



Variants of Swarm Optimization Techniques

¹M. Sivasankari and ²R. Anandan

¹Department of Computer Science and Engineering,
Vels Institute of Science,
Technology and Advanced Studies,
Chennai.

²Department of Computer Science and Engineering,
Vels Institute of Science,
Technology and Advanced Studies,
Chennai.

Abstract

A Swarm depicts a group of homogeneous objects working together towards a common goal. Swarm based algorithm takes inspiration from the nature and behavior of Swarms. It can be used in various optimization techniques to produce low cost, robust solutions. Each particle is an agent which has limited abilities on its own, but their efforts as a group, will deliver fruitful outcomes and accomplish the targets.

This research paper constitutes the Swarm Intelligence overview and the Optimization techniques.

Bio Computing is an alternative to the traditional computing techniques as its well defined, and better computable in a reasonable time with serial computers. It also showcases better predictability. The foundation of Swarm Intelligence is the Inspiration, Identification and Application of nature's events. The fields of application include Robotics/Artificial Intelligence, Process Optimization and even Entertainment.

We will further divulge into the different techniques available and its merits/demerits.

1. Introduction

Computational Intelligence is a broader field of AI that comprises of the paradigms that relate to some kind of naturally occurring systems. Within CI, the Swarm Intelligence (SI) systems are characteristically a group of unpretentious agents intermingling with each other and with their location. The two principles of SI is self organization and stigmergy (Simulation by work). The features of Stigmergy is the Agent Communication in an indirect manner, for alteration of the environment. The similar and simple rules might create an varied array of designs based on the environmental states. The remainder of the paper deals with the different varieties of Swarm systems and the algorithms inspired by their nature

The different types of algorithm handled are Ant Colony Optimization, Bee Colony, Cuckoo Search, Gravitational Search, Intelligent Water Drop, Stochastic Diffusion Search and finally the Particle Swarm Optimization.

Ant Colony Optimization

The models and Techniques have taken the inspiration from a colony of Ants and has been fruitful in general purpose optimization. It is also the most studied technique in SI. The ants track their paths using the pheromones, which is used to mark the best path and that is then followed by the others in the Colony. Goss et al. [2] created a model of the

Common Behavior of Ants

We consider the assumption that the ants take a longer path in time m_1 and the shorter path in time m_2 ,

P_1 is the probability of an ant choosing the initial bridge

$$P_1 = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h},$$

Where parameters k and h are based on the experimental results

and, $p_2 = 1 - p_1$.

The Ant Colony Optimization Metaheuristic is used for a combinatorial optimization technique. Metaheuristic framework works along with lesser modifications and is relevant for all optimization problems. The adequate model for this is

$$P = (S, C, f)$$

Her the search space is represented by S

C denotes the constraints we have on the variables

F denotes the objective function and the goal is to maximise it

The above theorem represents that pheromone model of Ant Colony Optimization and the value is associated with the solution element. It refers to the probable solutions.

We create the solution in Ant Colony Optimization by using a fully connected construction graph and traversing along.

The graph is represented as

$$GC(V, E),$$

Where V = Set of vertices

E = set of edges

The components of the solution can contain either Vertices or Edges.

Algorithm

Assign the parameters and initialize the trail of the pheromone

While (Termination Condition is not met) Do
 Construction of Ant Solution
 PerformLocal Search
 Updation of Pheromone Result

End While

Applications of ACO

Majority of applications involve the NP Hard problems. This will identify the best solution for even the problems having the exponential time worst case complexity. We also use this ACO methodology i.e. The Dynamic Shortest Path in solving Telecommunication networking problem

It has also been very effective in routing problems in Telecommunication. Current trend is to concentrate more on problems involving more than one objectives, dynamic alterations of the data.

Bee Colony Optimization

This is also a meta-heuristic technique uses in the optimization problems. The artificial bee colony optimization is used when we have incomplete information. There is no global control and is based on cooperation. This enables the bees to achieve what they cannot do individually. Parallel variants are also developed which increases the speed.

Main step in the algorithm was developed on the basis of foraging behavior of the bees. Foraging provides the answer and waggle dance will assess the performance of the existing solution and direct the generation to new ones.

Information exchange and collective knowledge helps identify promising areas and discard the less promising solutions.

BCO is effective to identify solution for various combinatorial problems. The two steps involved out of which the first is to build the solution piece by piece. The second is to improve the complete solutions. Each phase has a forward and a backward pass. The NC steps are finished and the best of all the B solutions is assessed and finalised. Then the global best is identified. The maximum iteration is done and is stopped when we have no more perfection or improvements on the objective function value.

The bees would make a loyalty decision based on the after comparing with other existing solutions and then arrive into a conclusion of their solution.

Lets calculate the chance/probability of a bee that its faithful to the previously created solution i.e. the first forward pass:

$$p_b^{u+1} = e^{-\frac{O_{\max} - O_b}{u}}, \quad b = 1, 2, \dots, B$$

where:

O_b - represents the normalized/standardized value of the solution the bee in the b th position;

$O_{\max}$ —denotes the highest among normalized values of solutions be it partial or complete;

u —this represents the number of forward passes

Pseudo Code is given below

Initialization: Collect the data, input parameters (D and N), and stopping criteria.

Do

(1) Assign empty solution a.

(2) For (i = 0; i < N; i + +)

//below is the solution for forward pass

(a) For (d = 0; d < D; d + +)

For (s = 0; s < f(N); s + +)//counting the moves

(i) Evaluate possible moves;

(ii) Choose one move using the roulette wheel;

//below is the solution for backward pass

(b) For (d = 0; d < D; d + +)

Evaluate the (partial/complete) solution of bee d;

(c) For (d = 0; d < D; d + +)

Loyalty decision for bee d;

(d) For (d = 0; d < D; d + +)

If (d is uncommitted), choose a recruiter by the roulette wheel.

*(3) Evaluate all solutions and find the best one. Update bestsol and f (bestsol)
 while stopping criterion is not satisfied.
 return (bestsol; f (bestsol))*

Each implementation of BCO would be personalized for a particular problem. It would be then executed until the stopping criteria is met. Final solution is the best solution found before that's met.

To conclude, the Bee Colony Optimization technique has the ability to expand the search to the capable regions through exchange of valuable information. The modified process is implemented by limiting the search in different iterations. [3]

Stochastic Diffusion Search

It is a multi-agent optimisation algorithm formed of two steps, testing and diffusion. The test phase calculates an individual's objective value, and the diffusion phase controls decentralized communication between agents. The standard SDS algorithm comprises a population of agents with each agent's features initialized randomly within a search space. These features serve as the agent's hypothesis. The test phase calculates whether their hypothesis is correct or incorrect and sets each individual's status to active or inactive based upon said hypothesis. The diffusion phase then governs communication between active and inactive agents. If an agent is inactive, it will select another agent arbitrarily, else the original agent will adopt it's hypothesis as their own. However, if this chosen agent is inactive then the agent will randomize their hypothesis once more to continue exploring the search space. Many variations of the algorithm exist to adapt to different problem spaces by altering properties such as the recruitment strategy and context mechanism used.

Each agent is initialised with a random x and y coordinate in the image to serve as it's hypothesis. Upon the test phase, the agent will check it's x "neighbourhood". The neighbourhood is the number of pixels that each agent will check horizontally to the left and right of itself. The agent will check if the pixel n steps away from it on the left matches the pixel n steps away from it on the right. Each agent performs a count of how many of these corresponding pixels are equal in colour value and if this number is equal to the neighbourhood size, then the agent's status will update to active. Once active, the agent will draw a vertical line at it's successful x coordinate to highlight the symmetry axis.

The size of the neighbourhood makes a huge impact on symmetrical solutions found. When using a much smaller neighbourhood size, agents find many more local examples of symmetry. Although in that small area the agents may find an instance of symmetry, it does not correspond to our perception of symmetry when we view the image as a whole.

SDS Algorithm –

```

Agents Initialisation ()
While ( Not meeting the stopping criteria )
    Test the Hypothesis()
    Implement Diffusion Technique()
End

```

Several diverse problems has been solved using the SDS methods. It has been extensively used in Visual applications such as object recognition, real time tracking of lips in video films. Focused Stochastic Diffusion Network is a new method used in identifying the candidate positions in the wheel chair movement in a complex environment. This is done by tracking the cells used in this problem.

Cuckoo Search Algorithm [9]

Cuckoo search (CS) was introduced in 2009, we were able to perceive extensive uses across several real world applications as well as optimization problem resolution. The major inspiration is from the behavior of the Cuckoo birds (brooding nature). This was then proposed for the continuous improvement problems and has been the foundation for various numerical problems. The main areas of implementation are scheduling, optimization, planning and forecasting, image processing etc.

Yang and Deb [11] used three ideal rules to implement this in the optimization tool

1. The cuckoo bird lays eggs in a random nest and then dumps it. This is looked upon as a set of solution coordinates
2. Among this only some of them will include the best nest and will hatch them to form the next gen birdies. This would be the best solutions
3. There are chances that the host bird identifies the alien egg. It will then decide to abandon the egg alone or the whole nest. If it discards the whole nest, then it will end up identifying the new location.

Cuckoo Search (CS) algorithm is inspired from the above three nature of the Cuckoo bird and the algorithm is derived as follows

```

begin
Identify the Objective Function
Create the initial set of host nests (n)
While (Stop Criteria is met OR Max Generation Not reached)
    Get a random cuckoo (i)
    Assess its quality
    Choose a nest from the above set of nests(j)
    If(Fitness(i) > Fitness(j))
        Mark the new solution as j;
End

```

Given below is the process of identifying the new solution

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus Levy(\beta)$$

The commonly used case is to assign $\alpha = 1$.

Table 1: Variants of Cuckoo Search.

Name	Author
Discrete binary CS	Gherboudj et al.
Discrete CS	Jati and Manurung
Discrete CS for TSP	Ouaarab et al.
Neural-based CS	Khan and Sahai
Quantum inspired CS	Layeb
Emotional chaotic cuckoo	Lin et al.
Cuckoo Search Based LM	Nawi et al.
Parallelized CS	Subotic et al.
Modified CS	Tuba et al.
Modified CS	Walton et al.
Modified adaptive CS	Zhang et al.
Multiobjective CS	Yang and Deb
A Novel Complex Valued	Zhou and Zheng
CS based on Gauss distribution	Zheng and Zhou
CS based on Gaussian disturbance	Wang et al.

1E. Particle swarm optimization : It is a heuristic/empirical method put forward by Doctor Kennedy and Eberhart in 1995(Kennedy J,Eberhart R,;EberhartR,Kennedy J) and mainly for global optimization.

The basis of this algorithm is derived from the flocking behavior of fish and birds. The solution derived is equivalent to the birds swarm. The information of the food source is which is good is actually an optimal solution. The birds moving in search of this optimal solution is like the development phase of the solution. The final food destination is the most optimal solution.

The basic logic of this algorithm is to assume that the swarm contains “n” particles and the position of each of this is the possible solution in the search space.

The particles change its state/position as per the below three rules:

1. To keep up the torpor/inertia
2. Move towards the optimal solution
3. Reach the destination towards the most optimal solution.

Every swarm particle will have a position and speed assigned to it. The most optimal speed and position will also be marked as per the surrounding particles. The below equation represent the position of the particle along with speed[8]

$$v_{id}^{k+1} = v_{id}^k + c_1 r_1^k (pbest_{id}^k - x_{id}^k) + c_2 r_2^k (gbest^k - x_{id}^k)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}$$

In this equality,

v_i and x_i represents the speed of the particle “i”

d represent the dimension of the position;

p_{best} stands for the optimal solution

G_{best}^k is the d -dimension quantity of the swarm at its most optimistic position.

We limit the speed of the particle between $-v_{dmax}$, and v_{dmax} , so that it does not go far away from the search space.

V_{dmax} value will give us the picture of whether the solution is too far or near to the best solution. If its too high, then it means its far from the solution.

if the value is too less, then the solution will represent local optimistic solution;

c_1 and c_2 represent the fast moving particle, adjusting the length during the flight. This figure determines the position of the target. The smaller value depicts that its far away and if the value is high, then it either is reaching the target or flown past it.

r_1 and r_2 represent random number, between 0-1

Given below are the advantages of the PSO algorithm:

1. This algorithm is based on the Artificial Intelligence and is used in many scientific researches.
2. This is based of the speed at which the particle moves and the speed of researching should also be fast. The most positive/optimistic particle will be accountable to communicate the information to the other particles .
3. Compared to other algorithms this is simpler in terms of calculations and fetches better results. It also has better optimization results

Listed below are the disadvantages of the PSO algorithm:

1. There is a risk of partial optimization which creates lower resolution results.
2. Another difficulty is that it cannot handle the scattering complications (Chen Yonggang, Yang Fengjie, Sun Jigui ,2006,(In Chinese)).
3. This algorithm is not beneficial in the moving guidelines of the particle in “energy field” as they involve the non-cordiate systems.

Particle swam optimization is easier and more efficient compared to the other algorithms and is a relatively recent heuristic optimization technique in the Swarm Optimization methods. It requires lesser input parameters , hence became more popular.

2. Conclusion

We have compared the different Swarm Optimization algorithms and the best suited for the search based applications varies based on the type of parameters available. Varied results have been published for PSO and has been established to be simple and easily achievable.

References

- [1] Swarm Intelligence - Introduction. Author- Thiemo Krink, EVALife Group, Dept. of Computer Science, University of Aarhus.
- [2] Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique by Marco Dorigo, Mauro Birattari, and Thomas Stützle Université Libre de Bruxelles, BELGIUM - Nov 2006.
- [3] BEE COLONY OPTIMIZATION PART I: THE ALGORITHM OVERVIEW - Tatjana DAVIDOVIĆ, Dušan TEODOROVIĆ and Milica ŠELMIĆ.
- [4] A new technique based on Artificial Bee Colony Algorithm for optimal sizing of stand-alone photovoltaic system - Ahmed F. Mohamed *, Mahdi M. Elarini, Ahmed M. Othman.
- [5] On the application of artificial bee colony (ABC) algorithm for optimization of well placements in fractured reservoirs; efficiency comparison with the particle swarm optimization (PSO) methodology - Behzad Nozohour-leilabady a, Babak Fazelabdolabadi.
- [6] <https://hackernoon.com/stochastic-diffusion-search-c3f86d9f74a5>
- [7] Stochastic Diffusion Search Review- Mohammad Majid al-Rifaie, Mark, J. Bishop, Department of Computing, Goldsmiths College, University of London.
- [8] Analysis of Particle Swarm Optimization Algorithm - Qinghai Bai, College of Computer Science and Technology, Inner Mongolia University for Nationalities.
- [9] Cuckoo Search: A Brief Literature Review - Iztok Fister Jr., Xin-She Yang, Dusan Fister, Iztok Fister, Iztok Fister Jr. Faculty of Electrical Engineering and Computer Science.
- [10] A Novel Cuckoo Search Optimization Algorithm Base on Gauss Distribution - Hongqing ZHENG1, Yongquan ZHOU.
- [11] Passino K. M., Biomimicry of Bacterial Foraging for Distributed Optimization, University Press, Princeton, New Jersey (2001).

