Expert Systems with Applications 41 (2014) 412–425

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A new algorithm inspired in the behavior of the social-spider for constrained optimization

Erik Cuevas *, Miguel Cienfuegos

Departamento de Electrónica, Universidad de Guadalajara, CUCEI, Av. Revolución 1500, Guadalajara, Jal, Mexico

ARTICLE INFO

Keywords:
Swarm algorithms
Constrained optimization
Bio-inspired algorithms

ABSTRACT

During the past decade, solving constrained optimization problems with swarm algorithms has received considerable attention among researchers and practitioners. In this paper, a novel swarm algorithm called the Social Spider Optimization (SSO-C) is proposed for solving constrained optimization tasks. The SSO-C algorithm is based on the simulation of cooperative behavior of social-spiders. In the proposed 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 gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors that are typically found in the colony. For constraint handling, the proposed algorithm incorporates the combination of two different paradigms in order to direct the search towards feasible regions of the search space. In particular, it has been added: (1) a penalty function which introduces a tendency term into the original objective function to penalize constraint violations in order to solve a constrained problem as an unconstrained one; (2) a feasibility criterion to bias the generation of new individuals toward feasible regions increasing also their probability of getting better solutions. In order to illustrate the proficiency and robustness of the proposed approach, it is compared to other well-known evolutionary methods. Simulation and comparisons based on several well-studied benchmarks functions and real-world engineering problems demonstrate the effectiveness, efficiency and stability of the proposed method.

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 aggregative conduct of insects or animals is known as swarm behavior. Entomologists have studied this collective phenomenon to model biological swarms while engineers have 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 with no supervision is frequently addressed 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" (Bonabeau, Dorigo, & Theraulaz, 1999). Swarm intelligence has some advantages such as scalability, fault tolerance, adaptation, speed, modularity, autonomy and parallelism (Kassabalidis, El-Sharkawi, Marks, Arabshahi, & Gray, 2001).

The key components of swarm intelligence are self-organization and labor division. In a self-organizing system, each of the covered units responds to local stimuli individually and may act together to accomplish a global task, via a labor separation which avoids a centralized supervision. The entire system can thus efficiently adapt to internal and external changes.

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 (Kennedy & Eberhart, 1995), the cooperative behavior of bee colonies such as the Artificial Bee Colony (ABC) technique (Karaboga, 2005), the social foraging behavior of bacteria such as the Bacterial Foraging Optimization Algorithm (BFOA) (Passino, 2002), the simulation of the herding behavior of krill individuals such as the Krill Herd (KH) method (Hossein & Hossein-Alavi, 2012), the mating behavior of firefly insects such as the Firefly (FF) method (Yang, 2010) and the emulation of the lifestyle of cuckoo birds such as the Cuckoo Optimization Algorithm (COA) (Rajabioun, 2011).

* Corresponding author. Tel.: +52 33 1378 5900x27714.
E-mail addresses: erik.cuevas@cucei.udg.mx (E. Cuevas), daniel.zaldivar@cucei.udg.mx, marco.perez@cucei.udg.mx (M. Cienfuegos).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.07.067

In particular, insect colonies and animal groups provide a rich set of metaphors for designing swarm optimization algorithms. Such cooperative entities are complex systems that are composed by individuals with different cooperative-tasks where each member tends to reproduce specialized behaviors depending on its gender (Bonabeau, 1998). However, most of swarm algorithms model individuals as unisex entities that perform virtually the same behavior. Under such circumstances, algorithms waste the possibility of adding 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 including population diversity and 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 (Wang et al., 2011; Wan-li & Mei-qing, 2013). The cause for such problems is associated to the operators that modify individual positions. In such algorithms, during their evolution, the position of each agent for the next iteration is updated yielding an attraction towards the position of the best particle seen so-far (in case of PSO) or towards other randomly chosen individuals (in case of ABC). As the algorithm evolves, those behaviors cause that the entire population concentrates around the best particle or diverges without control. It does favors the premature convergence or damage the exploration–exploitation balance (Banharnsakun, Achalakul, & Sirinaovakul, 2011; Wang, Sun, Li, Rahnamayan, & Jeng-shyang, 2013).

The interesting and exotic collective behavior of social insects have fascinated and attracted researchers for many years. The collaborative swarming behavior observed 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 (Gordon, 2003). Social-spiders are a representative example of social insects (Lubin, 2007). A social-spider is a spider species whose members maintain a set of complex cooperative behaviors (Uetz). 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 (Aviles, 1986). In a social-spider colony, each member, depending on its gender, executes a variety of tasks such as predation, mating, web design, and social interaction (Aviles, 1986; Burgess, 1982). The web it is an important part of the colony because it is not only used as a common environment for all members, but also as a communication channel among them (Maxence, 2010) Therefore, important information (such as trapped prays or mating possibilities) is transmitted by small vibrations through the web. 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 (Eric & Yip, 2008).

On the other hand, in real-world applications, most optimization problems are subject to different types of constraints. These kinds of problems are known as constrained optimization problems. A constrained optimization problem is defined as finding parameter vector $\mathbf{x}$ that minimizes an objective function $J(\mathbf{x})$ subject to inequality and/or equality constraints:

$$\begin{array}{lll} \text{minimize } J(\mathbf{x}), & \mathbf{x}=(x_1,\ldots,x_n)\in\mathbf{R}^n & \\ & l_i \leqslant x_i \leqslant u_i, & i=1,\ldots,n \\ \text{subject to}: & g_j(\mathbf{x})\leqslant 0 & \text{for } j=1,\ldots,q \\ & h_j(\mathbf{x})=0 & \text{for } j=q+1,\ldots,m \end{array} \tag{1}$$

The objective function J is defined on a search space, $\mathbf{S}$, which is defined as a n -dimensional rectangle in $\mathbf{R}$ ($\mathbf{S}\subseteq\mathbf{R}$). Domains of variables are defined by their lower and upper bounds (l_i and u_i). A feasible region $\mathbf{F}\subseteq\mathbf{S}$ is defined by a set of m additional constraints ($m\geqslant 0$) and $\mathbf{x}$ is defined on feasible space ($\mathbf{x}\in\mathbf{F}\in\mathbf{S}$). At any point $\mathbf{x}\in\mathbf{F}$, constraints g_j that satisfy $g_j(\mathbf{x})=0$ are called active constraints at $\mathbf{x}$. By extension, equality constraints h_j are also called active at all points of S (Michalewicz & Schoenauer, 1995). Constrained optimization problems are hard to optimization algorithms and also no single parameter (number of linear, nonlinear and active constraints, the ratio $\rho=|\mathbf{F}|/|\mathbf{S}|$, type of the function, number of variables) is proved to be significant as a major measure of difficulty of the problem (Michalewicz, Deb, Schmidt, & Stidsen, 1999).

Since most of the optimization algorithms have been primarily designed to address unconstrained optimization problems, constraint handling techniques are usually incorporated in the algorithms in order to direct the search towards the feasible regions of the search space. Methods dealing with the constraints were grouped into four categories (Koziel & Michalewicz, 1999): (i) methods based on preserving feasibility of solutions by transforming infeasible solutions to feasible ones with some operators; (ii) methods based on penalty functions which introduce a penalty term into the original objective function to penalize constraint violations in order to solve a constrained problem as an unconstrained one; (iii) methods that make a clear distinction between feasible and infeasible solutions; (iv) other hybrid methods combining evolutionary computation techniques with deterministic procedures for numerical optimization. Considering such mechanisms, several swarm algorithms have been modified to solve constrained optimization problems. Such methods include modified versions of PSO (He & Wang, 2007), ABC (Karaboga & Akay, 2011) and FF (Amir, Xin-She, & Amir, 2011). Each one of the four approaches employed to deal constrains presents advantages such as implementation-easiness and fast calculation whereas adversely posses disadvantages such as tuning difficulties and increase of function evaluations (Gan, Peng, Peng, Chen, & Inoussa, 2010). Therefore, it is reasonable to incorporate a combination of these approaches into a single algorithm in order to increase their potential and to eliminate their drawbacks.

In this paper, a novel swarm algorithm, called the Social Spider Optimization (SSO-C) is proposed for solving constrained optimization tasks. The SSO-C algorithm is based on the simulation of the cooperative behavior of social-spiders. In the proposed 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 gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors that are typical in a colony. For constraint handling, the proposed algorithm incorporates the combination of two different paradigms in order to direct the search towards feasible regions of the search space. In particular, it has been added: (1) a penalty function which introduces a tendency term into the original objective function to penalize constraint violations in order to solve a constrained problem as an unconstrained one; (2) a feasibility criterion to bias the generation of new individuals toward feasible regions increasing also their probability of getting better solutions. Different to most of existent swarm algorithms, in the proposed approach, each individual is modeled considering two 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 commonly present in the popular PSO and ABC algorithms, such as the premature convergence and the incorrect exploration–exploitation balance. In order to illustrate the proficiency and robustness of the proposed approach, it is compared to other similar methods. The comparison examines several standard constrained benchmark functions which are commonly considered in the literature. The results show a high performance of the proposed method for searching a global optimum in several benchmark functions and real-world engineering problems.

This paper is organized as follows. In Section 2, we introduce basic biological aspects of the algorithm. In Section 3, the novel SSO-C algorithm and its characteristics are both described. Section 4 presents the experimental results and the comparative study. Finally, in Section 5, conclusions are drawn.

2. Biological 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 allowing task specialization among group members (Hölldobler & Wilson, 1998; Oster & Wilson, 1978). A social insect colony functions as an integrated unit that not only possesses the ability to operate at a distributed manner, but also to undertake enormous construction of global projects (Hölldobler & Wilson, 1990). It is important to acknowledge that global order in social insects can arise as a result of internal interactions among members.

A few species of spiders have been documented exhibiting a degree of social behavior (Lubin, 2007). The behavior of spiders can be generalized into two basic forms: solitary spiders and social spiders (Aviles, 1986). This classification is made based on the level of cooperative behavior that they exhibit (Burgess, 1982). In one side, solitary spiders create and maintain their own web while live in scarce contact to other individuals of the same species. In contrast, social spiders form colonies that remain together over a communal web with close spatial relationship to other group members (Maxence, 2010). Fig. 1 presents two pictures that show different environments formed by the social spider.

A social spider colony is composed of two fundamental components: its members and the communal web. Members are divided into two different categories: males and females. An interesting characteristic of social-spiders is the highly female-biased population. Some studies suggest that the number of male spiders barely reaches the 30% of the total colony members (Aviles, 1986, 1997). In the colony, each member, depending on its gender, cooperate in different activities such as building and maintaining the communal web, prey capturing, mating and social contact (Eric & Yip, 2008). Interactions among members are either direct or indirect (Rayor, 2011). Direct interactions imply body contact or the exchange of fluids such as mating. For indirect interactions, the communal web is used as a "medium of communication" which conveys important information that is available to each colony member (Maxence, 2010). This information encoded as small vibrations is a critical aspect for the collective coordination among members (Eric & Yip, 2008). Vibrations are employed by the colony members to decode several messages such as the size of the trapped preys, characteristics of the neighboring members, etc. The intensity of such vibrations depend on the weight and distance of the spiders that have produced them.

In spite of the complexity, all the cooperative global patterns in the colony level are generated as a result of internal interactions among colony members (Gove, Hayworth, Chhetri, & Rueppell, 2009). Such internal interactions involve a set of simple behavioral rules followed by each spider in the colony. Behavioral rules are divided into two different classes: social interaction (cooperative behavior) and mating (Rypstra & Prey Size, 1991).

As a social insect, spiders perform cooperative interaction with 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 others, irrespectively of gender (Aviles, 1986). For a particular female spider, such attraction or dislike is commonly developed over other spiders according to their vibrations which are emitted over the communal web and represent strong colony members (Eric & Yip, 2008). Since the vibrations depend on the weight and distance of the members which provoke them, stronger vibrations are produced either by big spiders or neighboring members (Maxence, 2010). 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 (Eric & Yip, 2008).

Different to female spiders, the behavior of male members is reproductive-oriented (Pasquet, 1991). Male spiders recognize themselves as a subgroup of alpha males which dominate the colony resources. Therefore, the male population is divided into two classes: dominant and non-dominant male spiders (Pasquet, 1991). Dominant male spiders have better fitness characteristics (normally size) in comparison to non-dominant. In a typical behavior, dominant males are attracted to the closest female spider in the communal web. In contrast, non-dominant male spiders tend to concentrate upon the center of the male population as a strategy to take advantage of the resources wasted by dominant males (Ulbrich & Henschel, 1999).

Mating is an important operation that no 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 female members (Jones & Riechert, 2008). Under such circumstances, when a dominant male spider locates one or more female members within a specific range, it mates with all the females in order to produce offspring (Damian, Andrade, & Kasumovic, 2011).

(a) (b)

Fig. 1. Two pictures (a) and (b) that show different environments formed by the social spider.

3. The Social Spider Optimization (SSO-C) algorithm

In this paper, the operational principles from the social-spider colony have been used as guidelines for developing a new swarm optimization algorithm. The SSO-C assumes that entire search space is a communal web, where all the social-spiders interact to each other. In the proposed 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 is symbolized by the social-spider. The algorithm models two different search agents (spiders): males and females. Depending on gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors that are commonly assumed within 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. Therefore, N_f is calculated by the following equation:

$$N_f = \text{floor}[(0.9 - \text{rand} \cdot 0.25) \cdot N] \tag{2}$$

where rand is a random number between [0,1] whereas 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:

$$N_m = N - N_f \tag{3}$$

Therefore, the complete population $\mathbf{S}$, composed by N elements, is divided in two sub-groups $\mathbf{F}$ and $\mathbf{M}$. The Group $\mathbf{F}$ assembles the set of female individuals ($\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{N_f}\}$) whereas $\mathbf{M}$ groups the male members ($\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{N_m}\}$), where $\mathbf{S} = \mathbf{F} \cup \mathbf{M}$ ($\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N\}$), such that $\mathbf{S} = \{\mathbf{s}_1 = \mathbf{f}_1, \mathbf{s}_2 = \mathbf{f}_2, \ldots, \mathbf{s}_{N_f} = \mathbf{f}_{N_f}, \mathbf{s}_{N_f+1} = \mathbf{m}_1, \mathbf{s}_{N_f+2} = \mathbf{m}_2, \ldots, \mathbf{s}_N = \mathbf{m}_{N_m}\}$.

3.1. Penalty function and substituted function

Since the proposed approach aims to solve constrained optimization problems, the original objective function $J(\mathbf{x})$ is replaced by other substituted function $C(\mathbf{x})$ which considers the original objective function $J(\mathbf{x})$ minus a penalty function $P(\mathbf{x})$ that introduces a tendency term to penalize constraint violations produced by $\mathbf{x}$. Therefore, considering the constrained optimization problem defined in Eq. (1), the substituted function is defined as follows:

$$P(\mathbf{x}) = \mu \cdot \sum_{i=1}^{q} g_i^2(\mathbf{x}) + \nu \cdot \sum_{j=q+1}^{m} h_j^2(\mathbf{x}) \tag{4}$$
$$C(\mathbf{x}) = J(\mathbf{x}) + P(\mathbf{x})$$

where μ and ν represents the penalty coefficients which weight the relative importance of each kind of constraint. In this work, μ and ν are set to 1×10^3 and 10, respectively.

3.2. Fitness assignation

In the biological metaphor, the spider size is the characteristic that evaluates the individual capacity to perform better over its assigned tasks. In the proposed approach, every individual (spider) receives a weight w_i which represents the solution quality that corresponds to the spider i (irrespective of gender) of the population $\mathbf{S}$. In order to calculate the weight of every spider the next equation is used:

$$w_i = \frac{worst_{\mathbf{S}} - C(\mathbf{s})}{worst_{\mathbf{S}} - best_{\mathbf{S}}} \tag{5}$$

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

$$best_{\mathbf{S}} = \min_{k \in \{1,2,\ldots,N\}} (C(\mathbf{s}_k)) \quad \text{and} \quad worst_{\mathbf{S}} = \max_{k \in \{1,2,\ldots,N\}} (C(\mathbf{s}_k)) \tag{6}$$

3.3. 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 is encoded as small vibrations that are critical for the collective coordination of all individuals in the population. The vibrations depend on the weight and distance of the spider which has generated them. Since the distance is relative to the individual that provokes the vibrations and the member who detects them, members located 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:

$$Vib_{i,j} = w_j \cdot e^{-d_{i,j}^2} \tag{7}$$

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 perceived-vibrations by considering any pair of individuals, three special relationships are considered within the SSO-C approach:

1. Vibrations $Vibc_i$ are perceived by the individual i ($\mathbf{s}_i$) as a result of the information transmitted by the member $c(\mathbf{s}_c)$ who is an individual that has two important characteristics: it is the nearest member to i and possesses a higher weight in comparison to $i(w_c > w_i)$.
$$Vibc_i = w_c \cdot e^{-d_{i,c}^2} \tag{8}$$
2. The vibrations $Vibb_i$ perceived by the individual i as a result of the information transmitted by the member $b(\mathbf{s}_b)$, with b being the individual holding the best weight (best fitness value) of the entire population $\mathbf{S}$, such that $w_b = \max_{k \in \{1,2,\ldots,N\}}(w_k)$.
$$Vibb_i = w_b \cdot e^{-d_{i,b}^2} \tag{9}$$
3. The vibrations $Vibf_i$ perceived by the individual i ($\mathbf{s}_i$) as a result of the information transmitted by the member $f(\mathbf{s}_f)$, with f being the nearest female individual to i.
$$Vibf_i = w_f \cdot e^{-d_{i,f}^2} \tag{10}$$

Fig. 2 shows the configuration of each special relationship: (a) $Vibc_i$, (b) $Vibb_i$ and (c) $Vibf_i$.

3.4. Initializing the population

Like other evolutionary algorithms, the SSO-C is an iterative process whose first step is to randomly initialize the entire population (female and male). The algorithm begins by initializing the set $\mathbf{S}$ 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:

$$\begin{array}{ll} f_{i,j}^0 = p_j^{low} + \text{rand}(0,1) \cdot \left(p_j^{high} - p_j^{low}\right) & m_{k,j}^0 = p_j^{low} + \text{rand}(0,1) \cdot \left(p_j^{high} - p_j^{low}\right) \\ i = 1,2,\ldots,N_f;\ j = 1,2,\ldots,n & k = 1,2,\ldots,N_m;\ j = 1,2,\ldots,n \end{array} \tag{11}$$

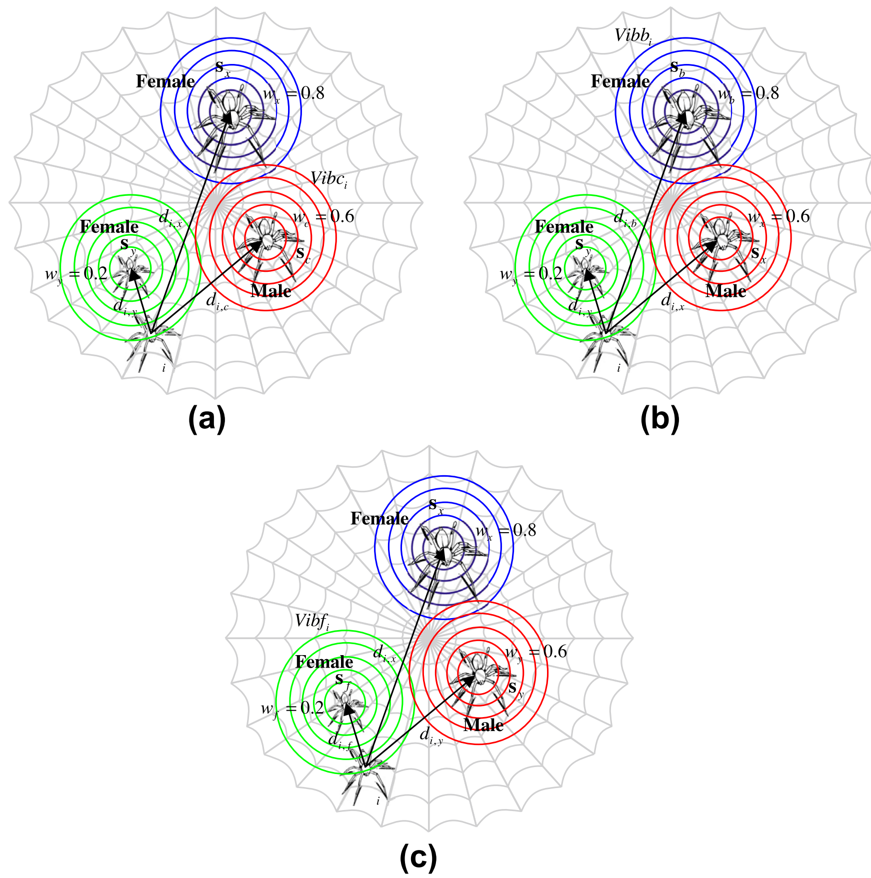


Fig. 2. Configuration of each special relation: (a) $Vibc_i$, (b) $Vibb_i$ and (c) $Vibf_i$.

where j, i and k are the parameter and individual indexes respectively whereas zero signals the initial population. The function rand $(0,1)$ generates a random number between 0 and 1. Hence, $f_{i,j}$ is the jth parameter of the ith female spider position.

3.5. Cooperative operators

3.5.1. 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 others irrespective of gender. For a particular female spider, such attraction or dislike is commonly developed over other spiders according to their vibrations which are emitted over the communal web. Since vibrations depend on the weight and distance of the members which have originated them, strong vibrations are produced either by big spiders or other neighboring members lying nearby the individual which is perceiving them. The final decision of attraction or dislike over a determined member is taken considering 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 that holds a higher weight and produces the vibration $Vibc_i$. The second one considers the change regarding the best individual of the entire population $\mathbf{S}$ who produces the vibration $Vibb_i$. Finally, the third one incorporates a random movement.

Since the final movement of attraction or repulsion depends on several random phenomena, the selection 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:

$$\mathbf{f}_i^{k+1} = \begin{cases} \mathbf{f}_i^k + \alpha \cdot Vibc_i \cdot \left(\mathbf{s}_c - \mathbf{f}_i^k\right) + \beta \cdot Vibb_i \cdot \left(\mathbf{s}_b - \mathbf{f}_i^k\right) + \delta \cdot \left(\text{rand} - \frac{1}{2}\right) & \text{with probability } PF \\ \mathbf{f}_i^k - \alpha \cdot Vibc_i \cdot \left(\mathbf{s}_c - \mathbf{f}_i^k\right) - \beta \cdot Vibb_i \cdot \left(\mathbf{s}_b - \mathbf{f}_i^k\right) + \delta \cdot \left(\text{rand} - \frac{1}{2}\right) & \text{with probability } 1 - PF \end{cases} \tag{12}$$

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 that holds 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 that holds the attraction or repulsion vector over the local best element $\mathbf{s}_c$ and the global best individual $\mathbf{s}_b$ seen so-far. This particular type of interaction avoids the quick concentration of particles at only one point and encourages each particle to search around the local candidate region within its neighborhood ($\mathbf{s}_c$), rather than interacting to 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 towards the global best position, making the algorithm less susceptible to premature convergence. Second, it encourages particles to explore their own neighborhood thoroughly before converging towards the global best position. Therefore, it provides the algorithm with global search ability and enhances the exploitative behavior of the proposed approach.

3.5.2. Male cooperative operator

According to the biological behavior of the social-spider, male population is divided into two classes: dominant and non-dominant male spiders. Dominant male spiders have better fitness characteristics (usually regarding the size) in comparison to 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 resources that are wasted by dominant males.

For emulating such cooperative behavior, the male members are divided into 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, those under the median value are labeled as non-dominant **ND** males. In order to implement such computation, the male population **M** ($\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{N_m}\}$) is arranged according to their weight value in decreasing order. Thus, the individual whose weight w_{N_f+m} is located in the middle is considered the median male member. Since indexes of the male population **M** in regard to the entire population **S** are increased 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:

$$\mathbf{m}_i^{k+1} = \begin{cases} \mathbf{m}_i^k + \alpha \cdot Vibf_i \cdot (\mathbf{s}_f - \mathbf{m}_i^k) + \delta \cdot (\text{rand} - \frac{1}{2}) & \text{if } w_{N_f+i} > w_{N_f+m} \\ \mathbf{m}_i^k + \alpha \cdot \left(\frac{\sum_{h=1}^{N_m} \mathbf{m}_h^k \cdot w_{N_f+h}}{\sum_{h=1}^{N_m} w_{N_f+h}} - \mathbf{m}_i^k \right) & \text{if } w_{N_f+i} \leqslant w_{N_f+m} \end{cases}, \tag{13}$$

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

By using this operator, two different behaviors are produced. First, the set **D** of particles is attracted to others in order to provoke mating. Such behavior allows incorporating diversity into the population. Second, the set **ND** of particles is attracted to the weighted mean of the male population **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.

3.6. 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:

$$r = \frac{\sum_{j=1}^{n} \left(p_j^{high} - p_j^{low} \right)}{2 \cdot n} \tag{14}$$

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

$$Ps_i = \frac{w_i}{\sum_{j \in \mathbf{T}^k} w_j}, \tag{15}$$

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

Once the new spider $\mathbf{s}_{new}$ is formed, it is compared to the spider $\mathbf{s}_{wo}$ holding the worst weight of the colony, where $w_{wo} = \min_{l \in \{1,2,\ldots,N\}}(w_l)$. In order to direct the search towards feasible regions of the search space, a feasibility criterion is incorporated to bias the generation of new spiders toward feasible regions increasing also their probability of getting better solutions. Such feasibility criterion allows choosing one of two different solutions ($\mathbf{s}_{new}$ or $\mathbf{s}_{wo}$) applying the two following rules:

- Between $\mathbf{s}_{new}$ and $\mathbf{s}_{wo}$, it is chosen the spider whose penalty function $P(\mathbf{s})$ present the lower value. Therefore, it is selected $\mathbf{s}_{new}$, if ($P(\mathbf{s}_{new}) < P(\mathbf{s}_{wo})$) or $\mathbf{s}_{wo}$ if ($P(\mathbf{s}_{new}) > P(\mathbf{s}_{wo})$).
- If ($P(\mathbf{s}_{new}) = P(\mathbf{s}_{wo})$), it is selected the spider with better weight. Thus, it is chosen $\mathbf{s}_{new}$, if ($w_{new} \geqslant w_{wo}$), otherwise $\mathbf{s}_{wo}$ is considered.

In case of $\mathbf{s}_{new}$ would be selected, the new spider $\mathbf{s}_{new}$ assumes the gender and index from the replaced spider $\mathbf{s}_{wo}$. Such fact assures that the entire population **S** maintains the original rate between female and male members.

In order to demonstrate the mating operation, Fig. 3 illustrates a simple optimization problem which can be stated as follows:

$$\begin{aligned} &\text{minimize}, && J(\mathbf{x}) = x_1^4 - 2x_1^2 x_2 + x_1^2 + x_1 x_2^2 - 2x_1 + 4 \\ &\text{subject to}: && -2 \leqslant x_1 \leqslant 4 \quad -2 \leqslant x_2 \leqslant 4 \\ & && g(\mathbf{x}) = 0.25x_1^2 + 0.75x_2^2 - 1 \leqslant 0 \\ & && h(\mathbf{x}) = x_1^2 + x_2^2 - 2 = 0 \end{aligned} \tag{16}$$

As an example, it is assumed a population **S** of eight different 2-dimensional members ($N = 8$), five females ($N_f = 5$) and three males ($N_m = 3$). Fig. 3(a) shows the initial configuration of the proposed example with three different female members $\mathbf{f}_2(\mathbf{s}_2)$ and $\mathbf{f}_4(\mathbf{s}_4)$ constituting the set $\mathbf{E}^2$ which is located inside of the influence range r of a dominant male $\mathbf{m}_2(\mathbf{s}_7)$. Then, the new candidate spider $\mathbf{s}_{new}$ is generated from the elements $\mathbf{f}_2$, $\mathbf{f}_4$ and $\mathbf{m}_2$ which constitute the set $\mathbf{T}^2$. 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 $\{f_{2,1}, f_{4,1}, m_{2,1}\}$. The value of the second decision variable $s_{new,2}$ is also chosen in the same manner. Table 1 shows the data for constructing the new spider through the roulette method. Once the new spider $\mathbf{s}_{new}$ is formed, the feasibility criterion is applied, considering the worst member $\mathbf{f}_5$ that is present in the population **S**. Since the penalty function $P(\mathbf{s}_{new})$ is lesser than $P(\mathbf{s}_5)$, $\mathbf{f}_5$ is replaced by $\mathbf{s}_{new}$. Therefore, $\mathbf{s}_{new}$ assumes the same gender and index from $\mathbf{f}_5$. Fig. 3(b) shows the configuration of **S** after the mating process.

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

3.7. Computational procedure

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

Step 1: Considering N as the total number of n-dimensional colony members, define the number of male N_m and females N_f spiders in the entire population **S**.

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

where rand is a random number between [0,1] whereas floor $(\cdot)$ maps a real number to an integer number.

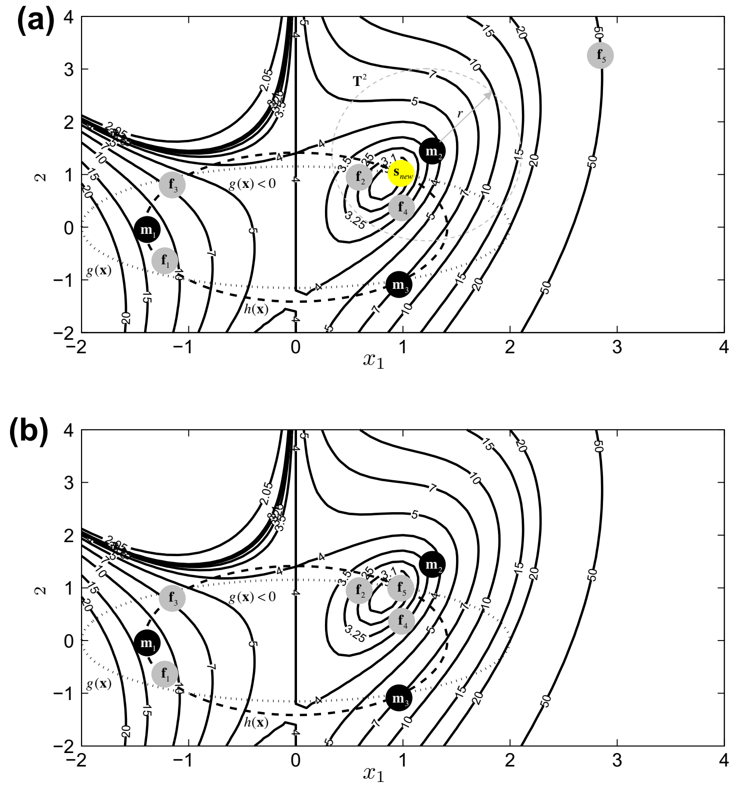
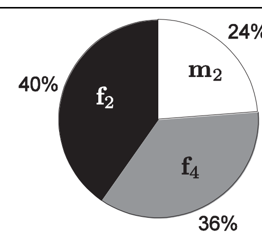


Fig. 3. Example of the mating operation: (a) initial configuration before mating and (b) configuration after the mating operation.

Table 1
Data for constructing the new spider $\mathbf{s}_{new}$ through the roulette method.

Spider		Position	w_i	Ps_i	$P(\mathbf{s})$	Roulette
$\mathbf{s}_1$	$\mathbf{f}_1$	$(-1.2,-0.8)$	0.88	–	0	
$\mathbf{s}_2$	$\mathbf{f}_2$	$(0.7,1)$	0.83	0.40	18.85	
$\mathbf{s}_3$	$\mathbf{f}_3$	$(-1.1,0.9)$	0.95	–	8.10	
$\mathbf{s}_4$	$\mathbf{f}_4$	$(1,0.3)$	0.74	0.36	474	
$\mathbf{s}_5$	$\mathbf{f}_5$	$(2.8,3.3)$	0	–	86002	
$\mathbf{s}_6$	$\mathbf{m}_1$	$(-1.4,0)$	0.79	–	260	
$\mathbf{s}_7$	$\mathbf{m}_2$	$(1.1,1.5)$	0.49	0.24	1002	
$\mathbf{s}_8$	$\mathbf{m}_3$	$(1,-1)$	1	–	0	
$\mathbf{s}_{new}$		$(1,1)$	1.2	–	0	

Step 2: Initialize randomly the female ($\mathbf{F}=\{\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_{N_f}\}$) and male ($\mathbf{M}=\{\mathbf{m}_1,\mathbf{m}_2,\ldots,\mathbf{m}_{N_m}\}$) members where $\mathbf{S}=\{\mathbf{s}_1=\mathbf{f}_1,\mathbf{s}_2=\mathbf{f}_2,\ldots,\mathbf{s}_{N_f}=\mathbf{f}_{N_f},\mathbf{s}_{N_f+1}=\mathbf{m}_1,\mathbf{s}_{N_f+2}=\mathbf{m}_2,\ldots,\mathbf{s}_N=\mathbf{m}_{N_m}\}$ and calculate the radius of mating.

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

for ($i=1$; $i<N_f+1$; i++)
for ($j=1$; $j<n+1$; j++)
$f_{i,j}^0=p_j^{low}+\mathrm{rand}(0,1)\cdot\left(p_j^{high}-p_j^{low}\right)$
end for
end for
for ($k=1$; $k<N_m+1$; k++)
for ($j=1$; $j<n+1$; j++)
$m_{k,j}^0=p_j^{low}+\mathrm{rand}\cdot\left(p_j^{high}-p_j^{low}\right)$
end for
end for

Step 3: Calculate the weight of every spider of **S** (Section 3.1).
for ($i=1$, $i<N+1$; i++)

$$w_i=\frac{worst_{\mathbf{S}}-C(\mathbf{s})}{worst_{\mathbf{S}}-best_{\mathbf{S}}}$$

where $C(\cdot)$ represent the substituted function, $best_{\mathbf{S}}=\min_{k\in\{1,2,\ldots,N\}}(J(\mathbf{s}_k))$ and $worst_{\mathbf{S}}=\max_{k\in\{1,2,\ldots,N\}}(J(\mathbf{s}_k))$.
end for

Step 4: Move female spiders according to the female cooperative operator (Section 3.5).
for ($i=1$; $i<N_f+1$; i++)
Calculate $Vibc_i$ and $Vibb_i$ (Section 3.3)
If ($r_m<PF$); where $r_m\in$ rand $(0,1)$
$\mathbf{f}_i^{k+1}=\mathbf{f}_i^k+\alpha\cdot Vibc_i\cdot\left(\mathbf{s}_c-\mathbf{f}_i^k\right)+\beta\cdot Vibb_i\cdot\left(\mathbf{s}_b-\mathbf{f}_i^k\right)+\delta\cdot\left(\mathrm{rand}-\frac{1}{2}\right)$
else if
$\mathbf{f}_i^{k+1}=\mathbf{f}_i^k-\alpha\cdot Vibc_i\cdot\left(\mathbf{s}_c-\mathbf{f}_i^k\right)-\beta\cdot Vibb_i\cdot\left(\mathbf{s}_b-\mathbf{f}_i^k\right)+\delta\cdot\left(\mathrm{rand}-\frac{1}{2}\right)$
end if
end for

Step 5: Move the male spiders according to the male cooperative operator (Section 3.5).
Find the median male individual (w_{N_f+m}) from **M**.
for ($i=1$; $i<N_m+1$; i++)
Calculate $Vibf_i$ (Section 3.3)
If ($w_{N_f+i}>w_{N_f+m}$)
$\mathbf{m}_i^{k+1}=\mathbf{m}_i^k+\alpha\cdot Vibf_i\cdot(\mathbf{s}_f-\mathbf{m}_i^k)+\delta\cdot(\mathrm{rand}-\frac{1}{2})$

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 the mating operation (Section 3.6).
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
Apply the feasibility criterion
In case of $\mathbf{s}_{new}$ would be selected, $\mathbf{s}_{new}$ assumes the gender and index from $\mathbf{s}_{wo}$
end if
end if
end for

Step 7: If the stop criteria is met, the process is finished; otherwise, go back to Step 3.

3.8. Discussion about the SSO-C 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 (Yang, 2008). 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 potential 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 (Chen & Zhao, 2009). So far, the exploration–exploitation dilemma has been an unsolved issue within the framework of evolutionary algorithms.

EA defines 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 (Wang et al., 2011; Wan-li & Mei-qing, 2013). Such problems arise from operators that modify individual positions. In such algorithms, 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 any other randomly chosen individual (in case of ABC). Such behaviors produce that the entire population concentrates around the best particle or diverges without control as the algorithm evolves, either favoring the premature convergence or damaging the exploration–exploitation balance (Banharnsakun et al., 2011; Wang et al., 2013).

Different to other EA, at SSO-C each individual is modeled considering the gender. Such fact allows incorporating computational mechanisms to avoid critical flaws such as premature convergence and incorrect exploration–exploitation balance commonly present in both, the PSO and the ABC algorithm. From an optimization point of view, the use of the social-spider behavior as a metaphor introduces interesting concepts in EA: the fact of dividing the entire population into different search-agent categories and the employment of specialized operators that are applied selectively to each of them. By using this framework, it is possible to improve the balance between exploitation and exploration, yet preserving the same population, i.e. individuals who have achieved efficient exploration (female spiders) and individuals that verify extensive exploitation (male spiders). Furthermore, the social-spider behavior mechanism introduces an interesting computational scheme with three important particularities: first, individuals are separately processed according to their characteristics. Second, operators share the same communication mechanism allowing the employment of important information of the evolutionary process to modify the influence of each operator. Third, although operators modify the position of only an individual type, they use global information (positions of all individual types) in order to perform such modification. Fig. 4 presents a schematic representation of the algorithm-data-flow. According to Fig. 4, the female cooperative and male cooperative operators process only female or male individuals, respectively. However, the mating operator modifies both individual types.

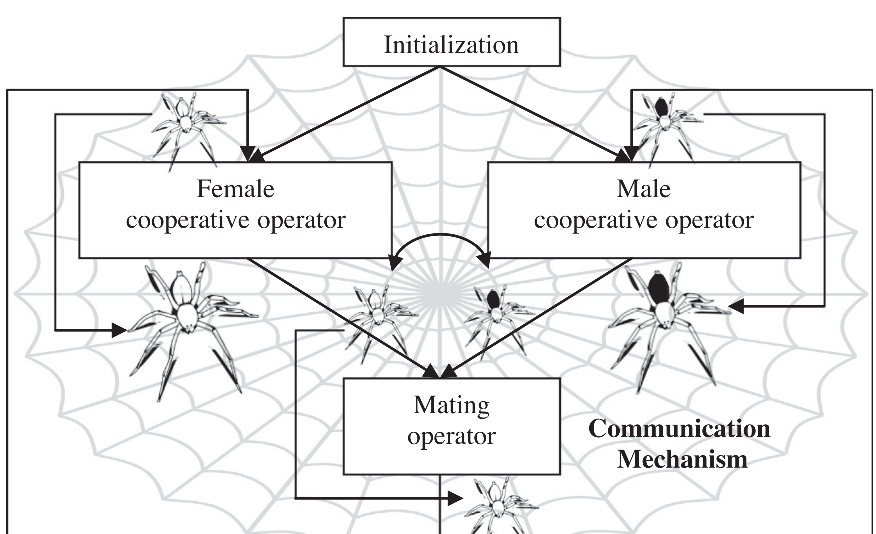


Fig. 4. Schematic representation of the SSO-C algorithm-data-flow.

4. Experimental results

In order to evaluate how well SSO-C performs on finding the global minimum of the benchmark constrained optimization problems, two categories of problems are considered in this paper. The first category includes the first 8 well-studied problems (Problems J_1–J_8) of CEC 2006 test suite (Liang et al., 2005). The mathematical expressions of these problems are provided in Appendix A. The second category includes a set of three well-studied real-world engineering optimization problems.

We have applied the SSO-C algorithm to the benchmark problems whose results have been compared to those produced by modified versions of PSO (He & Wang, 2007), ABC (Karaboga & Akay, 2011) and FF (Amir et al., 2011) which have been adapted to work over constrained optimization problems. Such approaches are considered as the most popular swarm algorithms for many optimization applications. The parameter setting for each algorithm in the comparison is described as follows:

1. PSO: The parameters are set to $M = 250$, $G_{\max} = 300$, $c_1 = 2$ and $c_2 = 2$; besides, the weight factor decreases linearly from 0.9 to 0.4 (He & Wang, 2007).
2. ABC: The algorithm has been implemented using the guidelines provided by its own reference (Karaboga & Akay, 2011), using $MR = 0.8$, $sn = 40$ and $MCN = 6000$.
3. FF (Amir et al., 2011): The parameters are set to Number of fireflies = 25, $\alpha = 0.001$, $q = 1.5$ and iteration number = 2000.
4. SSO-C: Once it has been determined experimentally, they are kept for all experiments in this section. Such parameters are set to $N = 50$, $PF = 0.7$, iteration number = 500.

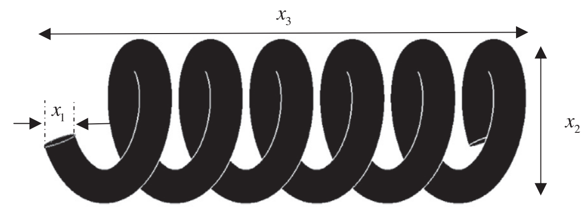


Fig. 5. A tension/compression string and its design features.

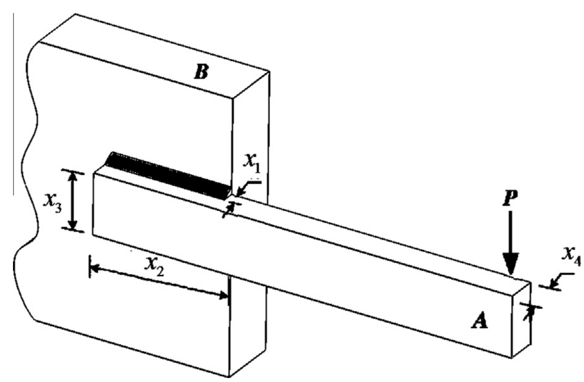


Fig. 6. A welded beam and its design features.

Table 2
Minimization results of benchmark functions of Table A.

Function	Optimal	Index	PSO	ABC	FF	SSO-C
J_1	−15.000	B	−15.000	−15.000	14.999	−15.000
		M	−15.000	−15.000	14.988	−15.000
		W	−15.000	−15.000	14.798	−15.000
		SD	0	0	6.4E−07	0
J_2	−0.803619	B	−0.80297	−0.803388	−0.803601	−0.803619
		M	−0.79010	−0.790148	−0.785238	−0.801563
		W	−0.76043	−0.756986	−0.751322	0.792589
		SD	1.2E−02	1.3E−02	1.67E−03	3.5E−05
J_3	−30665.539	B	−30665.501	−30665.539	−30664.322	−30665.539
		M	−30662.821	−30664.923	−30662.032	−30665.538
		W	−30650.432	−30659.131	−30648.974	−30665.147
		SD	5.2E−02	8.2E−02	5.2E−02	1.1E−04
J_4	−6961.814	B	−6961.728	−6961.814	−6959.987	−6961.814
		M	−6958.369	−6958.022	−6950.114	−6961.008
		W	−6942.085	−6955.337	−6947.626	−6960.918
		SD	6.7E−02	2.1E−02	3.8E−02	1.1E−03
J_5	24.306	B	24.327	24.48	23.97	24.306
		M	24.475	26.58	28.54	24.306
		W	24.843	28.40	30.14	24.306
		SD	1.32E−01	1.14	2.25	4.95E−05
J_6	−0.7499	B	−0.7499	−0.7499	−0.7497	−0.7499
		M	−0.7490	−0.7495	−0.7491	−0.7499
		W	−0.7486	−0.7490	−0.7479	−0.7499
		SD	1.2E−03	1.67E−03	1.5E−03	4.1E−09
J_7	0.0539415	B	0.05411	0.05394	0.05410	0.05394
		M	0.05416	0.05398	0.05417	0.05394
		W	0.05421	0.05411	0.05425	0.05394
		SD	1.35E−03	2.2E−03	3.1E−03	6.3E−09
J_8	961.715022	B	962.2132	961.9821	963.6281	961.9821
		M	963.9251	962.6421	965.4281	961.9987
		W	965.0251	964.2417	969.3217	962.0078
		SD	3.21	2.4	3.7	1.2

Table 3
Statistical features of the results obtained by various algorithms on the spring design problem.

	B	M	W	SD
PSO	0.01285757491586	0.014863120662235	0.019145260276060	0.001261916628168
ABC	0.01266523390012	0.012850718301673	0.013210405561696	0.000118451102725
FF	0.01266523390345	0.012930718743571	0.013420409563257	0.001453639618101
SSO-C	0.01266523278831	0.012764888178309	0.012867916581611	0.000092874978053

4.1. Comparisons on J_1–J_8 benchmark problems

In this section we evaluate the performance of our algorithm on 8 well-defined constrained optimization problems (Liang et al., 2005). These problems are widely recognized for having different properties and include various types of objective functions (i.e., linear, nonlinear, and quadratic) with different number of decision variables (n) and linear/nonlinear equalities/inequalities. All equality constraints $h_j(\mathbf{x}) = 0$, are converted into inequality constraints $|h_j(\mathbf{x})| \leqslant \varepsilon$ with $\varepsilon = 0.0001$. The simulations have been conducted in MATLAB and executed 30 times independently on each problem whereas it is reported the statistical features of the results obtained.

Table 2 summarizes the results obtained by all algorithms considering the benchmark problems $J_1 - J_8$. In this table, results are based on the best (B), mean (M), worst (W) and standard deviation (SD) of the lowest function values obtained by each algorithm. As can be seen from the results of Table 2, SSO-C always reaches the global optimum of all problems whereas in terms of the average and worst performance, SSO-C presents the best stability. Similarly, the proposed algorithm obtains the best precision, since the standard deviation (SD) presents the lower values.

4.2. Comparisons on mechanical engineering design optimization problems

In order to assess the performance of SSO-C on complex real word engineering problems, three well-studied problems are adopted from literature. These problems are: the tension/compression spring design optimization problem, the welded beam design optimization problem and the speed reducer design optimization problem. We compare the performance of with those produced by modified versions of PSO (He & Wang, 2007), ABC (Karaboga & Akay, 2011) and FF (Amir et al., 2011).

4.2.1. Tension/compression spring design optimization problem

The aim in this problem is to minimize the weight of a tension/compression spring (see Fig. 5) subject to constraints on minimum deflection, shear stress, surge frequency and limits on outside diameter (Coello, 2000). This problem has three continuous variables defined as: the wire diameter (x_1), the mean coil diameter (x_2) and the number of active coils (x_3). The constrained optimization problem can be defined as follows:

Table 4
Statistical features of the results obtained by various algorithms on the welded beam design optimization problem.

	B	M	W	SD
PSO	1.8464084393	2.011146016	2.237388681	0.108512649
ABC	1.7981726165	2.167357771	2.887044413	0.254266058
FF	1.7248541012	2.197401062	2.931001383	0.195264102
SSO-C	1.7248523085	1.746461619	1.799331766	0.025729853

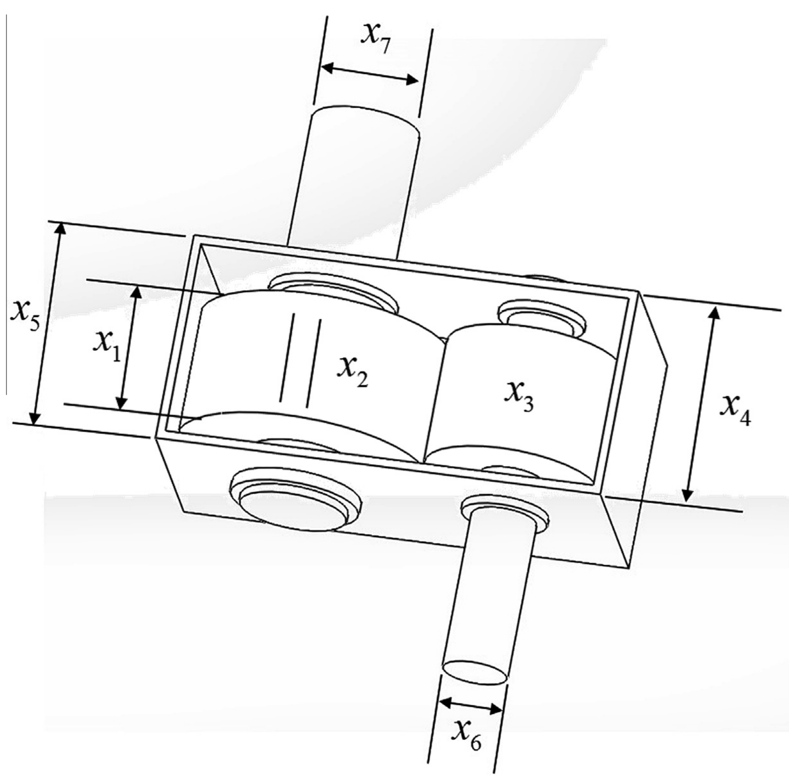


Fig. 7. The speed reducer and its design features.

Problem: Tension/compression spring design optimization problem

Minimize $J_{P1}(\mathbf{x}) = (x_3+2)x_2x_1^2$

Subject to:

$$g_1(\mathbf{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leqslant 0$$
$$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566x_2x_1^3 - x_1^4} + \frac{1}{5108x_1^2} \leqslant 0$$
$$g_3(\mathbf{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leqslant 0$$
$$g_4(\mathbf{x}) = \frac{x_2+x_1}{1.5} - 1 \leqslant 0$$

with boundary conditions $0.05 \leqslant x_1 \leqslant 2$, $0.25 \leqslant x_2 \leqslant 1.3$ and $2 \leqslant x_3 \leqslant 15$.

Results are tabulated in Table 3. It can be verified that the SSO-C presents a good performance whereas it maintains an acceptable stability. Moreover, in terms of the mean, worst, and standard deviation values, SSO-C dominates the 3 algorithms. However, there are other algorithms that can obtain along with SSO-C similar values, in terms of the best performance. The best solution obtained by SSO-C is $\mathbf{x}^*$ = (0.051689061657295, 0.356717753621158, 11.288964941289167) with $J_{P1}(\mathbf{x}^*)$ = 0.0126652327883194.

4.2.2. Welded beam design optimization problem

This problem is aimed to minimize the cost of a beam subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c) and end deflection of the beam (δ). The design variables are the thickness of the weld (x_1), length of the weld (x_2), width of the beam (x_3), and the beam thickness (x_4). Fig. 6 depicts a welded beam together with its design features. The mathematical formulation for this problem (Cagnina, Esquivel, & Coello, 2008) is described as follows:

Problem: Welded beam design optimization problem

Minimize $J_{P2}(\mathbf{x}) = 1.10471x_2x_1^2 + 0.04811x_3x_4(14+x_2)$

Subject to:

$$g_1(\mathbf{x}) = \tau(X) - 13600 \leqslant 0$$
$$g_2(\mathbf{x}) = \sigma(X) - 30000 \leqslant 0$$
$$g_3(\mathbf{x}) = x_1 - x_4 \leqslant 0$$
$$g_4(\mathbf{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14+x_2) - 5 \leqslant 0$$
$$g_5(\mathbf{x}) = 0.125 - x_1 \leqslant 0$$
$$g_6(\mathbf{x}) = \delta(X) - 0.25 \leqslant 0$$
$$g_7(\mathbf{x}) = 6000 - P_c(X) \leqslant 0$$

where:

$$\tau(X) = \sqrt{a^2 + 2ab(x_2/2R) + b^2}$$
$$a = \frac{6000}{\sqrt{2}x_1x_2} \quad b = \frac{6000 \cdot (14 + \frac{x_2}{2}) \cdot R}{2\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]}$$
$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$$
$$\sigma(X) = \frac{504000}{x_4x_3^2} \quad \delta(X) = \frac{65856000}{(30\times10^6)x_4x_3^3}$$
$$P_c(X) = \frac{4.013(30\times10^6)}{196}\sqrt{\frac{x_3^2x_4^6}{36}}\left[1 + \left(x_3\frac{\sqrt{\frac{30\times10^6}{4(12\times10^6)}}}{28}\right)\right]$$

with boundary conditions $0.1 \leqslant x_1 \leqslant 2$, $0.1 \leqslant x_2 \leqslant 10$, $0.1 \leqslant x_3 \leqslant 10$ and $0.1 \leqslant x_4 \leqslant 2$.

The experiment compares the SSO-C to other algorithms such as PSO and ABC and FF. The results for 30 runs are reported in Table 4. According to this table, SSO-C delivers better results than PSO, ABC in terms of the mean, worst, and standard deviation values. In particular, the test remarks the largest difference in performance which is directly related to a better trade-off between exploration and exploitation. The best solution obtained by SSO-C is $\mathbf{x}^*$ = (0.205729639786079, 3.470488665628002, 9.036623910357633, 0.205729639786080) with $J_{P2}(\mathbf{x}^*)$ = 1.724852308597365.

4.2.3. Speed reducer

A speed reducer is part of the gear box of mechanical systems, and it is also used for many other types of applications. The design of a speed reducer is considered as a challenging optimization problem in mechanical engineering (Jaberipour & Khorram, 2010). Such problem involves seven design variables which are represented in Fig. 7. These variables are the face width (x_1), the module of the teeth (x_2), the number of the teeth of pinion (x_3), the length of the first shaft between bearings (x_4), the length of the second shaft between bearings (x_5), the diameter of the first shaft (x_6) and the diameter of the second shaft (x_7).

The objective is to minimize the total weight of the speed reducer considering nine constraints and the physical limits of each variable. Therefore, the mathematical formulation can be summarized as follows:

Problem: Speed reducer problem

Minimize

$$J_{P3}(\mathbf{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2+x_7^2) + 7.477(x_6^3+x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

Subject to:

$$g_1(\mathbf{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leqslant 0$$
$$g_2(\mathbf{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leqslant 0$$
$$g_3(\mathbf{x}) = \frac{1.93}{x_2x_3x_4^3x_6^4} - 1 \leqslant 0$$
$$g_4(\mathbf{x}) = \frac{1.93}{x_2x_3x_5^3x_7^4} - 1 \leqslant 0$$
$$g_5(\mathbf{x}) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 1.69\times10^6}}{110x_6^3} - 1 \leqslant 0$$
$$g_6(\mathbf{x}) = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 157.5\times10^6}}{85x_7^3} - 1 \leqslant 0$$
$$g_7(\mathbf{x}) = \frac{x_2x_3}{40} - 1 \leqslant 0$$
$$g_8(\mathbf{x}) = \frac{5x_2}{x_1 - 1} - 1 \leqslant 0$$
$$g_9(\mathbf{x}) = \frac{x_1}{12x_2} - 1 \leqslant 0$$

with boundary conditions $2.6 \leqslant x_1 \leqslant 3.6$, $0.7 \leqslant x_2 \leqslant 0.8$, $17 \leqslant x_3 \leqslant 28$, $7.3 \leqslant x_4 \leqslant 8.3$, $7.3 \leqslant x_5 \leqslant 8.3$, $2.9 \leqslant x_6 \leqslant 3.9$ and $5.0 \leqslant x_7 \leqslant 5.5$.

Table 5 gives the comparison of the SSO-C results with the other methods considering 30 independent executions. Although all methods approximately reach the best value (the optimal solution), the SSO-C algorithm presents the best possible stability (**SD**). Such fact can be interpreted as the SSO-C capacity of obtaining the best value a higher number of times in comparison with the other algorithms. The best solution obtained by SSO-C is $\mathbf{x}^*$ = (3.500000, 0.70000, 17.00000, 7.30001, 7.71532, 3.35021, 5.28665) with $J_{P3}(\mathbf{x}^*)$ = 2996.11329802963.

Table 5
Statistical features of the results obtained by various algorithms on the speed reducer design optimization problem.

	B	M	W	SD
PSO	3044.45297067327	3079.26238870675	3177.51515620881	26.2173114313142
ABC	2996.11571099399	2998.06283682786	3002.75649061430	6.35456225271198
FF	2996.94726112401	3000.00542842129	3005.83626814072	8.35653454754367
SSO-C	2996.11329802963	2996.11329802963	2996.11329802963	1.33518419604E−12

5. Conclusions

In this paper, a novel swarm algorithm, called the Social Spider Optimization (SSO-C) has been proposed for solving constrained optimization tasks. The SSO-C algorithm is based on the simulation of the cooperative behavior of social-spiders. In the proposed 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 gender, each individual is conducted by a set of different evolutionary operators which mimic different cooperative behaviors that are typical in a colony.

For constraint handling, the proposed algorithm incorporates the combination of two different paradigms in order to direct the search towards feasible regions of the search space. In particular, it has been added: (1) a penalty function which introduces a tendency term into the original objective function to penalize constraint violations in order to solve a constrained problem as an unconstrained one; (2) a feasibility criterion to bias the generation of new individuals toward feasible regions increasing also their probability of getting better solutions.

In contrast to most of existent swarm algorithms, the proposed approach models each individual considering two genders. Such fact allows not only to emulate the cooperative behavior of the colony in a realistic way, but also to incorporate computational mechanisms to avoid critical flaws commonly delivered by the popular PSO and ABC algorithms, such as the premature convergence and the incorrect exploration–exploitation balance.

SSO-C has been experimentally tested considering a suite of 8 benchmark constrained functions and three real-word engineering problems. The performance of the proposed approach has been also compared to modified versions of PSO (He & Wang, 2007), ABC (Karaboga & Akay, 2011) and FF (Amir et al., 2011) which have been adapted to work over constrained optimization problems. Results have confirmed an acceptable performance of the proposed method in terms of the solution quality and stability.

The SSO's remarkable performance is associated with three different reasons: (i) their operators allow a better particle distribution in the search space, increasing the algorithm's ability to find the global optima; (ii) the division of the population into different individual types, provides the use of different rates between exploration and exploitation during the evolution process; and (iii) the constraint handling mechanism allows efficiently to conduct unfeasible solutions to feasible solutions even when new individuals are generated.

Acknowledgement

The proposed algorithm is part of the vision system used by a biped robot supported under the grant CONACYT CB 181053.

Appendix A. List of benchmark functions

See Table A.

Table A
Constrained test functions used in the experimental study.

Problem J_1

Minimize $J_1(\mathbf{x}) = 5\sum_{d=1}^{4} x_d - 5\sum_{d=1}^{4} x_d^2 - \sum_{d=5}^{13} x_d$

Subject to:

$g_1(\mathbf{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leqslant 0$

$g_2(\mathbf{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leqslant 0$

$g_3(\mathbf{x}) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leqslant 0$

$g_4(\mathbf{x}) = -8x_1 + x_{10} \leqslant 0$

$g_5(\mathbf{x}) = -8x_2 + x_{11} \leqslant 0$

$g_6(\mathbf{x}) = -8x_3 + x_{12} \leqslant 0$

$g_7(\mathbf{x}) = -2x_4 - x_5 + x_{10} \leqslant 0$

$g_8(\mathbf{x}) = -2x_6 - x_7 + x_{11} \leqslant 0$

$g_9(\mathbf{x}) = -2x_8 - x_9 + x_{12} \leqslant 0$

with boundary conditions $0 \leqslant x_d \leqslant 1$ $(d = 1,\ldots,9,13)$, $0 \leqslant x_d \leqslant 100$ $(d = 10,11,12)$. The optimum objective value is $J_1(\mathbf{x}^*) = -15$

Problem J_2

Minimize $J_2(\mathbf{x}) = -\left|\left(\sum_{d=1}^{n} \cos^4(x_d) - 2\prod_{d=1}^{n} \cos^2(x_d)\right) \Big/ \left(\sqrt{\sum_{d=1}^{n} d \cdot x_d^2}\right)\right|$

Subject to:

$g_1(\mathbf{x}) = 0.75 - \prod_{d=1}^{n} x_d \leqslant 0$

$g_2(\mathbf{x}) = \sum_{d=1}^{n} x_d - 0.75n \leqslant 0$

where $n = 20$ and $0 \leqslant x_d \leqslant 1$. The best objective value is $J_2(\mathbf{x}^*) = -0.803619$

Problem J_3

Minimize $J_3(\mathbf{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$

Subject to:

$g_1(\mathbf{x}) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.00022053x_3x_5 - 92 \leqslant 0$

$g_2(\mathbf{x}) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leqslant 0$

$g_3(\mathbf{x}) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leqslant 0$

$g_4(\mathbf{x}) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leqslant 0$

$g_5(\mathbf{x}) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leqslant 0$

$g_6(\mathbf{x}) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leqslant 0$

(continued on next page)

With boundary conditions $78 \leqslant x_1 \leqslant 102$, $33 \leqslant x_2 \leqslant 45$, and $27 \leqslant x_d \leqslant 45$ (d = 3, 4, 5). The optimum objective value is $J_3(\mathbf{x}^*) = -30665.53867$

Problem J_4

Minimize $J_4(\mathbf{x}) = (x_1 - 10)^3 + (x_2 - 20)^3$

Subject to: $g_1(\mathbf{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leqslant 0$

$g_2(\mathbf{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leqslant 0$

With boundary conditions $13 \leqslant x_1 \leqslant 100$ and $0 \leqslant x_2 \leqslant 100$. The optimum objective value is $J_4(\mathbf{x}^*) = -6961.81387$

Problem J_5

Minimize $J_5(\mathbf{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$

Subject to: $g_1(\mathbf{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leqslant 0$

$g_2(\mathbf{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leqslant 0$

$g_3(\mathbf{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leqslant 0$

$g_4(\mathbf{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leqslant 0$

$g_5(\mathbf{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x^4 - 40 \leqslant 0$

$g_6(\mathbf{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leqslant 0$

$g_7(\mathbf{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \leqslant 0$

$g_8(\mathbf{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leqslant 0$

where $-10 \leqslant x_d \leqslant 10$ (d = 1,...,10). The global objective value is $J_5(\mathbf{x}^*) = 24.306209$

Problem J_6

Minimize $J_6(\mathbf{x}) = x_1^2 + (x_2 - 1)^2$

Subject to: $h_1(\mathbf{x}) = x_2 - x_1^2 = 0$

With $-1 \leqslant x_d \leqslant 1$ (d = 1,2). The global objective value is $J_6(\mathbf{x}^*) = 0.7499$

Problem J_7

Minimize $J_7(\mathbf{x}) = e^{x_1x_2x_3x_4x_5}$

Subject to: $h_1(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$

$h_2(\mathbf{x}) = x_2x_3 - 5x_4x_5 = 0$

$h_3(\mathbf{x}) = x_1^3 + x_2^3 + 1 = 0$

With boundary conditions $-2.3 \leqslant x_d \leqslant 2.3$ (d = 1, 2), $-3.2 \leqslant x_d \leqslant 3.2$ (d = 3, 4, 5), and $10 \leqslant x_d \leqslant 1000$ (d = 4,...,8). The optimum objective value is $J_7(\mathbf{x}^*) = 0.0539415$

Problem J_8

Minimize $J_8(\mathbf{x}) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$

Subject to: $h_1(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 - 25 = 0$

$h_2(\mathbf{x}) = 8x_1 + 14x_2 + 7x_3 - 56 = 0$

where $0 \leqslant x_d \leqslant 10$, ($d$ = 1, 2, 3). The optimum objective value is $J_8(\mathbf{x}^*) = 961.715022289961$

References

Amir, G., Xin-She, Y., & Amir, A. (2011). Mixed variable structural optimization using Firefly Algorithm. *Computers and Structures, 89*, 2325–2336.

Aviles, L. (1986). Sex-ratio bias and possible group selection in the social spider *Anelosimus eximius*. *The American Naturalist, 128*(1), 1–12.

Avilés, L. (1997). Causes and consequences of cooperation and permanent-sociality in spiders. In B. C. Choe (Ed.), *The evolution of social behavior in insects and arachnids* (pp. 476–498). Cambridge, MA: Cambridge University Press.

Banharnsakun, A., Achalakul, T., & Sirinaovakul, B. (2011). The best-so-far selection in artificial bee colony algorithm. *Applied Soft Computing, 11*, 2888–2901.

Bonabeau, E. (1998). Social insect colonies as complex adaptive systems. *Ecosystems, 1*, 437–443.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*. New York, NY, USA: Oxford University Press, Inc..

Burgess, J. W. (1982). Social spacing strategies in spiders. In P. N. Rovner (Ed.), *Spider communication: Mechanisms and ecological significance* (pp. 317–351). Princeton, NJ: Princeton University Press.

Cagnina, L. C., Esquivel, S. C., & Coello, C. A. C. (2008). Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica, 32*, 319–326.

Chen, D. B., & Zhao, C. X. (2009). Particle swarm optimization with adaptive population size and its application. *Applied Soft Computing, 9*(1), 39–48.

Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry, 41*, 113–127.

Damian, O., Andrade, M., & Kasumovic, M. (2011). Dynamic population structure and the evolution of spider mating systems. *Advances in Insect Physiology, 41*, 65–114.

Eric, C., & Yip, K. S. (2008). Cooperative capture of large prey solves scaling challenge faced by spider societies. *Proceedings of the National Academy of Sciences of the United States of America, 105*(33), 11818–11822.

Gan, M., Peng, H., Peng, Xi., Chen, X., & Inoussa, G. (2010). An adaptive decision maker for constrained evolutionary optimization. *Applied Mathematics and Computation, 215*, 4172–4184.

Gordon, D. (2003). The organization of work in social insect colonies. *Complexity, 8*(1), 43–46.

Gove, R., Hayworth, M., Chhetri, M., & Rueppell, O. (2009). Division of labour and social insect colony performance in relation to task and mating number under two alternative response threshold models. *Insectes Sociaux, 56*(3), 19–331.

He, Q., & Wang, L. (2007). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation, 186*, 1407–1422.

Hölldobler, B., & Wilson, E. O. (1998). Journey to the Ants: A Story of Scientific Exploration, Belknap Press of Harvard University Press, ISBN 0-674-48525-4.

Hölldobler, B., & Wilson, E. O. (1990). *The ants*. 0-674-04075-9. Harvard University Press.

Hossein, A., & Hossein-Alavi, A. (2012). Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation, 17*, 4831–4845.

Jaberipour, M., & Khorram, E. (2010). Two improved harmony search algorithms for solving engineering optimization problems. *Communications in Nonlinear Science and Numerical Simulation, 15*, 3316–3331.

Jones, T., & Riechert, S. (2008). Patterns of reproductive success associated with social structure and microclimate in a spider system. *Animal Behaviour, 76*(6), 2011–2019.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report-TR06. Engineering Faculty, Computer Engineering Department, Erciyes University.

Karaboga, D., & Akay, B. (2011). A modified artificial bee colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing, 11*, 3021–3031.

Kassabalidis, I., El-Sharkawi, M. A., Marks, R. J., II, Arabshahi, P., & Gray, A. A. (2001). Swarm intelligence for routing in communication networks. *Global telecommunications conference, GLOBECOM '01* (Vol. 6, pp. 3613–3617). IEEE.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the 1995 IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948).

Koziel, S., & Michalewicz, Z. (1999). Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary Computation, 7*(1), 19–44.

Liang, J. J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan1, P. N., & Coello, C. A. C., et al. (2005). Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Technical report #2006005. Singapore: Nanyang Technolgical University.

Lubin, T. B. (2007). The evolution of sociality in spiders. In H. J. Brockmann (Ed.), *Advances in the study of behavior* (Vol. 37, pp. 83–145).

Maxence, S. (2010). Social organization of the colonial spider Leucauge sp. in the Neotropics: Vertical stratification within colonies. *The Journal of Arachnology, 38*, 446–451.

Michalewicz, Z., Deb, K., Schmidt, M., & Stidsen, T. (1999). Evolutionary algorithms for engineering applications. In K. Miettinen, P. Neittaanmäki, M. M. Mäkelä, & J. Périaux (Eds.), *Evolutionary algorithms in engineering and computer science* (pp. 73–94). Chichester, England: John Wiley and Sons.

Michalewicz, Z., & Schoenauer, M. (1995). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation, 4*(1), 1–32.

Oster, G., & Wilson, E. (1978). *Caste and ecology in the social insects*. Princeton, NJ: Princeton University press.

Pasquet, A. (1991). Cooperation and prey capture efficiency in a social spider, Anelosimus eximius (Araneae, Theridiidae). *Ethology, 90*, 121–133.

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine, 22*(3), 52–67.

Rajabioun, R. (2011). Cuckoo optimization algorithm. *Applied Soft Computing, 11*, 5508–5518.

Rayor, E. C. (2011). Do social spiders cooperate in predator defense and foraging without a web? *Behavioral Ecology & Sociobiology, 65*(10), 1935–1945.

Rypstra, Ann L., & Prey Size, R. S. (1991). Prey perishability and group foraging in a social spider. *Oecologia, 86*(1), 25–30.

Uetz, G. W. Colonial web-building spiders: Balancing the costs and. In E. J., Choe & B. Crespi (Eds.), *The evolution of social behavior in insects and arachnids* (pp. 458–475). Cambridge, England: Cambridge University Press.

Ulbrich, K., & Henschel, J. (1999). Intraspecific competition in a social spider. *Ecological Modelling, 115*(2–3), 243–251.

Wang, Y., Li, B., Weise, T., Wang, J., Yuan, B., & Tian, Q. (2011). Self-adaptive learning based particle swarm optimization. *Information Sciences, 181*(20), 4515–4538.

Wang, H., Sun, H., Li, C., Rahnamayan, S., & Jeng-shyang, P. (2013). Diversity enhanced particle swarm optimization with neighborhood. *Information Sciences, 223*, 119–135.

Wan-li, X., & Mei-qing, A. (2013). An efficient and robust artificial bee colony algorithm for numerical optimization. *Computers & Operations Research, 40*, 1256–1265.

Yang, X.-S. (2008). *Nature-inspired metaheuristic algorithms*. Beckington: Luniver Press.

Yang, X. S. (2010). *Engineering optimization: An introduction with metaheuristic applications*. John Wiley & Sons.