APPROACHES OF META HEURISTICS OPTIMIZATION TECHNIQUES

By

J. SHEIK MOHAMED * S. RAMAKRISHNA ** P. CHITTIBABU ***

* Research Scholar, Department of Computer Science, SVUCCMCS, S. V. University, Tirupathi, Andhra Pradesh, India.
** Professor, Department of Computer Science, SVUCCMCS, S. V. University, Tirupathi, Andhra Pradesh, India.
*** Principal, APGCCS, Rajampet, Kadapa Dist, Andhra Pradesh, India.

ABSTRACT

Optimization is ubiquitous and spontaneous process that forms an integral part of day-to-day life. In the most basic sense, it can be defined as an art of selecting the best alternative among a given set of options. Optimization plays an important role in Engineering designs, Agricultural sciences, Manufacturing systems, Economics, Physical sciences, Pattern recognition[1] and other such related fields. The objective of optimization is to seek values for a set of parameters that maximize or minimize the objective functions subject to certain constraints. A choice of values for the set of parameters that satisfy all the constraints is called a feasible solution. Feasible solutions with objective function value(s) are as good as the values of any other feasible solutions, that are called as optimal solutions. In order to use optimization successfully, they must first determine an objective through which they can measure the performance of the system under study. That objective could be time, cost, weight, potential energy or any combination of quantities that can be expressed by a single variable. The objective relies on certain characteristics of the system, called variable or unknowns. The optimization algorithms come from different areas and are inspired by different techniques. But they are sharing some common characteristics. They are iteratives that are begun with an initial guess of the optimal values of the variables and then generate a sequence of improved estimates until they converge to a solution.

Keywords: Optimization, Metaheuristics, Genetic Algorithm, Ant Colony, Simulated Annealing.

INTRODUCTION

As long as humankind exist, they strive for perfection in many areas. They want to reach a maximum degree of happiness with the least amount of effort. In their economy, profit and sales must be maximized and costs should be low as possible. Therefore, optimization is one of the oldest of sciences which even extends into daily life[2]. If something is important, general, and abstract enough, there is always a mathematical discipline dealing with it. Global optimization is the branch of Applied Mathematics and Numerical Analysis that focuses well on, optimization. The goal of global optimizations to find the best possible elements $x^\star$ from a set X, according to a set of criteria $F = \{f1, f2, .., fn\}$. These criteria are expressed as mathematical functions, that are called objective functions[2].

1. Preliminary ideas

1.1 Heuristic

A heuristic is a part of an optimization algorithm that uses the information currently gathered by the algorithm to help to decide which solution candidate should be tested next or how the next individual can be produced. Heuristics are usually problem class dependent[4].

1.2 Metaheuristic

A metaheuristic is a method for solving very general classes of problems. It combines objective functions or heuristics in an abstract and hopefully in an efficient way, usually without utilizing deeper insight into their structure ,i.e., by treating them as black-box-procedures[4].

1.3 Online Optimization

Online optimization problems are tasks that need to be solved quickly in a time span between ten milliseconds to a few minutes. In order to find a solution in this short time, optimality is normally traded in for speed gains.

Examples for online optimization are robot localization, load balancing, services composition for business processes or updating a factory's machine job schedule after new orders come in. From the examples, it becomes clear that online optimization tasks are often carried out repetitively–new orders will, for instance, continuously arrive at a production facility and need to be scheduled to machines in a way that minimizes the waiting time of all jobs.

1.4 Offline Optimization

In offline optimization problems, time is not so important and a user is willing to wait may be even days if she can get an optimal or close-to-optimal result.

Such problems regard, for example, design optimization, data mining, or creating long term schedules for transportation crews. These optimization processes will usually be carried out only once in a long time. Before doing anything else, one must be sure about to which of these two classes the problem to be solved belongs.

1.5 Single Objective Functions

In the case of optimizing a single criterion f, an optimum is either its maximum or minimum, depending on what they are looking for. If they own a manufacturing plant and have to assign incoming orders to machines, they will do this in a way that minimizes the time needed to complete them. On the other hand, they will arrange the purchase of raw material, the employment of staff and the placing of commercials in a way that maximizes our profit. In Global optimization, it is a convention that optimization problems are most often defined as minimizations and if a criterion f is subject to maximization, they simply minimize its negation$(-f)$[9]. Figure 1 shows the global and local optima of a two-dimensional function.

Local Maximum: A local maximum $\hat{}\ xl \in X$ of one (objective) function

$f : X \rightarrow R$ is an input element with $f(\hat{}\ xl) \geq f(x)$ for all x neighboring $\hat{}\ xl$. If $X \subseteq Rn$, they can write:

$$\forall \hat{}\ xl\ \exists \varepsilon > 0 : f(\hat{}\ xl) \geq f(x)\ \forall x \in X, |x - \hat{}\ xl| < \varepsilon \qquad (1)$$

Local Minimum: A local minimum $\check{}\ xl \in X$ of one (objective) function

$f : X \rightarrow R$ is an input element with $f(\check{}\ xl) \leq f(x)$ for all x neighboring $\check{}\ xl$. If $X \subseteq R$, they can write:

$$\forall \check{}\ xl\ \exists \varepsilon > 0 : f(\check{}\ xl) \leq f(x)\ \forall x \in X, |x - \check{}\ xl| < \varepsilon \qquad (2)$$

Local Optimum: A local optimum $x\ l\star \in X$ of one (objective) function

$f : X \rightarrow R$ is either a local maximum or a local minimum.

Global Maximum: A global maximum $\hat{}x \in x$ of one (objective) function

$f : X \rightarrow R$ is an input element with $f(\hat{}x) \geq f(x)\ \forall x \in X$.

Global Minimum: A global minimum $\check{}x \in X$ of one (objective) function

$f : X \rightarrow R$ is an input element with $f(\check{}x) \leq f(x)\ \forall x \in X$.

1.6 Multiple Objective Functions

Global optimization techniques are not just used for finding the maxima or minima of single functions f. In many real-world design or decision making problems, they are rather applied to sets F consisting of $n = |F|$ objective functions fi, each representing one criterion to be optimized[9]

$$F = \{fi : X \rightarrow Yi : 0 < i \leq n, Yi \subseteq R\} \qquad (3)$$

Algorithms designed to optimize such sets of objective functions are usually named with the prefix multi objective,

Factory Example

Multi-objective optimization of ten means to compromise conflicting goals, and they can specify the following objectives that are all subjected to optimization:

- Minimize the time between an incoming order and the shipment o f the corresponding product.
- Maximize profit.
- Minimize costs for advertising, personal, raw materials etc..
- Maximize product quality.
- Minimize negative impact on environment.

The last two objectives seem to be contradicted clearly to the cost minimization. Between, the personal costs and the time needed for production and the product quality should also have some kind of relation. The exact mutual influences between objectives can apparently become complicated and are not always so obvious.

1.7 Evolutionary Algorithm

Evolutionary algorithms (EAs) are population based metaheuristic Optimization algorithms that use biology-

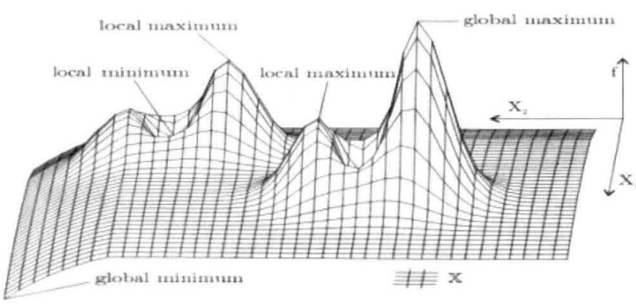

Figure 1. Global and local optima of a two-dimensional function

inspired mechanisms like mutation, crossover, natural selection and survival of the fittest in order to refine a set of solution candidates iteratively The advantage of evolutionary algorithms compared to other optimization methods is their "black box" character that makes only few assumptions about the underlying objective functions. Furthermore, the definition of objective functions usually require lesser insight to the structure of the problem space than the manual construction of an admissible heuristic. EAs therefore perform consistently well in many different problem categories.[10]

2. General Methods

Broadly speaking, optimization algorithms can be placed in two categories: the conventional or deterministic methods and the modern heuristics or stochastic methods.

Conventional methods adopt the deterministic approach. During the optimization process, any solutions found are assumed to be exact and the computation for the next set of solutions completely depends on the previous solutions found. That's why conventional methods are also known as deterministic optimization methods. In addition, these methods involve certain assumptions about the formulation of the objective functions and constraint functions. An important and widely used class of deterministic algorithms are the gradient-based methods capable of choosing their search directions according to the derivative information of the objective functions. Conventional methods include algorithms such as linear programming, non-linear programming, dynamic programming, Newton's method and others. Unfortunately, conventional optimization algorithms are not efficient at coping with the demanding real world problems without derivate information. In other words, selection of the initial points for the deterministic optimization methods has a decisive effect on their final results. However, a foresight of appropriate starting points is not always available in practice. One common strategy is to run the deterministic algorithms with random initialization numerous times and retain the best solution; however, this can be a time-consuming procedure. Therefore, in view of the practical utility of optimization problems, there is a need for efficient and robust computational algorithms, which can numerically solve on computers, the mathematical models of medium as well as large size optimization problem arising in different fields.

In the past few decades, several global optimization algorithms have been developed that are based on the nature inspired analogy. These are mostly population based metaheuristics which are also called general purpose algorithms because of their applicability to a wide range of problems. Some popular global optimization algorithms include Evolution Strategies (ES), Evolutionary Programming (EP), Genetic Algorithms (GA), Genetic Programming (GP), Simulated Annealing (SA) [7], Artificial Immune System (AIS), Tabu Search (TS) [5], Ant Colony Optimization (ACO) [3], Particle Swarm Optimization (PSO), Harmony Search (HS) algorithm, Bee Colony Optimization (BCO), Gravitational Search Algorithm (GSA), etc.

3. General Procedure of Optimization Algorithm

The algorithms have proved their mettle in solving complex and intricate optimization problems arising in various fields. Generally, global optimization algorithms differ from traditional search and optimization paradigms in three main ways by,

3.1 Utilizing a population of individuals (potential solutions) in the search domain

Most traditional optimization algorithms move from one point to another in the search domain using some deterministic rule. One of the drawbacks of this kind of approach is the likelihood of getting stuck at a local optimum. Evolutionary Computation (EC) paradigms, on the other hand, use a collection of points, called population. By inducing the evolutionary-like operators, they can generate a new population with the same number of members in each generation so that the probability of getting stuck is reduced.

3.2 Using direct fitness information instead of function derivatives or other related knowledge

EC paradigms do not require information that is auxiliary to the problem, such as function derivatives, but only the value of the fitness function, as well as the constraints functions for constrained problems. Thus fitness is a direct metric of the performance of the individual population member on the function being optimized.

3.3 Using probabilistic, rather than deterministic rules

These stochastic operators are applied to operations which lead to the search towards regions where individuals are likely to find better values of the fitness function. So, the reproduction is often carried out with a probability which is dependent on the individual's fitness value.[10]

A universal procedure can be described that can be implemented no matter what the type of algorithm is.

- Initializing the population in the search domain by seeding the population with random values.
- Evaluating the fitness for each individual of the population
- Generating a new population by reproducing selected individuals through evolutionary operations, such as crossover, mutation and so on.
- Looping to step 2 until stopping criteria are satisfied.

One can recognize two common aspects in the population-based heuristic algorithms exploration and exploitation. The exploration is the ability of expanding search space, where as the exploitation is the ability of finding the optima around a good solution. In premier iterations, a heuristic search algorithm explores the search space to find new solutions. To avoid trapping in a local optimum, the algorithm must use the exploration in the first little iteration. Hence, the exploration is an important issue in a population-based heuristic algorithm. By lapse of iterations, exploration fades out and exploitation fades in, so the algorithm tunes itself in semi-optimal points. To have a high performance search, an essential key is a suitable tradeoff between exploration and exploitation. However, all the population-based heuristic algorithms employ the exploration and exploitation aspects but they use different approaches and operators. In other words, all search algorithms have a common framework. From a different point of view, the members of a population-based search algorithm pass three steps in each iteration to realize the concepts of exploration and exploitation: self-adaptation, cooperation and competition. In the self-adaptation step, each member (agent) improves its performance. In the cooperation step, members collaborate with each other by transferring information. Finally, in the competition step, members compete to survive. These steps have usually stochastic forms, and could be realized in different ways. These steps, inspired from nature, are the principle ideas of the population based heuristic algorithms. These concepts guide an algorithm to find a global optimum.

4. Approaches of Diferent Optimization Techniques

4.1 Genetic Algorithms

Genetic algorithm is one of the most popular types of Evolutionary algorithms. To be more precise, it constitutes a computing model for simulating natural and genetic selection that is attached to the biological evolution described in Darwin's Theory which was first issued by Holland (1975a, b) [6]. In this computing model, a population of abstract representations(called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem could result in better solutions, which are traditionally represented in binary form as strings comprising of 0 and 1 s with fixed-length, but other kinds of encoding are also possible which include real-values and order chromosomes. The program then will assign the proper number of bits and the coding.

Being a member of the family of evolutionary computation, the first step of GA is population initialization which is usually done stochastically. The GA usually uses three simple operators called selection, recombination (usually called crossover) and mutation. Selection is the step of a genetic algorithm in which a certain number of individuals is chosen from the current population for later breeding (recombination or crossover), the choosing rate is normally proportional to the individual's fitness value. There are several general selection techniques. Tournament selection and fitness proportionality selection (also known as roulette-wheel selection) considers all given individuals. Other methods only choose those individuals with a fitness value greater than a given arbitrary constant. Crossover and mutation taken together is called reproduction. They are analogous to biological crossover and mutation respectively.

The most important operator in GA is crossover which refers to the recombination of genetic information during sexual reproduction. The child shares in common with its parents many characteristics. Therefore, in GAs, the offspring has an equal chance of receiving any given gene from either one of two parents because the parents' chromosomes are combined randomly. Till date, there are many crossover techniques for organisms which use different data structures to store themselves, such like One-point crossover, two-point crossover, Uniform Crossover as well as Half Uniform Crossover. The probabilities for crossovers vary according to the problem. Generally speaking, values between 60 and 80% are typical for one-point crossover as well as two-point crossover. Uniform crossovers work well with slightly lower probabilities on the other hand. The probability could also be altered during evolution. So a higher value might initially be attributed to the crossover probability. Then it is decreased linearly until the end of the run, ending with a value of one-half or two-thirds of the initial value. Additionally, for real-value cases, the crossover operation could be expressed as

$$y^1 = \lambda x^1 + (1-\lambda)x^2$$
$$y^2 = \lambda x^2 + (1-\lambda)x^1 \qquad (4)$$

where y^1 and y^2 are two descendants created by two given parents x^1 and x^2, and $\lambda = U[0, 1]$.

Mutation is the stochastic flipping of chromosome bits that occurs in each generation which is used to maintain genetic diversity from one generation of a population of chromosomes to the next. Mutation occurs step by step until the entire population has been covered. Again, the coding used is decisive. In the case of binary chromosomes, it simply flips the bit, while in real-valued chromosomes a noise parameter N[0, F] is added to the value at that position. F could be chosen in such way that it decreases in time, e.g., F = 1/(1 + %t), where t is the number of current iterations. The probability of mutation is normally kept as a low constant value for the entire run of the GA, such as 0.001. Despite these evolutionary operators, in some cases, a strategy called "elites" is used, where the best individual is directly copied to the next generation without undergoing any of the genetic operators. In one single interaction, new parents are selected for each child and the process continues until a proper size of individuals for the new population is reached. This process ultimately results in the population of the new generation differing from the current one. Generally, the average fitness of the population should be improved by this procedure since only the best organisms from the last generation are selected for breeding.[10]

The implementation of the genetic algorithm is described as follows

Step 1: Initialization

The algorithm starts with a set of solutions (represented by chromosomes) called population. A set of chromosomes is randomly generated. Each chromosome is composed of genes.

Step 2: Evaluation

For every chromosome, its fitness value is calculated. Each chromosome's fitness value is analyzed one by one. Compared with the existing best fitness value, if one chromosome can generate better fitness, renew the values of the defined vector and variable with this chromosome and its fitness value; otherwise, keep their values unchanged.

Step 3: Selection

Population selection within the algorithm utilizes the principle of survival of the fittest, which is based on the

Darwinian's concept of Natural Selection. A random number generator is employed to generate random numbers whose values are between 0 and 1.

Step 4: Crossover

The crossover operation is one of the most important operations in GA. The basic idea is to combine some genes from different chromosomes. A GA recombines bit strings by copying segments from chromosomes pairs.

Step 5: Mutation

Some useful genes are not generated during the initial step. This difficulty can be overcome by using the mutation approach. The basic mutation operator randomly generates a number as the crossover position and then changes the value of this gene randomly.

Step 6: Stopping Criteria

Steps 2-5 are repeated until the predefined number of generations are reached. The optimal solution can be generated after terminations.

4.2 Ant Colony Optimization

The first successful optimization algorithm based on swarm intelligence, the Ant Colony Algorithm (ACO) was introduced by Dorigo [3]. This algorithm is a technique for solving optimization problems that relies on probability. The inspiring source of ACO is the foraging behavior of ants in their search for food, i.e., over a period of time, the ants are able to determine the shortest path from their home to a food source. In nature, ants look for food randomly but having found food, on their return to the nest, they will lay down pheromone trails along the path which dissipate over time and distance. If other ants are attracted by the trails and find the food guided by the trails, the trails will be enhanced, because they leave the same thing when they go back. Over time, however, the pheromone trail starts to evaporate thus reducing its attractiveness. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have for evaporation. In the case of a short path, by contrast, pheromone density is kept at a high level because there is no time for evaporation before the laying down of a new layer of pheromones.[4]

Let H be home, F be food source, and A-B be an obstacle in the route. At time t = 0, ants choose left and right side paths uniformly in their search for food source.

At time t = 1, ants which have chosen the path from F-B-H reach the food source earlier and are returning back to their home, whereas ants which have chosen path H-A-F are still halfway in their journey to the food source.

At time t = 2, since ants move at approximately constant speed, the ants which chose the shorter, right side path (H-B-F) reach the home faster, depositing more pheromone in H-B-F route.

At time t = 3, pheromone accumulates at a higher rate on the shorter path (H-B-F), which is therefore, automatically preferred by the ants and consequently all ants will follow the shortest path. The darkness of shade is approximately proportional to the amount of pheromone deposited by ants.

The most important operations in ACO are arc selection and Pheromone Update, which constitute the foundations of the behavior of ant colonies. Arc selection describes that an ant will move from node i to node j with probability $P_{i,j}$, which is defined as:

$$P_{i,j} = \frac{\tau_{i,j}^{\alpha} \eta_{i,j}^{\beta}}{\sum \tau_{i,j}^{\alpha} \eta_{i,j}^{\beta}} \tag{5}$$

where $\tau_{i,j}$ represents the amount of pheromone in arc (i, j) and $\eta_{i,j}$ is the desirability of arc (i, j), and α , β are two adjustable parameters that control the relative weight of trail intensity and desirability.

Pheromone can be updated in:

$$\tau_{i,j} = \rho \tau_{i,j} + \Delta \tau_{i,j} \tag{6}$$

where $\tau_{i,j}$ is the amount of pheromone in given arc (i, j), Δ is the rate of pheromone evaporation and ρ is the amount of pheromone deposited, typically given by:

$$\Delta \tau_{i,j}^{k} = \begin{cases} 1 / Lk \ if \ \ ant\, k\, travels\, on\, arc\,(i,j) \\ 0 \qquad otherwise \end{cases} \tag{7}$$

where L_k is the cost of the kth ant's tour (typically length). The ACO is an efficient solution to a large variety of dynamical optimization problems.

4.3 The Bee Colony Optimization

The Bee Colony Optimization (BCO) meta-heuristic belongs to the group of Swarm Intelligence techniques. The BCO is nature-inspired meta-heuristic which is to be applied for finding solutions of difficult combinatorial optimization problems.

Artificial bees live in an environment characterized by discrete time. Colony of B artificial bees collaboratively searches for the optimal solution of a given problem. Each artificial bee generates one solution to the problem. There are two alternating phases (forward pass and backward pass) constituting single step in the BCO algorithm. In each forward pass, every artificial bee visits NC solution components, creates partial solution, and after that returns to the hive. According to the key idea in the present version of the BCO algorithm, the hive is a non-natural object, with no precise location and does not influence the algorithm execution. It is used just to denote the synchronization point at which bees are exchanging information about the current state of the search. The number of solution components NC to be visited within one forward pass is prescribed by the analyst at the beginning of the search process. For example, let bees Bee 1, Bee 2, ..., Bee B are engaged in solving a problem consisting of n components. In the case when NC = 1, at each forward pass, bees are supposed to visit a single component. [8]

Having obtained new partial solutions, the bees meet in the hive and start the backward pass. In the backward pass, all artificial bees share information about the quality of their partial solutions. Having all solutions evaluated, each bee decides with a certain probability whether it will stay loyal to its solution or not. The bees with better solutions have more chances to keep and advertise their solutions. Artificial bees that are loyal to their partial solutions are at the same time recruiters, i.e. their solutions would be advertised. Once the solution is abandoned by a bee, it becomes uncommitted and has to select one of the advertised solutions. This decision is taken with a probability too so that better advertised solutions have bigger opportunity to be chosen for further exploration. In such a way, within each backward pass, all bees are divided into two groups (R recruiters, and remaining B − R uncommitted bees) as it is shown in Figure 2. Values for R

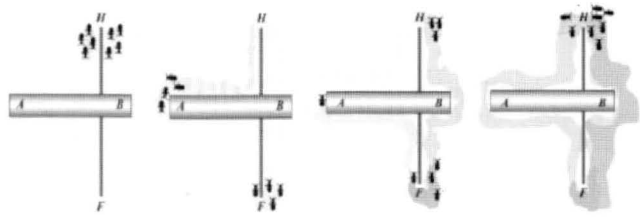

Figure 2. Illustration of ant colony principle, how real ants find shortest path in their search for food

and B − R are changing from one backward pass to another.

The case when, after comparing all generated partial solutions, Bee 2, from the previous example decided to abandon the already generated partial solution and to join Bee B, Bee 2 and Bee B "fly together" along the path already generated by the Bee B. In practice, this means that partial solution generated by Bee B is associated (copied) to Bee 2 also. When they reach the end of that common path, they are free to make an individual decision about the next constructive step to be made. The Bee 1 will keep the already generated partial solution without being chosen by any of the uncommitted hive-mates, and therefore, it will perform a new constructive step independently.

The two phases of the search algorithm, forward and backward pass, Figure 3 describes an illustration of the third forward pass, that passes are alternating in order to generate all required feasible solutions (one for each bee). When all solutions are completed, the best one is determined, it is used to update global best solution and an iteration of the BCO is accomplished. At this point, all B solutions are deleted, and the new iteration could start. The BCO runs iteration by iteration until a stopping condition is met. The possible stopping conditions could be, for example, the maximum total number of iterations without the improvement of the objective function, maximum allowed CPU time, etc. At the end, the best

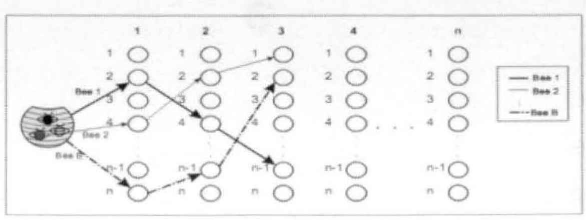

Figure 3. An illustration of the third forward pass

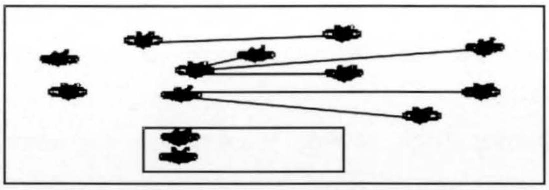

Figure 4. Recruiting of uncommitted followers

found solution (the so called global best) is reported as the final one.

The algorithm parameters whose values need to be set prior to the algorithm execution are as follows.

B - The number of bees in the hive

NC - The number of constructive moves during one forward pass.

At the start of the search, all the bees are in the hive. The pseudo-code of the BCO algorithm could be described in the following way

Step1: Initialization: an empty solution is assigned to each bee;

Step2: For each bee: // (the forward pass)

Step3: Set k = 1; // (count constructive moves in the forward pass)

Step4: Evaluate all possible constructive moves;

Step5: Choose one move using the roulette wheel;

Step6: k = k + 1; If k ≤NC Goto step (2);

Step7: All bees are back to the hive; // (backward pass starts)

Step8: Evaluate (partial) objective function value for each bee;

Step9: Each bee decides randomly whether to continue its own exploration and become a recruiter, or to become a follower;

Step10: For each follower, choose a new solution from recruiters by the roulette wheel;

Step11: If solutions are not completed Goto step 2;

Step12: Evaluate all solution and find the best one;

Step13: If the stopping criteria is not met Goto step 2;

Step14: Output the best solution found.

Steps (2) and (4) are problem-dependent and should be resolved in each BCO algorithm implementation. Figure 4 shows the Recruiting of uncommitted followers.

Conclusion

Most of the real world optimization problems often involve large scale nonlinear optimization. In the past, many optimization techniques used to find optimal solutions were constrained by the complexities of non- linear relationships in the model formulation and increase in the number of state variables and constraints. For this reason, recently many heuristic and metaheuristic global optimization algorithms have been proposed and the use of these algorithms for solving optimization problems has become more practical in the last few years. In this paper, many attempts were made to present a concise description on the well known existing global optimization methods, which provide a solution to many real world complex optimization problems that are difficult to be tackled using the traditional gradient-based methods, due to their nature that implies discontinuities of the search space, non-differentiable objective functions, imprecise arguments and function values.

References

[1]. **Al-Sultan, K.S., (1995).** *"A tabu search approach to the clustering problem Pattern Recognition",* 28(9): 1443-1451.

[2]. **Christian Blum and Andrea Roli, (2003).** "Metaheuristics in combinatorial optimization: Overview and conceptual comparison"., *ACM Computing Surveys,* 35(3):268–308.

[3]. **Dorigo, M. and V. Maniezzo, (1992).** "Optimization, Learning and Natural Algorithms". Ph.D. Thesis, *Politecnico di Milano, Italy.*

[4]. **Fred Glover and Gary A. Kochenberger, (2003).** "Handbook of Metaheuristics, volume57 of International Series in Operations Research & Management Science"., *Springer, New York, USA.*

[5]. **Glover, F., (1989).** *"Tabu search-part I., ORSA, J. Computer".,* 1(3): 190-206.

[6]. **Holland, J., (1975).** "An introduction with application to biology, control and artificial intelligence Adaptation in Natural and Artificial System", *MIT Press, Cambridge, MA.*

[7]. **Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi, (1983).** *"Optimization by simulated annealing Science",* 220(4598): 671-680.

[8]. **Shyr, W.J., (2008).** "Introduction and Comparison of Three Evolutionary Based Intelligent Algorithms for Optimal Design", *Third 2008 International Conference on Convergence and Hybrid Information Technology,* pp: 879-884.

[9]. **Teofilo F. Gonzalez, (2007).** "Hand book of Approximation Algorithms and Metaheuristics"., *Chapmann & Hall/CRC Computer and Information Science Series.*

[10]. **Thomas Back, (1996).** "Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms"., *Oxford University US.*

[11]. **Z bigniew Michale wiczand David B. Fogel, (2004).** "How to Solve It: Modern Heuristics"., *Springer, second revised and extended edition.*

ABOUT THE AUTHORS

* *Research Scholar, Department of Computer Science, SVUCCMCS, S. V. University, Tirupathi, Andhra Pradesh, India.*

** *Professor, Department of Computer Science, SVUCCMCS, S. V. University, Tirupathi, Andhra Pradesh, India.*

*** *Principal, APGCCS, Rajampet, Kadapa Dist, Andhra Pradesh, India.*

Price List/Subscription/Renewal Form-2014

Yes, I would like to Subscribe/Renew i-manager's Journals

Engineering Journals	Print		Online		Print & Online	
	Annual	Three Yrs	Annual	Three Yrs	Annual	Three Yrs
Journal on Future Engineering & Technology (JFET)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Software Engineering (JSE)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Electrical Engineering (JEE)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Electronics Engineering (JELE)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Mechanical Engineering (JME)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Civil Engineering (JCE)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Embedded Systems (JES)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Information Technology (JIT)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Structural Engineering (JSTE)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Communication Engineering & Systems (JCS)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Wireless Communication Networks (JWCN)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Instrumentation & Control Engineering (JIC)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Circuits & Systems (JCIR)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Digital Signal Processing (JDP)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Power Systems Engineering (JPS)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Computer Science(JCOM)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Material Science (JMS)	₹ 1800/-	₹ 4800/-	₹ 1400/-	₹ 3800/-	₹ 2800/-	₹ 7500/-
Journal on Mathematics (JMAT)	₹ 1200/-	₹ 3200/-	₹ 1000/-	₹ 2800/-	₹ 2000/-	₹ 5500/-

☐ Journal on Cloud Computing (JCC)	☐ ₹ 1800/-	☐ ₹ 4800/-	☐ ₹ 1400/-	☐ ₹ 3800/-	☐ ₹ 2800/-	☐ ₹ 7500/-
☐ Journal on Mobile Applications & Technologies (JMT)	☐ ₹ 1800/-	☐ ₹ 4800/-	☐ ₹ 1400/-	☐ ₹ 3800/-	☐ ₹ 2800/-	☐ ₹ 7500/-
☐ Journal on Image Processing (JIP)	☐ ₹ 1800/-	☐ ₹ 4800/-	☐ ₹ 1400/-	☐ ₹ 3800/-	☐ ₹ 2800/-	☐ ₹ 7500/-
☐ Journal on Pattern Recognition (JPR)	☐ ₹ 1800/-	☐ ₹ 4800/-	☐ ₹ 1400/-	☐ ₹ 3800/-	☐ ₹ 2800/-	☐ ₹ 7500/-
Educational Journals						
☐ Journal of Educational Technology (JET)	☐ ₹ 1800/-	☐ ₹ 4800/-	☐ ₹ 1400/-	☐ ₹ 3800/-	☐ ₹ 2800/-	☐ ₹ 7500/-
☐ Journal on School Educational Technology (JSCH)	☐ ₹ 1200/-	☐ ₹ 3200/-	☐ ₹ 1000/-	☐ ₹ 2800/-	☐ ₹ 2000/-	☐ ₹ 5500/-
☐ Journal on Educational Psychology (JPSY)	☐ ₹ 1800/-	☐ ₹ 4800/-	☐ ₹ 1400/-	☐ ₹ 3800/-	☐ ₹ 2800/-	☐ ₹ 7500/-
☐ Journal on English Language Teaching (JELT)	☐ ₹ 1200/-	☐ ₹ 3200/-	☐ ₹ 1000/-	☐ ₹ 2800/-	☐ ₹ 2000/-	☐ ₹ 5500/-
Management Journal						
☐ Journal on Management (JMGT)	☐ ₹ 1500/-	☐ ₹ 4000/-	☐ ₹ 1200/-	☐ ₹ 3600/-	☐ ₹ 2500/-	☐ ₹ 7200/-
Scientific Journals						
☐ Journal on Nursing (JNUR)	☐ ₹ 2000/-	☐ ₹ 5200/-	☐ ₹ 1500/-	☐ ₹ 4500/-	☐ ₹ 3000/-	☐ ₹ 9000/-
☐ Journal on Pharmacy (JPHARM)	☐ ₹ 3000/-	☐ ₹ 8000/-	☐ ₹ 2500/-	☐ ₹ 7000/-	☐ ₹ 5000/-	☐ ₹ 14200/-
☐ Journal on Dentistry (JDENTAL)	☐ ₹ 3000/-	☐ ₹ 8000/-	☐ ₹ 2500/-	☐ ₹ 7000/-	☐ ₹ 5000/-	☐ ₹ 14200/-

Subscription Form/Renewal Form

From (Personal Name) : ______________________ Position : ______________________

Institution/ Organisation's Name/Address : ______________________

____

____

E-mail Address : ______________________ Website (url) : ______________________

Tel. : ______________ Fax : ______________ Mobile : ______________

Payment Details : ____

We are enclosing a D.D. For Rs : ____

in favour of i-manager Publications payable at Nagercoil.

D.D. No. : ______________ Date : ______________ Bank Name : ______________

(or)

Our Bank Account Details:

Name	: i-manager Publications
Account No	: 133602000002000
Bank	: Indian Overseas Bank
Branch	: Meenakshipuram
IFS Code	: IOBA0001336

Please fill in this order form and mail to:

i-manager Publications,
3/343, Hill View, Town Railway Nagar, Nagercoil,
Kanyakumari Dist., Tamil Nadu-629 001
Ph: 04652-276675
web: www.imanagerpublications.com