Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 01, January-2016

Nature Inspired Meta-heuristics: A Survey

Nidhi Saini
Student,
Computer Science & Engineering
DAV Institute of Engineering and Technology
Jalandhar, India

Abstract: **Nature provides a major inspiration for solving many complex problems in computer science. One of such problem is Optimization. The field of optimization contains many algorithms inspired from nature to find the best and optimal solutions. Many novel techniques had been developed for the purpose of optimization. This paper gives an overview of various nature-inspired meta-heuristics and algorithms which are being used currently for solving the global optimization problems. The algorithms will be discussed on the basis of their concept, working, accuracy and speed. The applications of the algorithms are also presented in detail.**

Keywords: Algorithms; Global optimization; Meta-heuristics; Nature-inspired; Optimization

I. Introduction

Optimization is always a need of computer science. In computer science, there are many problems which have a number of possible solutions but all of them are not capable to get accepted. So we needed some techniques which will help to find the best solution among them. So the concept of optimization was introduced for this purpose. Optimization means to select or find the best solution from a set of candidate solutions. There are large number of problems related to optimization in computer science which needs to be solved as efficiently as possible. Some of these problems are Travelling Salesman Problem (TSP) [10], knapsack problem [11], vehicle routing and n-Queen problem [12]. For this purpose many meta-heuristics are developed. Meta-heuristics are the higher-level procedures which are used to find, generate or select the best solution from a set of candidate solutions. Although Polynomial-time algorithms are available to solve optimization problems, but they are not capable enough to solve all the problems. Nature contains the solution for every problem which may arise, same is with the problem of optimization. Nature has provided many ways to solve the optimization problems. Various optimization techniques developed till now are inspired from the nature and day-to-day life. In global optimization, we need to find the solution which will have minimum value for the objective function of the problem. Such a solution is called global minimum. Meta-heuristics consists of various iterations through which various candidate solutions of a problem are tested. Based on that, the best solution is decided.

Evolutionary algorithms and swarm-intelligence algorithms are such nature-inspired algorithms. These optimization techniques have been applied into various fields of computer science to solve complex problems like networking, code optimization and many more. We are going to classify these algorithms based on their inspirations. These algorithms tend to solve the problems of computer science efficiently. This paper will discuss most efficient Evolutionary Algorithms, Swarm-intelligence algorithms and single solution based algorithms. Each algorithm is inspired from a different behaviour of nature and solve the problems using that concepts.

Evolution means generation and Evolutionary algorithms use the concept of generations to find the best solution from a set of candidate solutions. In these algorithms, generations refer to the iterations. These are one of the earliest algorithms developed to solve the optimization problems. These are well-established and efficient algorithms.

Swarm-intelligence algorithms are inspired from the behaviour of various living organisms like birds, ants or bees. These are simulation-based optimization techniques which makes use of an artificial environment to understand the working of any proposed algorithm. Like, in Ant colony optimization algorithm artificial ant colonies are created to implement the algorithm. Although artificial environment is not as efficient as real environment, still it is highly beneficial for execution of such algorithms.

II. OPTIMIZATION ALGORITHMS

1. *Genetic Algorithms (GA)*

Genetic Algorithms are a type of Evolutionary algorithms proposed by J.H. Holland in 1975[1]. From its name only, it is clear that GA is inspired from the evolutions in the human life. These are based on the concept of generations which are simulated as iterations in this algorithm. Genetic Algorithm can be explained by taking example of human beings. It makes use of the concept of natural selection to solve the problems in computer science. Firstly, GA initialize the population of solution. Among these number of solutions, each candidate solution is considered as an individual who goes through process of evolution. An individual is represented as a binary string of finite length. At the time of its invention, Genetic Algorithms attracted every computer science professional as it was most efficient stochastic algorithm proposed by that time which consisted of numerous interesting properties. The main objective of GA was the optimization of non-linear systems which have a large number of variables. In genetic algorithms, the user have access to define own rules for the processes of selection, crossover and mutation. Genetic algorithms can be implemented in the following three steps:]

a. Selection: Each candidate solution consists of a fitness score which tells about the ability of that solution to compete with other solutions. It is determined by either by some objective function or the subjective judgement.

b. Crossover: Crossover operator is responsible for mating of the selected candidate solutions. In this, a crossover site is chosen randomly. Then, the bit strings of the selected candidate solutions are exchanged upto the crossover site. This will result into two new candidate solutions which are better then their parent solutions. These new solutions are then included in the next generation of population.

c. Mutation: Due to premature convergence in crossover operation, some new candidate solutions may have flipped bits. It maintains genetic diversity among the candidate solutions in each generation.

The process followed in each iteration of Genetic Algorithm can be described diagrammatically as follows:

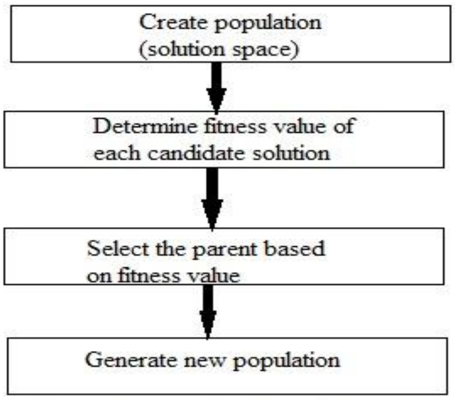

figure1: Steps in Genetic algorithm

2. Differential Evolution (DE)

Differential evolution is given by Storn and Price in 1995.[2] This technique became highly popular due to its speed and efficiency which was far better than existing algorithms in that time. DE was a superior algorithm in all domains compared to other algorithms developed before it. The main aim of proposing Differential Evolution algorithms was to optimize the continuous functions. DE is another evolution based algorithm used for optimization developed to optimize real parameters and functions. This strategy is quite flexible to solve most of the optimization problems with simplicity. DE have very small code for execution and a few control parameters which makes it easier to use as compared to other Evolutionary Algorithms. DE can find appropriate solutions for non-continuous and non-differentiable problems too, which is extremely difficult for other algorithms for optimization. DE has the same components as GA except mutation. Differential Evolution was basically developed for minimization problems, but it can also be used to solve optimization problems by transforming them into minimization task. The objective functions are responsible for such transformation of problems.

Mutation in GA: In DE is arithmetic combination of candidate solutions, whereas mutation in GA is outcome of small regular changes in the genes of individuals(candidate solutions). In the process of evolution, mutation supports exploration in the beginning stage and then followed by the exploitation.

DE make use of NP parameter vectors as a population for each generation.[2] It is more accurate, fast and easily implementable algorithm as compared to the other algorithms of the same class. DE can be executed on the parallel machines to achieve high speed which makes it useful for solving real-world problems. The only thing which can effect the performance of DE is noise. The most difficult task in DE strategy is to define the control parameters. To solve this problem many self-adapting algorithms have been proposed which are capable enough to allocate control parameters and give good results. The value of control parameters can be assigned in the following two ways:

a) *Tuning:* The values of the control parameters are decided beforehand, and then the algorithm is run. These control parameters will remain same throughout the execution and the algorithm have to tune up with these values.

b) *Control:* The values of control parameters are changed while running the algorithm. The algorithm have ability to decide the appropriate values.

3. Particle Swarm Optimization (PSO)

This optimization technique was developed by Russel Elberhart and James Kennedy in 1995 for optimizing the continuous non-linear functions[4]. PSO is an algorithm of swarm-intelligence based on the behaviour of flock of birds. This is an easy approach to solve the global optimization problems. If a flock of birds is flying over an area having some source of food. The noise of bird closed to the source will be louder as compared to the bird which is away from the food source. If some bird reach the closest place near the bird, it will chirp the loudest, so the other birds can hear him and reach this new close point. Through this way all the birds in a flock will try to adapt to the place which is closer to food source as compared to their own place. It can be implemented by using both global search and local search. In this each candidate solution in the search space is considered as a particle having a random velocity with which it flies or travel through the search space of number of candidate solutions.

Local search: In local search, a particle tries to attain the best possible solution by obtaining the appropriate coordinates in the space. The value of these coordinates is represented by *pbest*. This value will be obtained by the particle on its own without the help of its neighbours.

Global Search: In global search, it aims to find the overall best solution called *gbest* by obtaining best location for the particle in search space. This is fulfilled by communicating with the neighbouring particles present in the search space and then tends to find the best location for itself. If any neighbouring particle has the better location then the current location of the particle, the it moves to that new better location. The algorithm keeps track of the following three global variables:

- Target position or the position of the food source
- Global best (gBest) value indicating which particle's data is currently closest to the food source. It is achieved using global search.
- Stopping value which tells when the algorithm should stop if the food source is not found yet.

4. *Ant Colony Optimization (ACO)*

Ant Colony Optimization is a stochastic optimization algorithm of swarm-intelligence based on the behaviour of ants searching for food. ACO was proposed by Marco Dorigo and Di Caro in 1999[5] to solve the problems of combinatorial optimization. This algorithm provides approximate solutions to the defined problem, these solutions are based on the behaviour of the ants.

Combinatorial Optimization (CO): Combinatorial Optimization is an approach used to find the best solution from a discrete set of feasible solutions. CO is a branch of optimization and related to the subjects of algorithm theory, operations research and computational complexity theory. CO is used most widely in the research of software engineering, machine languages, mathematics and artificial intelligence.
The ants try to find the shortest path to the food. Ants often go out from their nest in search of food. Ants make use of a chemical called pheromone to communicate with each other. Whenever an ant finds a source of food, it carries the food to the nest. While coming back to the nest with food, ants release a chemical called pheromone trail on the land. The amount of pheromone released by an ant depends on the quality and the quantity of the food[5]. Other ants have ability to smell this chemical and reach the food source. The release of the chemical helps also help the ants to discover the shortest path from their nest to the food source. With the passage of time, the pheromone also starts evaporating. Artificial ant colonies are created to understand the working of ACO algorithm in detail. This model makes use of the pheromone model to construct the problem space. This problem space consists of pheromone values. The pheromone values are the candidate solutions of the given problem to be solved. The steps to achieve best solution using Ant Colony. Optimization technique is as follows:

1. Construct the solution space consists of candidate solutions using the pheromone model
2. These candidate solutions are used to update the pheromone values in a way that is deemed to bias future sampling toward high quality solutions[5].

Self-reinforcing process: This process is followed by the ants. In this process, a path is created by the ants using high pheromone concentration is more followed by the ants as compared to the path having low pheromone concentration. This behaviour of the ants lead to a self-reinforcing process.

5. *Artificial Bee Colony (ABC)*

Artificial Bee Colony is another swarm-intelligence algorithm proposed by D.Karaboga and B.Basturk in 2007 [6]. ABC is a population-based algorithm which is inspired from the concept of the foraging behaviour of the swarm of honeybees. There are many algorithms proposed with inspiration from the behaviour of bees. Some of these algorithms are based on foraging behaviour while some are based on the mating behaviour of the bees. There are three types of honeybees in a colony:

(a) *Employed bees:* Each employed bee is assigned a food source. The employed bee goes to its food source and check it. After checking, they come back to hive and starts dancing over there. The number of employed honeybees will be the same as the number of sources of food around their hive.
(b) *Onlooker bees:* The onlooker bees have to watch the dance of the employed bees and then select the food source depending on their dance.
(c) *Scout bees:* The scout bees are those whose food source are abandoned and they start looking for the new food sources around their hive. After finding an appropriate food source, the scout bees become the employed bees.

According to this algorithm an artificial bee colony is generated. In which all the employed bees go to their food source in their mind and find the neighbouring source. The process can be simulated for solving an optimization problem as follows:

i. Initialize the population of candidate solutions
ii. This population goes through the various iterations of the employed bees, scouts and onlookers. The number of employed bees will be the same as number of solutions.
iii. An employed bee make changes to its source location and tries to find a new location which will give more amount of nectar as compared to the previous source location.
iv. The location of this new location is remembered by the bee and the old one is forgotten. This means the new selected solution gives better results as compared to the previous solution.
v. All the employed bees perform the same functions as mentioned in step iii and step iv. In this way, a number of competitive solutions are found which give good results. Then all the bees share their locations with the onlookers by dancing in the hive.
vi. The onlookers are responsible to watch the dance and decide a food location based on the dances. The food source with the highest quality is get selected by the onlookers and nectar is extracted from them. This method is used to find the best solution among the previously selected solutions.
vii. All the abandoned food resources from which nectar has been taken out are abolished and new food sources are found to replace them.

6. *Simulated Annealing (SA)*

Simulated Annealing is a single solution based meta-heuristic proposed by Scott Kirkpatrick and Mario Vecchi in 1983 [7]. SA is a stochastic and probabilistic approach developed to solve the problems of optimization. It was highly useful and thus became popular due to its numerous properties. It was successfully applied to solve the various optimization problems like TSP and graph partitioning. SA has emerged from the concept of matter physics. The process of heating and then cooling slowly which results in stronger structure is called annealing. Simulated Annealing algorithm is used to simulate the behaviour of any material on heating and then cooled down to change the structure of that material. This makes the microstructure of that material stronger than before. SA can be used to make the improvement in several iterations. Each iteration will provide with better results compared to the previous iteration. It considers a set of candidate solution from which the best solution is to be determined. Firstly, the melting point of the material is determined and the algorithm starts by selecting any random candidate solution. This candidate solution will go through various iterations. In an iteration, a neighbouring candidate solution is chosen and then compared to the current candidate solution using an objective function. If the neighbouring candidate solution is better then it replaces the current candidate solution. If not, the current candidate solution will be accepted with a probability which is exponential value of the ratio of energy difference between two candidate solutions and temperature at the moment. On each level of temperature, various neighbouring candidate solutions are checked before reaching the equilibrium point. After reaching equilibrium point, the material starts cooling down and temperature decreases with a schedule called geometric scheduling [7]. The temperature is decreased constant for the purpose of cooling whereas htere is no fixed criteria while heating the temperature. As the temperature starts decreasing lesser number of candidate solutions get accepted. On reaching a certain point, the algorithm stops based on the criteria defined for the particular problem. The output of this algorithm after terminating will be the best solution found from the given set of candidate solutions. In some cases, the objective functions are not smooth enough to achieve the low-lying optimum at low temperatures. So, it will be more beneficial to search for the best solutions at moderate temperatures. SA has been proved to be one of the most effective and efficient nature-inspired optimization algorithm.

7. *Intelligent Water Drops Algorithm (IWD)*

Intelligent Water Drop is proposed by Hamed Shah Hosseini in 2007 [8]. IWD is a nature-inspired algorithm for global optimization based on the behaviour of water drops in a river and the soil present in the river bed. The goal of each river is to reach the sea for which the river have to travel a long distance. For this purpose, the river find and follows the optimum path which is decided by its surroundings, that is, water drops in the river and soil around the river. IWD algorithm is used to determine the shortest path from the source point to the destination point. In some cases the destination is not known, then IWD tends to find the most optimal solution for the given problem. The nature of water drops is flawless. The drops of water will always flow from higher place to the lower place on land due to the force of gravitation. Due to this gravity, the water drops always try to follow the shortest path but there are many barriers which lie in the path of rivers which tend to change the natural flow of a river and obstructs the flow of water drops. These barriers create a different path for river, which can be considered as the actual path. This actual path will be different and full of twists and turns which we often see in the flow of a river. The rivers change their path based on their surroundings. In spite of water drops, the flow of river also depends on the soil present along the river bed. The water drops in the river also carry the soil with them. The soil also flows from one place to another with the flow of water. At some places, the speed of water will be more and on some places it will be less. The fast moving water will carry more soil in it. The fast moving water will have more tendency to take away soil with it. Due to this reason, that place will have lesser quantity of soil which will result in more place for water. The places where flow of water is less contains more soil. The soil is took off from the places of fast flow of water and dropped in the places where the flow of water is less.

For solving the problems with this approach, artificial water drops are generated which have almost same properties as that of natural water drops. Although there are numerous ways which may effect the flow of a river stream. But in IWD only two parameters are considered, that are, the speed of water drop(velocity) and the soil carried by water drops (soil). As the intelligent water drop moves on, the values of both parameters will change based on the surroundings at that moment. So IWD will move on with discrete finite - length steps [8]. The drop will move from its current location to next better location. At this new location the speed of water drop will be increased non-linearly proportional to the quantity of soil between the previous and new locations. More soil will be added to the water drops on the way. The quantity of this added soil depends inversely on the time taken by the water to travel the distance between two locations. Thus, the time taken bu IWD is proportional to the velocity whereas inversely proportional to the distance to be travelled to reach the next location [8]. The soil present in the path is responsible for reducing the speed of flowing water drops. So the paths with less quantity of soil are considered better in this algorithm and chosen more probably. Initially IWD was developed to solve the Travelling Salesman Problem, but later it was found capable enough to solve other engineering problems.

8. *Simulated Raindrop Algorithm (SRD)*

Simulated Raindrop Algorithm is proposed by Amin Ibrahim, Shahryar Rahnamayan and M.V. Martin in 2014 [9]. SRD is a single solution based meta-heuristic inspired from the concept of raindrops falling on the land. Like other meta-heuristics, SRD algorithm also starts by selecting a random candidate solution from a defined set of candidate solutions. In successive iterations, the steps are taken to get the best solution from the set of candidate solutions. When a raindrop falls on the land it tries to reach the lowest point in that landscape. This same procedure is used in this algorithm to achieve the best solution for a problem. In SRD algorithm, the terrain is represented by the objective function, the flow of raindrop from a higher point in terrain to the lower point is represented by the local search among the raindrop splashes and the lowest point in that terrain is represented as the optimal solution [9]. The steps of reaching an optimal solution through SRD algorithm are as follows:

i. An initial candidate solution is selected randomly based on the problem dimension

ii. The number of splashes generated by this raindrop are calculated

iii. In every iteration, number of splashes are generated for each raindrop. These splashes are used to find the best solution. The splashes of the raindrop are generated as follows:

(a) The splashes generated are of varying size. The displacement of a splash plays the vital role in the efficiency of this algorithm. Splash displacement[9] was used to determine the improving moves, which should be taken. In initial stages, the displacement is more and reduces in the further stages.

(b) After selecting the improving splash, the splash replacement takes place.

First, if any of the splash generated by the raindrop in current iteration proves to be the best solution then the location of that splash is taken by the raindrop in next iteration. In next iteration, the number of splashes are halved to that of previous iterations.

Second, if there is no improved solution provided by any of the splashes then the raindrop remains at same location in next iteration but new splashes are generated from it. From these splashes the best solution is determined. This process is repeated until the splash with best solution is found. By going through various iterations using the above steps, the best solution is achieved. This algorithm has provided the better results as compared to the well-known S-metaheuristic Simulated Annealing. This algorithm gives the better and more improved results with the increase in problem dimension. This work is in its initial stage but still it has proved to be highly beneficial to solve the problems of computer science.

III. Comparative Study

Here are the various differences which can be made in the above mentioned algorithms:

Table 1: Comparative Study of optimization algorithms

Sr. No.	Algorithm	Category	Inspiration	Improved approach	Applications
1	Genetic Algorithm (GE)	Evolutionary	Generations of human life	Multi-objective Genetic Algorithm	Robot path planning, artificial creativity, bioinformatics, Automated design, CAD, File allocation, Scheduling applications, Mobile communication infrastructure, mutation testing, TSP, vehicle routing, wireless sensor networks and many more.
2	Differential Evolution (DE)	Evolutionary	Generations of human life	-	Data mining, robotics, portfolio optimization, pattern recognition, cancer diagnosis, Design of image exploring agent, power transformer fault classification automatic synthesis of analog electrical circuits, automated synthesis of analogue electrical circuits symbolic regression
3	Particle Swarm Optimization (PSO)	Swarm-intelligence	Flock of birds	PS2O	Optimization of hydrocarbon field development planning, Power system optimization problems, edge detection, Anomaly detection, color image segmentation, sequential ordering problem, constrained portfolio optimization problem, selective particle regeneration for data clustering, Extracting rules from fuzzy neural network, Signature verification & many more
4	Arti Colony Optimization (ACO)	Swarm-intelligence	Collective Behaviour of ants searching for food	Ant Colony System	Travelling Salesman Problem, open shop scheduling problem, sequential ordering problem, scheduling problems, vehicle routing problems, assignment problems & many more
5	Artificial Bee Colony (ABC)	Swarm-intelligence	Foraging Behaviour of ants and bees	Fitness scaled chaotic Artificial Bee Colony (FSABC)	Image Segmentation, Structural optimization, Pathological Barin Detection, nanoelectronic based phase-locked loop, pattern classification, reliability redundancy allocation problems, clustering, resource-constrained project scheduling problem
6	Simulated Annealing (SA)	Single Solution based	Matter Physics	Adaptive Simulated Annealing	Min-cut partitioning, Travelling Salesman Problem, local search, vehicle routing, n-Queen problem and many more
7	Intelligent Water Drop Algorithm (IWD)	Swarm-intelligence	Flow of river	IWD - CO	Travelling Salesman Problem, vehicle routing, robot path planning, graph colouring, MKP, MANET, continuous optimization, Maximum Clique problem, Optimization of manufacturing process models , code coverage, optimization of route protocols
8	Simulated Raindrop Algorithm (SRD)	Single Solution based	Falling of Raindrops on land	-	Mitigate DDoS attacks in Cloud Computing

IV. CONCLUSION

This paper has discussed some of the most important and well-known optimization algorithms proposed till now. The efficiency of these meta-heuristics is well-explored as they have been used to solve the numerous problems of computer science as well as optimization. All of the above discussed algorithms are inspired from the nature which proves that nature has the ability to provide the solutions to any of the problems which occurs in the life even if those problems are completely virtual like optimization. All these algorithms have enough capability to solve the problems related to optimization in future also as they have been developed in general and not for any particular problem. These algorithms are can be categorised based on their inspirations. Almost all the algorithms have the wide range of application areas. In the past few years, the optimization strategies have been explored widely to get maximum benefits out of them. Many new algorithms are also being created nowadays to get most optimal and best solutions from the problems related to optimization. Still these algorithms needed more exploration as they can prove to be more efficient and effective. The applicability of such meta-heuristics can be more widened. This paper has also discussed the comparison among the algorithms. The comparison has been made on the basis of their category, inspiration, improvement and applications.

The research work to be done in the field of optimization have no limitations and can be increased as much as possible.

REFERENCES

[1] J. H. Holland, Genetic algorithms and the optimal allocation of trials, SIAM J. Comput. 2 (2) (1973) 88–105

[2] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.

[3] J. Kennedy and R. C. Eberhart,“Particle swarm optimization,” In Proceedings of the 1995 IEEE International Conference on Neural Network, Vol. 4. pp. 1942-1948. IEEE Press, 1995.

[4] M. Dorigo, V. Maniezzo, & A. Colorni (1996). Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B, 26, 29–41.

[5] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization 39 (2007), 459–471

[6] S. Kirkpatrick, & M. P. Vecchi, “Optimization by simmulated annealing,”. Science, 220 (4598), pp. 671-680, 1983.

[7] H. Shah-Hosseini,, “Problem solving by intelligent water drops,” Evolutionary Computation, 2007. CEC 2007. IEEE Congress on , vol., no., pp.3226-3231, 25-28 Sept. 2007.

[8] A. Ibrahim, S. Rahnamayan and M.V. Martin, “Simulated Raindrop Algorithm for Global Optimization”, In proceedings of the 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering, 1-8

[9] S. Lin, “Computer Solutions of the Travelling Salesman Problem”, Bell Syst. Journal, vol. 44, 1965, pp. 2245-2269H. Shah-Hosseini, “Intelligent water drops algorithm: a new optimization method for solving the multiple knapsack problem,” Int. Journal of Intelligent Computing and Cybernetics, Vol. 1, No. 2, pp. 193-212, 2008a.

[10] Shah-Hosseini, “The Intelligent Water Drops algorithm: A nature inspired swarm-based optimization algorithm,” Int. J. Bio-Inspired Computation, Vol. 1, Nos.1/2, pp. 71–79, 2008b.