A brief survey on metaheuritic based techniques for optimization problems

Kumar Dilip*
Suruchi Kaushik**

Abstract

This paper aims to provide a brief review of few popular metaheuristic techniques for solving different optimization problems. In many non-trivial real life optimization problems finding an optimal solution is a very complex and computationally expensive task. Application of the classical optimization techniques is not suitable for such problems due to its inherent complex and large search space. In order to solve such optimization problems, metaheuristic based techniques have been applied and popularized in recent years. These techniques are increasingly getting the recognition as effective tools for solving various complex optimization problems in reasonable amount of computation time. In this brief survey of metaheuristic techniques we discuss few existing as well as ongoing developments in this area.

Keywords: Optimization problems; metaheuristics; Gentic algorithm; Ant Colony Optimization

www.IndianJournals.com
Members Copy, Not for Commercial Sale
Downloaded From IP - 115.254.44.5 on dated 24-Apr-2019

I. Introduction

Application of metaheuristic based techniques for solving real life complex decision making problems is gaining popularity as the underlying search space of such problems are complex and huge in size [2,22]. Although, the heuristic based methods have been considered as a viable option for solving the complex optimization problems as they are likely to provide good solutions in reasonable amount of time. However the limitation with the heuristic based technique is the focus on the specific feature of the underlying problem, which makes the design of approach very difficult. In order to address this issue the application of metaheuristic based methods is considered as a feasible option. They are not problem specific and can be effectively adapted for the different types of optimization problems. Alternatively, the metaheuristic techniques provide a generic algorithmic approach to solve various optimization problems by making comparatively few adjustments according to problem specification. In general three common features can be identified in most of the metaheuristic techniques among others. First, majority of them are inspired by several working mechanisms of nature which include biology and physics. Second, they consider many random variables to perform the flexible stochastic search of the large search space. And third, they also involve the various parameters and proper tuning of them can greatly affect the overall performance of the techniques for the considered problem. The effectiveness of the metaheuristic technique for problem at hand significantly lies on two major concepts, known as intensification or exploitation and diversification or exploration. The exploration tries to identify the potential search area containing good solutions while exploitation aims to intensify the search in some promising area of search space. The optimal balance between these two mechanisms during search process may lead towards comparatively better solutions [2, 22].

The application of metaheuristic techniques is considered well suited for those optimization problems where no acceptable problem-specific algorithms are available for solving them. The application area of metaheruistic techniques include, finance, marketing, services, industries, engineering, multi-criteria decision making among others. These techniques may provide good or acceptable solutions to various complex optimization problems in this area with effective computation time.

Kumar Dilip*
Department of IT
IITM
Suruchi Kaushik**
Department of IT
IITM

In recent years , popular metaheuristic techniques such as Evolutionary algorithm, Genetic algorithm, Ant Colony Optimization, Particle Swarm Optimization, Bee colony optimization, Simulated Annealing, Tabu Search etc. have been widely used for different optimization problems[11,12, 13, 16, 17, 21, 24, 25, 26]. All of the above techniques have certain underlying working principle and various strategic constructs that may enable them to solve the problems efficiently. However, in recent few years a new kind of metahueristic which is unlike the above approaches, do not belong to a specific metaheuristic category but combines the approaches form the different areas like computer science, biology, artificial intelligence and operation research etc. These new class of metaheuristic techniques are normally referred as Hybrid metaheuristc. In order to improve the performance, concept of quantum computing has also been applied to solve the optimization problems. With the intent of further improving the performance of the approaches various quantum inspired metaheuristic techniques have been proposed in literatures [14].

www.IndianJournals.com
Members Copy, Not for Commercial Sale
Downloaded From IP - 115.254.44.5 on dated 24-Apr-2019

The lists of metaheuristic techniques are extensive and it is difficult to summarize them in a brief survey, this paper also not intended to do so. Rather, this paper attempt to give a brief introductory overview of few popular metaheuristic techniques. In the next section classification of the metaheuristic based techniques has been described.

II. Classification of metaheuritstic techniqeus

Many criteria can be found for the classification of various metaheuristic techniques. However the more common classification of metaheuristic techniques, based on the use of single solution and population of solutions can be found in literature. The popular single solution based techniques also known as the trajectory methods include, Simulated Annealing, Tabu Search, Variable Neighborhood Search, Guided Local Search, Iterated local search [27,28]. The single solution based approaches start with single initial solution and gradually move off from this solution depicting a trajectory movement in large search space [27, 28].

Unlike single solution based metaheuristic techniques the population based metaheuristic techniques begin with a population of solutions and in every algorithmic iteration attempt to move towards the better solutions. In recent years the population based metaheuristic techniques have been gaining comparatively more popularity and more new population based techniques are getting reported in literature [21, 22, 23]. Keeping this in mind this paper majorly focus on the population based techniques. However the details of the single solution based or trajectory based metaheuristic techniques can be found in the literature [21, 22, 23]. In the next section we describe two popular population based metahuristic techniques.

III. Population based metaheuristic techniques

The majority of population based methods either belongs to class of Evolutionary algorithms or Swarm Intelligence based methods. The inherent mechanism of evolutionary algorithm is mainly based on the Darwin's theory of the survival of the fittest. The population of solutions improves iteratively generation after generation. Fitter solutions are selected to reproduce the better solutions for the next generation. However, in Swarm intelligence based techniques, instead of a single agent, the collective intelligence of the group is exploited to find the better solutions iteratively.

Evolutionary algorithms refer to a class of metaheuristic techniques whose underlying working mechanism is based on the Darwin's theory of evolution. According to this theory the fitter living beings which can better adapt in the changing environment can survive and can be selected to reproduce the better offspring. This generic class of techniques includes evolutionary programming, Genetic algorithms, Genetic programming, evolutionary strategies etc.[15,18,19,20,29]. Though these techniques differ in their algorithmic approach, yet their core underlying working is similar. The evolutionary algorithms are mainly characterized by three important aspects, first the solution or individual representation, second the evolution function and third population dynamics throughout the algorithmic runs. All of the evolutionary techniques in every generation or algorithmic iteration attempt to select the better solutions in terms of its objective function values. These solutions further apply the mechanism of recombination and mutation operator to produce the

Procedure Evolutionry Algorithm

Begin Procedure
Initialize the population of the individuals or solutions,
Evaluate the fitness of the each individulas,
While stopping criteria not met, do
    Select the fitter individual as parents
    Recombine the pair of fitter solutions to produce offspring
    Perform the mutation on the offspring solutions
    Evaluate the new individuls or solutions
    Select the fitter solutions for the next generation
End While
End Procedure
Return solution.

Figure 1: A generic view of Evolutionary Algorithm

www.IndianJournals.com
Members Copy, Not for Commercial Sale
Downloaded From IP - 115.254.44.5 on dated 24-Apr-2019

better solutions in the next generations. Next a generic evolutionary approach has been described in order to depict the common algorithmic steps in the above evolutionary algorithms.

In the above procedure each iteration indicates a generation in which population of individuals or candidate solutions are evaluated to check its fitness according to given objective function of the problem at hand. Among those individuals the set of fitter individuals are selected by applying some suitable selection mechanism. The pairs of fitter solutions are selected to perform the recombination to produce the better offspring solutions. Further the mutation is performed on the offspring with the intent of promoting the diversity in the solutions. These newly created solutions are evaluated for the given objective function to check their suitability to use it for the next generation. The above procedure will continue iteratively till the termination condition is satisfied. The possible termination condition can be predetermined number of generation or the condition when there is no further improvement in solutions. There may also be other possible criteria for the termination of the algorithmic runs.

Genetic Algorithm (GA)

The idea of Genetic algorithm were first introduced by John Holland in 1970's. This evolutionary search Technique has been widely applied for different types of real world optimization problems. As an evolutionary technique, the concepts of Genetic algorithms are based on the Darwin's evolutionary theory in which fitter indivdulas are likely to survive and having the higher probability of production offsprings for the next genration. This very idea has been adapted in the algorithmic framework of genetic algorithms. The candiadate slutions or population of individuals iteratively evolve towards the search space of fitter or better solutions in each algorithmic iteration. In order to apply the GA for problem solving, the algorithmic requirement is to decide the repersentation of the solution or the chromosome. A binary or alphabetic string of fixed length is common representation of candidate solution in GA implementation. Next rquirement is to choose from the various selection strategy in order to select the fitter solutions, most popular selection and use of various possible crossover and mutation operators. A candidate solution is represented by a chromosome and a number of chromosomes constitute the entire population of the current generation. A population in current generation evolves to next generation through above mentioned three main operators i.e. selection, crossover and mutation. All these operators play a crucial part in the performace of the Genetic algorithm for the considered problem and their proper tuning is essential aspect of the GA implementiation. In most of the cases the focus is on the crossover as a variation operator. The crossover operator is usually applied on the pair of the selected chromosome after performing selection strategy. The various crossover operators can be found in the literature and their application may

www.IndianJournals.com
Members Copy, Not for Commercial Sale
Downloaded From IP - 115.254.44.5 on dated 24-Apr-2019

depend upon the considered problem and or also on the solution representation. With the help of crossover operator two or more solutions may exchange their genetic materials or some part of the solutions and create new individuals. The cross over rate of the population indicates the total number of chromosomes or solutions that would undergo the crossover or recombination. Each chromosome in the population has a fitness value determined by the objective function. This fitness value is used by selection operator to evaluate the desirability of the chromosome for next generation. Generally, fitter solutions are preferred by the selection operator but some less fitter chromosomes can also be considered in order to maintain the population diversity. Crossover operator is applied on the selected chromosomes to recombine them and generate new chromosome which might have better fitness. Mutation operator is applied to maintain the population diversity throughout the optimization process by introducing random modifications in the population.The Evoluationary algorithms have been applied for the optimization problems of the diverse area. It has been succesfully applied for the different combinatorial optimization problems and constrained optimization problems[7]. In recent years, it is also getting popularity in the area of multi-criteria optimization problem. Finding the trade-off solutions for the multi-objective optimization problem is a complex task. Evoluationary algorithms based techniques like NSGA-II has been successfully applied for several multi-objective optimization problem [1,3,8,9,10].

In recent years the quantum inspired Genetic algorithm is also getting a lot of attention. It applies the pricipal of quantum computing combined with evolutionary algorithm [14]. Insetead of binary, numeric or symbolic repersentation, Quantum inspired algorithm applies Q-bit repersentation and Q-gate operator is used as a variation operator.

Next we describe the swarm intelligence based technique, Ant colony optimization or ACO.

Ant Colony Optimization (ACO)

Ant colony optimization is a metaheuristic wich is inspried by the behaviour of the real ants. This approach was first applied for solving Travelling Salesman problem [5]. In majority of the cases, where ACO is applied the problem subjected to is represented with a graph. ACO is a population based metaheuristic. Various ants of real world, in search of their food, work in a group and they find the shortest path from nest to the food source. This very behaviour of real ants has inspired the ant colony optimization, in which a group of simple agents work in co-operation in order to achieve the complex task. The real world ants attempt to find the quality food sources nearest to their colony. In this pursuit they deposit some chemicals on the search path also known as pheromones. The paths with good food sources and lesser distance from nest is likely to get more amount of pheromones. Paths with higher pheromone density are highly likely to be selected by following ants. Such behaviour of ants gradually leads towards the emergence of the shortest path from nest to good food source. Alternatively, it can be observed that the indirect communication or communication through enviroment, by using pheromone trails and without any central control among ants, they are likely to find the shortest path from their colony to food source. In addition, artficial ants of Ant Colony Optimization have some extra characteristics which real ants do not have. These characteristics include presence of memory in artificial ants of ACO, which helps in constructing the feasible candidate solutions and awareness about its environment for better decsion making during the solutions construction. In ACO, ants probabilistically construct solutions using two important information known as pheromone information and heuristic information. The pheromone information $\tau(ij)$ repersents the amount of pheromone on edge or solution component (i,j) and $\eta(ij)$ repersents the preference of selection of node j from node i, during solution construction. Both of these values are reperented using numeric values. Both of these values influence the process of search towards higher pheromone values and heuristic information values. In addition, the pheromone information or denstiy on the path are updated at every algorithmic iteration. The pheromone information repersents the past search experience while heuristic information is problem specfic which remains unchanged throughout the algorithmic run of ACO. The solution in each iteration is probabilistically constructed using the following formula:

$$p(ij) = \begin{cases} \dfrac{[\tau(ij)]^{\alpha} [\eta(ij)]^{\beta}}{\sum_{l \in N} [\tau(il)]^{\alpha} [\eta(il)]^{\beta}}, & if \ l \in N, \\ 0, & otherwise \end{cases}$$

P(ij) repersents the probability of selection of node *j* after node *i* in partially consturcted solution, *l* indicates the available nodes for the solution construction or the nodes which are not already part of partially constructed solution. Here α and β indicate the relative importance for pheromone information and heuristic information respectively.

After the completion of solution construction, a mechanism of evaporation is applied with the intent of forgetting the unattractive choices and no path become too dominating as it may lead towards the premature convergence. The path update at every iteration performed using the following formula:

$$\tau(ij) \leftarrow (1-\rho)\tau(ij) + \rho.\tau(0)$$

In the above formula, ρ indicates the pheromone decay coefficient, $\tau(0)$ indicate some intial pheromone value deposited on the edge (*ij*).

In addition, daemon actions such as local search can be applied as an optional action to further improve the quality of solution. The first ant colony based optimization technique was proposed in [6] to solve the single objective optimization problems. After the initial work of ant system, many variants of ant based optimization techniques have been proposed in literature for solving various combinatorial optimization problems such as Travelling salesman problem, vehicle routing problem, production scheduling, quadratic assignment problems, among others[4,5,6]. An abstract view of the ACO is as follows:

Procedure ACO

Initialize pheromone matrix τ,
Initialize heuristic factor η,
While stopping criteria not met do
Perform *ProbailisticSolutionsConstuction( )*
Perform *LocalSearchProcess( )* // optional action
Perform *PheromoneUpdateProcess()*
End While
End Procedure
Return best solution.

Figure 2. An ACO procedure [4,5,6]

An ant based system consists of multiple stages as shown in figure 2. In the first step, evaluation function and the value of pheromone information (τ) are initialized. In the next step, at each algorithmic iteration, each ant in a colony of ants incremently constructs the solution by probabilistically selecting the feasible components or nodes from the available nodes. As an optional action, local serach can be performed for further improvement of the quality of solution. Once each ant completes the process of the solution constuction, the process of pheromone update using evaporation mechanism is performed. The best solution/solutions in terms of the value of the given objective function is chosen to update the pheromone

www.IndianJournals.com
Members Copy, Not for Commercial Sale
Downloaded From IP - 115.254.44.5 on dated 24-Apr-2019

information. The algorithmic iteration of solution construction and pheromone update ends when it meets some predefined condition and the best solution is returned. This could be some predefined number of generation or the condition of stagnation when there is no further imporvment in solution is found.

The ACO has been widely and succesfully applied for the various problems which include Travelling Salesman problem, vehicle routing, Sequential ordering, Quadratic Assignment, Graph coloring, Course timetabling, Project sheduling, Total weighted tardiness, Open shop, Set covering, Multiple knapsack, Maximum clique, Constraint satisfaction, Classification rules, Bayesian networks, Protein folding among others [4]. In recent years it has been also gaining popularity for solving various multi-objective optimization problems.

Conclusion

In this survey we have briefly described the metaheuristic based techniques for solving various optimization problems. Considering the distinction between the metaheuristic techniques based single solutions approach and population based approaches, we described introductory idea of two popular and widely used population based approaches including Genetic algorithm and Ant colony optimization.

References

1. Asllani, A., & Lari, A. (2007). 'Using genetic algorithm for dynamic and multiple criteria web-site optimizations', European journal of operational research, Vol. 176, No. 3, pp. 1767-1777
2. Basseur, M., Talbi, E., Nebro, A. & Alba, E. (2006). 'Metaheuristics for Multiobjective Combinatorial Optimization Problems: Review and recent issues', INRIA Report, September 2006, pp. 1-39
3. Coello-Coello, C. A., Lamont, G. B. & van Veldhuizen, D. A. (2007). 'Evolutionary Algorithm for solving multi-objective problems, Genetic and Evolutionary Computation Series', Second Edition, Springer.
4. Dorigo, M. & stutzle, T. (2004). Ant colony optimization, Cambridge: MIT Press, 2004
5. Dorigo, M. & Gambardella, L.M.,(1997) 'Ant colonies for the traveling salesman problem', BioSystems, vol. 43, no. 2, pp. 73–81, 1997.
6. Dorigo, M., Maniezzo,V. & Colorni, A., (1996) 'Ant System: Optimization by a colony of cooperating agents,' IEEE Transactions on Systems, Man, and Cybernetics—Part B, vol. 26, no. 1, pp. 29–41, 1996.
7. Kazarlis, S.A., Bakirtzis, A.G. & Petridis, V (1996). 'A genetic algorithm solution to the unit commitment problem', IEEE Transactions on Power System, Volume 11, Number 1, pp. 82-92
8. Deb, K., Pratap, A., Agarwal, S & Meyarivan, T. (2002). 'A fast and elitist multiobjective Genetic Algorithm: NSGA-II', IEEE Transaction on Evolutionary Computation, Vol. 6, No. 2. pp. 182-197
9. Deb, K. (2010). Multi-objective optimization using Evolutionary algorithms. Wiley India.
10. Doerner, K. F., Gutjahr, W. J., Hartl, R. F., Strauss, C. and Stummer, C (2004). "Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection," Annals of Operations Research, vol. 131, pp. 79–99,2004.
11. T'kindt, V., Monmarch´e, N., Tercinet, F. & La¨ugt, D (2002). "An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem," European Journal of Operational Research, vol. 142, no. 2, pp. 250–257, 2002
12. Wang L., Niu, Q. & Fei, M.(2007) 'A Novel Ant Colony Optimization Algorithm', Springer Verlag Berlin Heidelberg. LNCS 4688, pp. 277– 286, 2007
13. Goldberg, D. E. (1989). Genetic Algorithm in Search, Optimization and Machine Learning, Pearson Education, India

www.IndianJournals.com
Members Copy, Not for Commercial Sale
Downloaded From IP - 115.254.44.5 on dated 24-Apr-2019

14. Han, K.–H. & Kim, J.–H., (2000)'Genetic quantum algorithm and its application to combinatorial optimization problem,' in Proc. Congress on Evolutionary Computation, vol. 2, pp. 1354-1360, La Jolla, CA,2000.

15. X. Yao, Y. Liu, Fast evolutionary programming, in: Evolutionary Programming, 1996, pp. 451–460.

16. F. Vandenbergh, A. Engelbrecht, A study of particle swarm optimization particle trajectories, Information Sciences 176 (2006) 937–971.

17. S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.

18. J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems), first ed., The MIT Press, 1992.

19. T. Bäck, H.P. Schwefel, An overview of evolutionary algorithms for parameter optimization, Evolutionary Computation 1 (1993) 1–23.

20. S. Baluja, Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Technical Report, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

21. F. Glover, Tabu search for nonlinear and parametric optimization (with links to genetic algorithms), Discrete Applied Mathematics 49 (1994) 231– 255.

22. M. Birattari, L. Paquete, T. Stützle, K. Varrentrapp, Classification of Metaheuristics and Design of Experiments for the Analysis of Components, Technical Report AIDA-01-05, FG Intellektik, FB Informatik, Technische Universität Darmstadt, Darmstadt, Germany, 2001.

23. E.G. Talbi, Metaheuristics: From Design to Implementation, first ed., Wiley-Blackwell, 2009.

24. S. Jung, Queen-bee evolution for genetic algorithms, Electronics Letters 39 (2003) 575–576.

25. D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report TR06, Erciyes University, 2005.

26. D. Karaboga, B. Akay, A survey: algorithms simulating bee swarm intelligence, Artificial Intelligence Review 31 (2009) 61–85.

27. N. Mladenovic, A variable neighborhood algorithm – a new metaheuristic for combinatorial optimization, in: Abstracts of Papers Presented at Optimization Days, Montréal, Canada, 1995, p. 112.

28. N. Mladenovic, P. Hansen, Variable neighborhood search, Computers and Operations Research 24 (1997) 1097–1100.

29. X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Transactions on Evolutionary Computation 3 (1999) 82–102.

www.IndianJournals.com
Members Copy, Not for Commercial Sale
Downloaded From IP - 115.254.44.5 on dated 24-Apr-2019