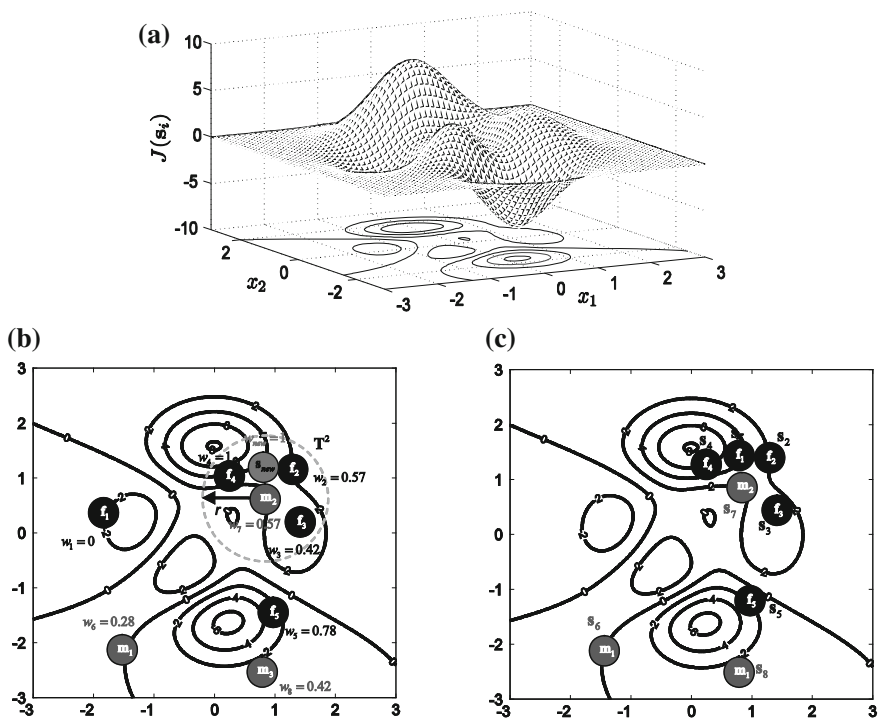


Fig. 2.2 Example of the mating operation: **a** optimization problem, **b** initial configuration before mating and **c** configuration after the mating operation

Table 2.1 Data for constructing the new spider s_{new} through the roulette method

Spider		Position	w_i	P_{S_i}	<div><table><tr><td>f_4</td><td>39%</td></tr><tr><td>f_2</td><td>22%</td></tr><tr><td>m_2</td><td>22%</td></tr><tr><td>f_3</td><td>16%</td></tr></table></div> Roulette	f_4	39%	f_2	22%	m_2	22%	f_3	16%
f_4	39%												
f_2	22%												
m_2	22%												
f_3	16%												
s_1	f_1	(-1.9, 0.3)	0.00	—									
s_2	f_2	(1.4, 1.1)	0.57	0.22									
s_3	f_3	(1.5, 0.2)	0.42	0.16									
s_4	f_4	(0.4, 1.0)	1.00	0.39									
s_5	f_5	(1.0, -1.5)	0.78	—									
s_6	m_1	(-1.3, -1.9)	0.28	—									
s_7	m_2	(0.9, 0.7)	0.57	0.22									
s_8	m_3	(0.8, -2.6)	0.42	—									
s_{new}		(0.9, 1.1)	1.00	—									

constructing the new spider through the roulette method. Once the new spider s_{new} is formed, it is calculated its weight w_{new} . As s_{new} is better than the worst member f_1 present in the population S , f_1 is replaced by s_{new} . With the replacement, s_{new} assumes the same sex and index than f_1 . Figure 2.2c shows the configuration of S after the mating process.

Under this operation, new generated particles exploit locally the search space inside of the range of mating in order to find better individuals.

2.3.6 Computational Procedure

The computational procedure for the presented algorithm can be summarized as follows:

Step 1	<p>Considering N as the total number of n-dimensional colony members, define the number of male N_m and females N_f in the entire population S</p> $N_f = \text{floor}[(0.9 - \text{rand} \cdot 0.25) \cdot N] \text{ and } N_m = N - N_f,$ <p>where rand is a random number between $[0, 1]$ whereas $\text{floor}(\cdot)$ maps a real number to an integer number</p>
Step 2	<p>Initialize randomly the female ($F = \{f_1, f_2, \dots, f_{N_f}\}$) and male ($M = \{m_1, m_2, \dots, m_{N_m}\}$) members where $S = \{s_1 = f_1, s_2 = f_2, \dots, s_{N_f} = f_{N_f}, s_{N_f+1} = m_1, s_{N_f+2} = m_2, \dots, s_N = m_{N_m}\}$ and calculate the range of mating</p> $r = \frac{\sum_{j=1}^n (p_j^{\text{high}} - p_j^{\text{low}})}{2 \cdot n}$ <p>for ($i = 1; i < N_f + 1; i++$) for ($j = 1; j < n + 1; j++$) $f_{i,j}^0 = p_j^{\text{low}} + \text{rand}(0, 1) \cdot (p_j^{\text{high}} - p_j^{\text{low}})$ end for end for for ($k = 1; k < N_m + 1; k++$) for ($j = 1; j < n + 1; j++$) $m_{k,j}^0 = p_j^{\text{low}} + \text{rand} \cdot (p_j^{\text{high}} - p_j^{\text{low}})$ end for end for</p>
Step 3	<p>Calculate the weight of every spider of S (Sect. 2.3.1)</p> <p>for ($i = 1, i < N + 1; i++$) $w_i = \frac{J(s_i) - \text{worsts}}{\text{bests} - \text{worsts}}$ where $\text{bests} = \max_{k \in \{1, 2, \dots, N\}} (J(s_k))$ and $\text{worsts} = \min_{k \in \{1, 2, \dots, N\}} (J(s_k))$ end for</p>

(continued)

(continued)

Step 4	Move female spiders according to the female cooperative operator (Sect. 2.3.4) for $i = 1; i < N_f + 1; i++$ Calculate $Vibc_i$ and $Vibb_i$ (Sect. 2.3.2) If $(r_m < PF)$; where $r_m \in \text{rand}(0, 1)$ $\mathbf{f}_i^{k+1} = \mathbf{f}_i^k + \alpha \cdot Vibc_i \cdot (\mathbf{s}_c - \mathbf{f}_i^k) + \beta \cdot Vibb_i \cdot (\mathbf{s}_b - \mathbf{f}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2})$ else if $\mathbf{f}_i^{k+1} = \mathbf{f}_i^k - \alpha \cdot Vibc_i \cdot (\mathbf{s}_c - \mathbf{f}_i^k) - \beta \cdot Vibb_i \cdot (\mathbf{s}_b - \mathbf{f}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2})$ end if end for
Step 5	Move the male spiders according to the male cooperative operator (Sect. 3.1.4) Find the median male individual (w_{N_f+m}) from M for $(i = 1; i < N_m + 1; i++)$ Calculate $Vibf_i$ (Sect. 2.3.2) If $(w_{N_f+i} > w_{N_f+m})$ $\mathbf{m}_i^{k+1} = \mathbf{m}_i^k + \alpha \cdot Vibf_i \cdot (\mathbf{s}_f - \mathbf{m}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2})$ else if $\mathbf{m}_i^{k+1} = \mathbf{m}_i^k + \alpha \cdot \left(\frac{\sum_{h=1}^{N_m} \mathbf{m}_h^k \cdot w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} - \mathbf{m}_i^k \right)$ end if end for
Step 6	Perform mating operation (Sect. 2.3.5) for $i = 1; i < N_m + 1; i++$ If $(\mathbf{m}_i \in \mathbf{D})$ Find \mathbf{E}^i If $(\mathbf{E}^i$ is not empty) Form \mathbf{s}_{new} using the roulette method If $(w_{new} > w_{wo})$ $\mathbf{s}_{wo} = \mathbf{s}_{new}$ end if end if end if end for
Step 7	If the stop criteria is met, the process is finished; otherwise, go back to Step 3

2.3.7 Discussion About the SSO Algorithm

Evolutionary algorithms (EA) have been widely employed for solving complex optimization problems. These methods are found to be more powerful than conventional methods based on formal logics or mathematical programming [32]. In an EA algorithm, search agents have to decide whether to explore unknown search positions or to exploit already tested positions in order to improve their solution quality. Pure exploration degrades the precision of the evolutionary process but increases its capacity to find new potentially solutions. On the other hand, pure

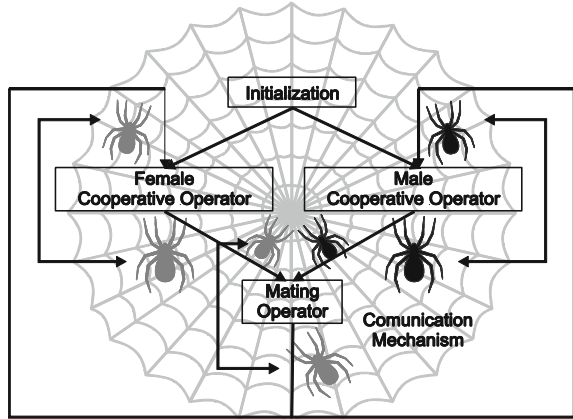
exploitation allows refining existent solutions but adversely drives the process to local optimal solutions. Therefore, the ability of an EA to find a global optimal solution depends on its capacity to find a good balance between the exploitation of found-so-far elements and the exploration of the search space [33]. So far, the exploration–exploitation dilemma has been an unsolved issue within the framework of evolutionary algorithms.

EA define individuals with the same property, performing virtually the same behavior. Under these circumstances, algorithms waste the possibility to add new and selective operators as a result of considering individuals with different characteristics. These operators could incorporate computational mechanisms to improve several important algorithm characteristics such as population diversity or searching capacities.

On the other hand, PSO and ABC are the most popular swarm algorithms for solving complex optimization problems. However, they present serious flaws such as premature convergence and difficulty to overcome local minima [10, 11]. The reason of these problems is the operators used for modifying the individual positions. In such algorithms, during their evolution, the position of each agent in the next iteration is updated yielding an attraction towards the position of the best particle seen so-far (in case of PSO) or of other randomly chosen individual (in case of ABC). Such behaviors produce that the entire population, as the algorithm evolves, concentrates around the best particle or diverges without control, favoring the premature convergence or damaging the exploration-exploitation balance [12, 13].

Different to other EA, in SSO, each individual is modeled considered two different genders. Such fact allows incorporating computational mechanisms to avoid critical flaws present in the popular PSO and ABC algorithms, such as premature convergence and incorrect exploration-exploitation balance. Since the optimization point of view, the use of the social-spider behavior as metaphor introduces interesting concepts in the evolutionary algorithms: The fact of dividing the entire population in different search-agent categories and the employment of specialized operators applied selectively to each of them. Using this framework, it is possible to improve the balance between exploitation and exploration, conserving in the same population, individuals who achieve efficient exploration (female spiders) and individuals that verify extensive exploitation (male spiders). Furthermore, the social-spider behavior mechanism introduces an interesting computational scheme. Such scheme presents three important particularities. First, individuals are separately processed according to their characteristics. Second, the operators share the same communication mechanism. This mechanism allows employing important information of the evolutionary process to modify the influence of each operator. Third, although the operators modify the position of only an individual type, they use global information (positions of all individual types) in order to perform the modification. Figure 2.3 presents a schematic representation of the algorithm-data-flow. According to Fig. 2.3, the female cooperative and male cooperative operators process only female or male individuals, respectively. However, the mating operator modifies both individual types.

Fig. 2.3 Schematic representation of the algorithm-data-flow



2.4 Experimental Results

A comprehensive set of 19 functions, collected from Refs. [34–40], has been used to test the performance of the presented approach. Table 2.2 present the benchmark functions used in our experimental study. In the table, n indicates the dimension of the function, $f(\mathbf{x}^*)$ the optimum value of the function, \mathbf{x}^* the optimum position and S the search space (subset of R^n). A detailed description of each function is given in Table 2.2.

2.4.1 Performance Comparison to Other Swarm Algorithms

The SSO algorithm was applied to 19 functions whose results have been compared to those produced by the Particle Swarm Optimization (PSO) method [3] and the Artificial Bee Colony (ABC) algorithm [4]. These are considered as the most popular swarm algorithms in many optimization applications. In all comparisons, the population has been set to 50 individuals. The maximum iteration number for all functions has been set to 1000. Such stop criterion has been selected to maintain compatibility to similar works reported in the literature [41, 42].

The parameter setting for each algorithm in the comparison is described as follows:

1. PSO [3]: The parameters are set to $c_1 = 2$ and $c_2 = 2$; besides, the weight factor decreases linearly from 0.9 to 0.2.
2. ABC: The algorithm has been implemented using the guidelines provided by its own reference [4], using the parameter *limit* = 100.
3. SSO: Once determined experimentally, the parameter *PF* was set to 0.7. It is kept for all experiments presented in this section.

Table 2.2 Test functions used in the experimental study

Name	Function	S	Dim	Minimum
Sphere	$f_1(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Schwefel 2.22	$f_2(\mathbf{x}) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Schwefel 1.2	$f_3(\mathbf{x}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
F4	$f_4(\mathbf{x}) = 418.9829n + \sum_{i=1}^n (-x_i \sin(\sqrt{ x_i }))$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Rosenbrock	$f_5(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$[-30, 30]^n$	$n = 30$	$\mathbf{x}^* = (1, \dots, 1);$ $f(\mathbf{x}^*) = 0$
Step	$f_6(\mathbf{x}) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Quartic	$f_7(\mathbf{x}) = \sum_{i=1}^n ix_i^4 + \text{random}(0, 1)$	$[-1.28, 1.28]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Dixon and price	$f_8(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=1}^n i(2x_i^2 - x_{i-1})^2$	$[-10, 10]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$

(continued)

Table 2.2 (continued)

Name	Function	S	Dim	Minimum
Levy	$f_9(\mathbf{x}) = 0.1 \left\{ \begin{array}{l} \sin^2(3\pi x_1) \\ + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] \\ + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \\ + \sum_{i=1}^n u(x_i, 5, 100, 4); \end{array} \right.$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-10, 10]^n$	$n = 30$	$\mathbf{x}^* = (1, \dots, 1);$ $f(\mathbf{x}^*) = 0$
Sum of Squares	$f_{10}(\mathbf{x}) = \sum_{i=1}^n ix_i^2$	$[-10, 10]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Zakharov	$f_{11}(\mathbf{x}) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$	$[-5, 10]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Penalized	$f_{12}(\mathbf{x}) = \frac{\pi}{n} \left\{ \begin{array}{l} 10 \sin(\pi y_1) + \\ \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \end{array} \right\}$ $y_i = 1 + \frac{(x_i + 1)}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$

(continued)

Table 2.2 (continued)

Name	Function	S	Dim	Minimum
Penalized 2	$f_{13}(\mathbf{x}) = 0.1 \left\{ \begin{array}{l} \sin^2(3\pi x_1) \\ + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] \\ + \sum_{i=1}^n u(x_i, 5, 100, 4) \end{array} \right\}$ <p>where $u(x_i, a, k, m)$ is the same as Penalized function</p>	$[-50, 50]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Schwefel	$f_{14}(\mathbf{x}) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500, 500]^n$	$n = 30$	$\mathbf{x}^* = (420, \dots, 420);$ $f(\mathbf{x}^*) = -418.9829 \times n$
Rastrigin	$f_{15}(\mathbf{x}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Ackley	$f_{16}(\mathbf{x}) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + \exp$	$[-32, 32]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Griewank	$f_{17}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Powelll	$f_{18}(\mathbf{x}) = \sum_{i=1}^{n/k} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	$[-4, 5]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$
Salomon	$f_{19}(\mathbf{x}) = -\cos \left(2\pi \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2} + 1$	$[-100, 100]^n$	$n = 30$	$\mathbf{x}^* = (0, \dots, 0);$ $f(\mathbf{x}^*) = 0$

The experiment compares the SSO to other algorithms such as PSO and ABC. The results for 30 runs are reported in Table 2.3 considering the following performance indexes: the Average Best-so-far (AB) solution, the Median Best-so-far (MB) and the Standard Deviation (SD) of best-so-far solution. The best outcome for each function is boldfaced. According to this table, SSO delivers better results than PSO and ABC for all functions. In particular, the test remarks the largest difference in performance which is directly related to a better trade-off between exploration and exploitation.

A non-parametric statistical significance proof known as the Wilcoxon's rank sum test for independent samples [43, 44] has been conducted over the "average best-so-far" (AB) data of Table 2.2, with a 5 % significance level. Table 2.4 reports the p -values produced by Wilcoxon's test for the pair-wise comparison of the "average best so-far" of two groups. Such groups are formed by SSO vs. PSO and SSO vs. ABC. As a null hypothesis, it is assumed that there is no significant difference between mean values of the two algorithms. The alternative hypothesis

Table 2.3 Minimization result of benchmark functions of Table 2.2 with $n = 30$

		SSO	ABC	PSO
$f_1(x)$	AB	1.96E-03	2.90E-03	1.00E+03
	MB	2.81E-03	1.50E-03	2.08E-09
	SD	9.96E-04	1.44E-03	3.05E+03
$f_2(x)$	AB	1.37E-02	1.35E-01	5.17E+01
	MB	1.34E-02	1.05E-01	5.00E+01
	SD	3.11E-03	8.01E-02	2.02E+01
$f_3(x)$	AB	4.27E-02	1.13E+00	8.63E+04
	MB	3.49E-02	6.11E-01	8.00E+04
	SD	3.11E-02	1.57E+00	5.56E+04
$f_4(x)$	AB	5.40E-02	5.82E+01	1.47E+01
	MB	5.43E-02	5.92E+01	1.51E+01
	SD	1.01E-02	7.02E+00	3.13E+00
$f_5(x)$	AB	1.14E+02	1.38E+02	3.34E+04
	MB	5.86E+01	1.32E+02	4.03E+02
	SD	3.90E+01	1.55E+02	4.38E+04
$f_6(x)$	AB	2.68E-03	4.06E-03	1.00E+03
	MB	2.68E-03	3.74E-03	1.66E-09
	SD	6.05E-04	2.98E-03	3.06E+03
$f_7(x)$	AB	1.20E+01	1.21E+01	1.50E+01
	MB	1.20E+01	1.23E+01	1.37E+01
	SD	5.76E-01	9.00E-01	4.75E+00
$f_8(x)$	AB	2.14E+00	3.60E+00	3.12E+04
	MB	3.64E+00	8.04E-01	2.08E+02
	SD	1.26E+00	3.54E+00	5.74E+04

(continued)

Table 2.3 (continued)

		SSO	ABC	PSO
$f_9(x)$	AB	6.92E-05	1.44E-04	2.47E+00
	MB	6.80E-05	8.09E-05	9.09E-01
	SD	4.02E-05	1.69E-04	3.27E+00
$f_{10}(x)$	AB	4.44E-04	1.10E-01	6.93E+02
	MB	4.05E-04	4.97E-02	5.50E+02
	SD	2.90E-04	1.98E-01	6.48E+02
$f_{11}(x)$	AB	6.81E+01	3.12E+02	4.11E+02
	MB	6.12E+01	3.13E+02	4.31E+02
	SD	3.00E+01	4.31E+01	1.56E+02
$f_{12}(x)$	AB	5.39E-05	1.18E-04	4.27E+07
	MB	5.40E-05	1.05E-04	1.04E-01
	SD	1.84E-05	8.88E-05	9.70E+07
$f_{13}(x)$	AB	1.76E-03	1.87E-03	5.74E-01
	MB	1.12E-03	1.69E-03	1.08E-05
	SD	6.75E-04	1.47E-03	2.36E+00
$f_{14}(x)$	AB	-9.36E+02	-9.69E+02	-9.63E+02
	MB	-9.36E+02	-9.60E+02	-9.92E+02
	SD	1.61E+01	6.55E+01	6.66E+01
$f_{15}(x)$	AB	8.59E+00	2.64E+01	1.35E+02
	MB	8.78E+00	2.24E+01	1.36E+02
	SD	1.11E+00	1.06E+01	3.73E+01
$f_{16}(x)$	AB	1.36E-02	6.53E-01	1.14E+01
	MB	1.39E-02	6.39E-01	1.43E+01
	SD	2.36E-03	3.09E-01	8.86E+00
$f_{17}(x)$	AB	3.29E-03	5.22E-02	1.20E+01
	MB	3.21E-03	4.60E-02	1.35E-02
	SD	5.49E-04	3.42E-02	3.12E+01
$f_{18}(x)$	AB	1.87E+00	2.13E+00	1.26E+03
	MB	1.61E+00	2.14E+00	5.67E+02
	SD	1.20E+00	1.22E+00	1.12E+03
$f_{19}(x)$	AB	2.74E-01	4.14E+00	1.53E+00
	MB	3.00E-01	4.10E+00	5.50E-01
	SD	5.17E-02	4.69E-01	2.94E+00

Maximum number of iterations = 1000

considers a significant difference between the “average best-so-far” values of both approaches. All p -values reported in Table 2.4 are less than 0.05 (5 % significance level) which is a strong evidence against the null hypothesis. Therefore, such evidence indicates that SMS results are statistically significant and that it has not occurred by coincidence (i.e. due to common noise contained in the process).

Table 2.4 p -values produced by Wilcoxon's test comparing SSO versus ABC and SSO versus PSO over the "average best-so-far" (AB) values from Table 2.3

Function	SSO versus ABC	SSO versus PSO
$f_1(x)$	0.041	1.8E-05
$f_2(x)$	0.048	0.059
$f_3(x)$	5.4E-04	6.2E-07
$f_4(x)$	1.4E-07	4.7E-05
$f_5(x)$	0.045	7.1E-07
$f_6(x)$	2.3E-04	5.5E-08
$f_7(x)$	0.048	0.011
$f_8(x)$	0.017	0.043
$f_9(x)$	8.1E-04	2.5E-08
$f_{10}(x)$	4.6E-06	1.7E-09
$f_{11}(x)$	9.2E-05	7.8E-06
$f_{12}(x)$	0.022	1.1E-10
$f_{13}(x)$	0.048	2.6E-05
$f_{14}(x)$	0.044	0.049
$f_{15}(x)$	4.5E-05	7.9E-08
$f_{16}(x)$	2.8E-05	4.1E-06
$f_{17}(x)$	7.1E-04	6.2E-10
$f_{18}(x)$	0.013	8.3E-10
$f_{19}(x)$	4.9E-05	5.1E-08

2.5 Summary

In this chapter, a novel swarm algorithm, namely the Social Spider Optimization (SSO) has been presented for solving optimization tasks. The SSO algorithm is based on the simulation of the cooperative behavior of social-spiders. In the presented algorithm, individuals emulate a group of spiders which interact to each other based on the biological laws of the cooperative colony. The algorithm considers two different search agents (spiders): males and females. Depending on the sex, each individual is conducted by a set of different evolutionary operators which mimic the different cooperative behaviors assumed in the colony.

Different to most of the existent swarm algorithms, in the presented approach, each individual is modeled considered two different genders. Such fact allows not only to emulate in a better realistic way the cooperative behavior of the colony, but also to incorporate computational mechanisms to avoid critical flaws present in the popular PSO and ABC algorithms, such as premature convergence and incorrect exploration-exploitation balance.

SSO has been experimentally tested considering a suite of 19 benchmark functions. The performance of SSO has been also compared to the following swarm algorithms: the Particle Swarm Optimization method (PSO) [16], and the Artificial Bee Colony (ABC) algorithm [38]. Results have confirmed a high performance of the presented method in terms of the solution quality for solving most of benchmark functions.

The SSO's remarkable performance is associated with two different reasons: (i) the defined operators allow a better particle distribution in the search space, increasing the algorithm's ability to find the global optima; and (ii) the division of the population in different individual types, provides the use of different rates between exploration and exploitation during the evolution process.

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm Intelligence: from Natural to Artificial Systems*. Oxford University Press Inc, New York, NY, USA (1999)
2. Kassabalidis, I., El-Sharkawi, M.A., Marks R.J., Arabshahi, P., Gray, A.A.: Swarm intelligence for routing in communication networks, *Global Telecommunications Conference, GLOBECOM'01*, vol. 6, pp. 3613–3617. IEEE (2001)
3. Kennedy, J., Eberhart, R.: Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948. (1995)
4. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization. TechnicalReport-TR06. Erciyes University, Turkey (2005)
5. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **22**(3), 52–67 (2002)
6. Hossein, A., Hossein-Alavi, A.: Krill herd: a new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **17**, 4831–4845 (2012)
7. Yang, X.S.: *Engineering optimization: An introduction with metaheuristic applications*. Wiley (2010)
8. Rajabioun, R.: Cuckoo optimization algorithm. *Appl. Soft Comput.* **11**, 5508–5518 (2011)
9. Bonabeau, E.: Social insect colonies as complex adaptive systems. *Ecosystems* **1**, 437–443 (1998)
10. Wang, Y., Li, B., Weise, T., Wang, J., Yuan, B., Tian, Q.: Self-adaptive learning based particle swarm optimization. *Inf. Sci.* **181**(20), 4515–4538 (2011)
11. Wan-li, X., Mei-qing, A.: An efficient and robust artificial bee colony algorithm for numerical optimization. *Comput. Oper. Res.* **40**, 1256–1265 (2013)
12. Wang, H., Sun, H., Li, C., Rahnamayan, S., Jeng-shyang, P.: Diversity enhanced particle swarm optimization with neighborhood. *Inf. Sci.* **223**, 119–135 (2013)
13. Banharnsakun, A., Achalakul, T., Sirinaovakul, B.: The best-so-far selection in artificial bee colony algorithm. *Appl. Soft Comput.* **11**, 2888–2901 (2011)
14. Gordon, D.: The organization of work in social insect colonies. *Complexity* **8**(1), 43–46 (2003)
15. Lubin, T.B.: The evolution of sociality in spiders. In: Brockmann, H.J. (ed.) *Advances in the study of behavior*, vol. 37, pp. 83–145 (2007)
16. Uetz, G.W.: Colonial web-building spiders: Balancing the costs and. In: Choe, E.J., Crespi, B. (ed.) *The Evolution of Social Behavior in Insects and Arachnids*, pp. 458–475. Cambridge, Cambridge University Press, England
17. Aviles, L.: Sex-ratio bias and possible group selection in the social spider *Anelosimus eximius*. *Am. Nat.* **128**(1), 1–12 (1986)
18. Burgess, J. W.: Social spacing strategies in spiders. In: Rovner, P.N. (ed.) *Spider Communication: Mechanisms and Ecological Significance*, pp. 317–351. Princeton University Press, Princeton, New Jersey (1982)
19. Maxence, S.: Social organization of the colonial spider *Leucauge* sp. in the Neotropics: Vertical stratification within colonies. *The. J. Arachnology* **38**, 446–451 (2010)

20. Eric, C., Yip, K.S.: Cooperative capture of large prey solves scaling challenge faced by spider societies. *Proc. Natl. Acad. Sci. USA* **105**(33), 11818–11822 (2008)
21. Oster, G., Wilson, E.: Caste and ecology in the social insects. Princeton University Press, Princeton, N.J. (1978)
22. Hölldobler, Bert., Wilson, E.O.: Journey to the ants: a story of scientific exploration. ISBN 0-674-48525-4 (1994)
23. Hölldobler, Bert., Wilson, E.O.: The Ants. Harvard University Press, USA. ISBN 0-674-04075-9 (1990)
24. Avilés, L.: Causes and consequences of cooperation and permanent-sociality in spiders. In: Choe, B.C. (ed.) *The Evolution of Social Behavior in Insects and Arachnids*, pp. 476–498. Cambridge University Press, Cambridge, Massachusetts (1997)
25. Rayor, E.C.: Do social spiders cooperate in predator defense and foraging without a web? *Behav. Ecol. Sociobiol.* **65**(10), 1935–1945 (2011)
26. Gove, R., Hayworth, M., Chhetri, M., Rueppell, O.: Division of labour and social insect colony performance in relation to task and mating number under two alternative response threshold models. *Insect. Soc.* **56**(3), 19–331 (2009)
27. Ann, L., Rypstra, R.S.: Prey Size, prey perishability and group foraging in a social spider. *Oecologia* **86**(1), 25–30 (1991)
28. Pasquet, A.: Cooperation and prey capture efficiency in a social spider, *Anelosimus eximius* (Araneae, Theridiidae). *Ethology* **90**, 121–133 (1991)
29. Ulbrich, K., Henschel, J.: Intraspecific competition in a social spider. *Ecol. Model.* **115**(2–3), 243–251 (1999)
30. Jones, T., Riechert, S.: Patterns of reproductive success associated with social structure and microclimate in a spider system. *Anim. Behav.* **76**(6), 2011–2019 (2008)
31. Damian, O., Andrade, M., Kasumovic, M.: Dynamic population structure and the evolution of spider mating systems. *Adv. Insect Physiol.* **41**, 65–114 (2011)
32. Yang, X.-S.: Nature-inspired metaheuristic algorithms. Luniver Press, Beckington (2008)
33. Chen, D.B., Zhao, C.X.: Particle swarm optimization with adaptive population size and its application. *Appl. Soft Comput.* **9**(1), 39–48 (2009)
34. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341–359 (1995)
35. Yang, E., Barton, N.H., Arslan, T., Erdogan, A.T.: A novel shifting balance theory-based approach to optimization of an energy-constrained modulation scheme for wireless sensor networks. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008*, June 1–6, 2008, Hong Kong, China. pp. 2749–2756, IEEE (2008)
36. Duan, X., Wang, G.G., Kang, X., Niu, Q., Naterer, G., Peng, Q.: Performance study of mode-pursuing sampling method. *Eng. Optim.* **41**(1) (2009)
37. Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. *Evolutionary Computation*, 2004. CEC2004. Congress on, vol. 2, pp. 1980–1987, 19–23 June 2004
38. Mezura-Montes, E., Velázquez-Reyes, J., Coello Coello, C.A.: A comparative study of differential evolution variants for global optimization, In: *Proceedings of the 8th annual conference on Genetic and evolutionary computation (GECCO'06)*, pp. 485–492 ACM, New York, NY, USA (2006)
39. Karaboga, D., Akay, B.: A comparative study of Artificial Bee Colony algorithm, *Applied Mathematics and Computation*, vol. 214, Issue 1, 1 Aug 2009, pp. 108–132. ISSN 0096-3003 (2009)
40. Krishnanand, K.R., Nayak, S.K., Panigrahi, B.K., Rout, P.K.: Comparative study of five bio-inspired evolutionary optimization techniques, *Nature & Biologically Inspired Computing, NaBIC*, World Congress on, vol., pp. 1231–1236 (2009)
41. Ying, J., Ke-Cun, Z., Shao-Jian, Q.: A deterministic global optimization algorithm. *Appl. Math. Comput.* **185**(1), 382–387 (2007)

42. Rashedia, E., Nezamabadi-pour, H., Saryazdi, S.: Filter modeling using gravitational search algorithm. *Eng. Appl. Artif. Intell.* **24**(1), 117–122 (2011)
43. Wilcoxon, F.: Individual comparisons by ranking methods. *Biometrics* **1**, 80–83 (1945)
44. Garcia, S., Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special session on real parameter optimization. *J Heurist* (2008). doi:[10.1007/s10732-008-9080-4](https://doi.org/10.1007/s10732-008-9080-4)

Advances of Evolutionary Computation: Methods and Operators

Cuevas, E.; Díaz Cortés, M.A.; Oliva Navarro, D.A.

2016, XIV, 202 p. 48 illus., 5 illus. in color., Hardcover

ISBN: 978-3-319-28502-3