Spider Monkey Optimization

June 10, 2023

1 Introduction

Spider monkey optimization(SMO) is a swarm optimization algorithm inspired by the Fussion-Fission Social(FFS) structure of spider monkeys.
The social organization and behavior of spider monkeys can be understood through the following facts:

- Spider monkeys live in a group of about 40-50 individuals.
- All individuals in this community forage in small groups by going off in different directions during the day and everybody share the foraging experience in the night at their habitat.
- The lead female spider monkey decides the foraging route.
- If the leader does not find sufficient food then she divides the group into smaller groups and these groups forage, separately.

The image below shows the communication and organization of the spider monkey during forging:

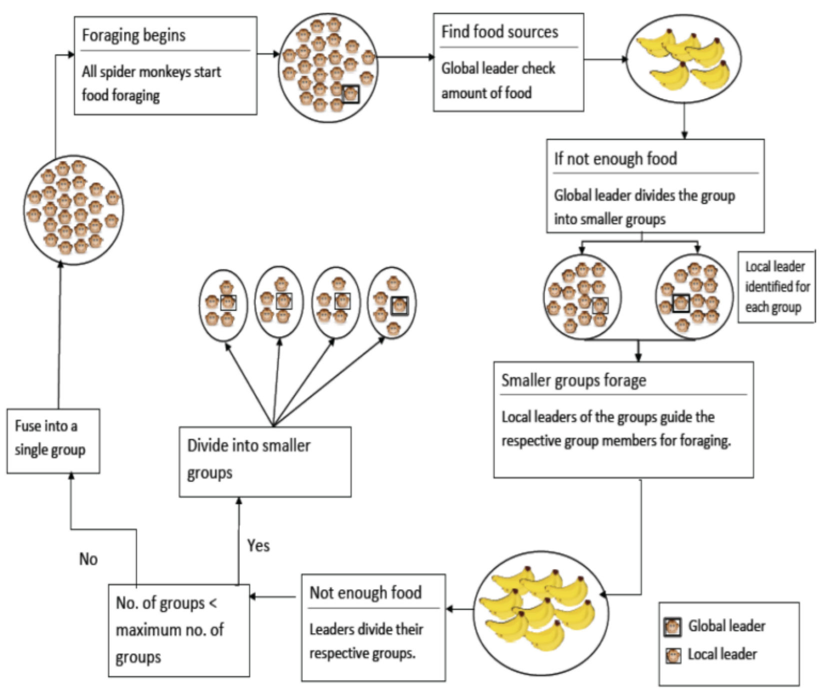

Figure 1: Foraging behavior of spider monkeys

2 SMO Process

The whole group has a global leader and this is divided into smaller groups and each one of them has a local leader
In SMO a spider monkey(SM) represents a potential solution. SMO consists of six phases:

- Local Leader Phase
- Global Leader Phase
- Local Leader Learning Phase
- Global Leader Learning Phase
- Local Leader Decision Phase
- Global Leader Decision Phase

2.1 Initialization

The swarm is initialized with N spider monkeys and SM_i represents the i^{th} spider monkey. Each SM_i is initialized as: Where SM_{minj} and SM_{maxj} are lower and upper bounds of the search space in the

$$SM_{ij} = SM_{minj} + U(0,1) \times (SM_{maxj} - SM_{minj})$$

Figure 2: Initialization

jth dimension and $U(0,1)$ is a uniformly distributed random number in the range $(0,1)$.

2.2 Local Leader Phase

The local Leader Phase is the vital phase of SMO, all the spider monkeys get a chance to update their position in this phase and the update is based on the local leader and local group member's experiences. Here the update equation is:

$$SMnew_{ij} = SM_{ij} + U(0,1) \times (LL_{kj} - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij})$$

Figure 3: LLP update equation

Where, SM_{ij} is the j^{th} dimension of the local leader of the k^{th} group and SM_{rj} is the j^{th} dimension of a randomly selected SM from the k^{th} group such that $r \neq i$ and $U(-1,1)$ is a uniformly distributed random number in the range $(-1,1)$.
Second term in the equation attracts the spider monkey towards the local leader and the third term introduces fluctuations so that premature stagnation can be avoided.

Pseudo code for LLP:

***Algorithm 1:** Position update process in Local Leader Phase (LLP)*

---

for each member$SM_i \in k^{th}$ group **do**
    for each $j \in \{1, \dots, D\}$**do**
    if $U(0,1) \geq pr$**then**

$$SMnew_{ij} = SM_{ij} + U(0,1)\times(LL_{kj} - SM_{ij}) + U(-1,1)\times(SM_{rj} - SM_{ij})$$

    else
    $SMnew_{ij} = SM_{ij}$
    end if
    end for
end for

Figure 4: Algorithm 1

Here pr represents the perturbation rate whose value generally lies in the range $[0.1, 0.8]$.

2.3 Global Leader Phase

After completing the LLP the algorithm takes the step towards the global leader phase.
The solutions are updated based on a selection probability, which is a function of the fitness and fitness can be calculated using the objective function f_i.

$$fitness\ function = fit_i = \begin{cases} \frac{1}{1+f_i}, & if\ f_i \geq 0 \\ 1 + abs(f_i), & if\ f_i < 0 \end{cases} \qquad (3)$$

Figure 5: Fitness Function

$$prob_i = \frac{fitness_i}{\sum_{i=1}^{N} \text{fitness}_i} \text{ or}$$

$$prob_i = 0.9\times \frac{fit_i}{\max\_fit} + 0.1$$

Figure 6: Probability

update function:

$$SMnew_{ij} = SM_{ij} + U(0,1)\times(GL_j - SM_{ij}) + U(-1,1)\times(SM_{rj} - SM_{ij})$$

Figure 7: Update function for GLP

where GL_j is the position of global leader in the j^{th} dimension.
The second term shows the attraction of parent SM towards the global leader and the last term reduces the chance of being stuck in a local optima.

Pseudo code for GLP:

Algorithm 2: *Position update process in Global Leader Phase (GLP)*

$count = 0$;
while $count < group$ size **do**
for each member $SM_i \in$ group **do**
  if $U(0,1) < prob_i$ **then**
  $count = count + 1$
  Randomly select $j \in \{1, \dots, D\}$
  Randomly select $SM_r \in$ group s.t. r≠ i
  $SMnew_{ij} = SM_{ij} + U(0,1)\times(GL_j - SM_{ij}) + U(-1,1)\times(SM_{rj} - SM_{ij})$
  end if
 end for
end while

Figure 8: Algorithm 2

In this phase solutions of high fitness will get more chance as compared to less fit solutions to update it's position.

2.4 Global Leader Learning Phase

In this phase, the algorithm finds the best solution of the entire swarm. The identified SM is considered as the global leader of the swarm. Further, the position of the global leader is checked and if it is not updated then the counter associated with the global leader, named as Global Leader Count(GLC), is incremented by 1, otherwise it is set to 0.
Global Limit Count is checked for a global leader and is compared with Global Leader Limit(GLL).

2.5 Local Leader Learning Phase

In this segment of the algorithm, the position of the local leader gets updated by applying a greedy selection among the group members. If the local leader doesn't update its position then a counter associated with the local leader called Local Limit Count(LLC) is incremented by 1; otherwise, the counter is set to 0. This process is applied to every group to find its respective local leader.
Local Limit Count is a counter that gets incremented till it reaches a fix threshold called Local Leader Limit(LLL).

2.6 Local Leader Decision Phase

If the LLC reaches the Local Leader Limit the solutions of this group are repelled from the existing local leader as it is exhausted(not updated up to LLL number of iterations).
Update equation:

$$SMnew_{ij} = SM_{ij} + U(0,1)\times(GL_j - SM_{ij}) + U(0,1)\times(SM_{rj} - LL_{kj})$$

Figure 9: Update equation for LLD

Solutions are attracted towards the global leader to change the existing search directions and positions. Further based on pr some dimensions of the solutions are randomly initialized.
LLL is calculated as $D \times N$ where D is the dimension and N is total number of SMs.

Pseudo Code for LLD:

Algorithm 3: *Local Leader Decision Phase (LLD):*

If $LocalLimitCount > LocalLeaderLimit$ **then**
  $LocalLimitCount = 0$
for each $j \in \{1, \dots, D\}$ **do**
  if $U(0,1) \geq pr$ **then**
    $SMnew_{ij} = SM_{minj} + U(0,1) \times (SM_{maxj} - SM_{minj})$
  else
    $SMnew_{ij} = SM_{ij} + U(0,1) \times (GL_j - SM_{ij}) + U(0,1) \times (SM_{rj} - LL_{kj})$
  end if
end for
end if

Figure 10: Algorithm 3

2.7 Global Leader Decision Phase

Similar to LLD, if the Global leader does not get recognized to a particular verge known as the Global Leader Limit the global leader divides the swarm into smaller groups if the existing number of groups is less than the pre-defined maximum number of groups else it will fuse all the groups into one unit group. GLL varies in the range of $N/2$ to $2N$.
Pseudo Code for GLD:

Algorithm 4 *Global Leader Decision Phase (GLD):*

if $GlobalLimitCount > GlobalLeaderLimit$ **then**
  $GlobalLimitCount = 0$
if $Number\ of\ groups < MG$ **then**
  Divide the swarms into groups
else
  Combine all the groups to make a single group
end if
  update Local Leaders position
end if

Figure 11: Algorithm 4

The following steps show the complete working mechanism of SMO to solve an optimization problem.

Algorithm 5: *Spider Monkey Optimization*

Step 1. **Initialize** population, local leader limit, global leader limit and perturbation rate *pr*;
Step 2. **Evaluate** the population;
Step 3. **Identify** global and local leaders;
Step 4. **Position update** by local leader phase (Algorithm 1);
Step 5. **Position update** by global leader phase (Algorithm 2);
Step 6. **Learning** through global leader learning phase;
Step 7. **Learning** through local leader learning phase;
Step 8. **Position update** by local leader decision phase (Algorithm 3);
Step 9. **Decide fission or fusion** using global leader decision phase (Algorithm 4);
Step 10. If termination condition is satisfied stop and declare the global leader position as the optimal solution else go to step 4.

Figure 12: Algorithm 5

3 Conclusion

SMO balances exploitation and exploration while avoiding stagnation at a local or global level. This makes SMO a better candidate among search-based optimization algorithms. For continuous optimization problems, SMO should be preferred over Particle Swarm Optimization(PSO), Artificial Bee Colony(ABC), or Differential Evolution(DE).SMO also outpaced these three in terms of dependability, effectiveness, and precision.
SMO has mainly four new control parameters: Local Leader Limit(LLL), Global Leader Limit(GLL), maximum no.of groups(MG), and perturbation rate pr. The presence of a large number of user-dependent parameters in SMO is a matter of concern for further research.