Twinkle Dahiya al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 2, Issue 2, June 2015, pp. 77-80

G.A. V/S Classical Approaches For Optimisation Problems

Twinkle Dahiya[1], Er. Kapil[2], Er.Aakash Goyal[3]
[1]M. tech. Student, SITM, Sonepat
[2]H.O.D., CSE deptt., SITM, Sonepat
[3]A.P. CSE deptt., JIET, Jind, India

***Abstract*—A number of search methods are there for solving optimization problems like NP completeness problem, CPU scheduling etc. Some methods work efficiently only when it is probable to precisely calculate gradients as per variables. Some methods work for discrete variables and some for continuous. Not even a single scheme or algorithm works best on a broad variety of optimization problems. In this paper we will study the searching schemes and compare them with the genetic algorithm and also representing its advantages over others.**

***Keywords*— Genetic Algorithm, Optimization, Scheduling, Search.**

I. INTRODUCTION

The general optimization problem is described as a process of selecting among alternative plans and assigning resources and time to the set of activities in the plan. Forexample, Exam Timetabling is a special case of scheduling which concerns every teaching Institution such as schools and universities. Organizations like Universities, Polytechnics and most educational Institutions in general use timetables to schedule classes and lectures, assigning time and places to future events in a way that makes use of the available resources in the most optimal way. Exam Scheduling is a kind of problem in which events (i.e. exams, classes) have to be arranged into a number of time slots, subject to various constraints. Every semester a new timetable is to be produced by taking into account staff available, number of students and number of courses and these require a large amount of work. A poorly designed timetable is inconvenient and also takes a lot of time for generation. [1]

Also, Wireless area sensor networks is a highly distributed networks, consists of various nodes which are capable of sending and gathering the data to terminal called sink. Routing is an important issue while sending the data. With the help of various techniques delay is decreased resulting in better reliability. [2]

There are many important tasks, for which it is very difficult to find a solution, but once we have it, it is easy to check the solution. This fact led to NP-complete problems. NP stands for nondeterministic polynomial and it means that it is possible to "guess" the solution by some nondeterministic algorithm and then check it, both in polynomial time. If we had a machine that can guess, we would be able to find a solution in some reasonable time. Exam scheduling problem belongs to NP complete problem. Another problem of Distributed and Flexible Job-shop Scheduling problem (DFJS) considers the scheduling of distributed manufacturing environments, where jobs are processed by a system of several Flexible Manufacturing Units (FMUs). Distributed scheduling problems deal with the assignment of jobs to FMUs and with determining the scheduling of each FMU, in terms of assignment of each job operation to one of the machines able to work it (job-routing flexibility) and sequence of operations on each machine. Its objective is to minimize the global makes over all the FMUs. [3] For solving these problems following search methods are available.

II. SEARCH METHODS

In Exam scheduling, a set of examinations are allocated to given set of classrooms over a given period of time to avoid the occurrence of conflicts between two examinations or group of students. These concepts should be considered for the smooth conduction of exams and avoiding the redundancies with any other exam in same time. So a good scheduling technique should be used in order to get the optimized timetable. Various search methods [1] that can be used are listed below. [6]

A.Calculus-Based Search Methods: These methods are local in scope i.e. the optima they seek are best in the neighbourhood of current point. These are subdivided into two main classes: indirect and direct. Indirect method seek local maxima by solving non-linear set of equations which comes by making the gradient of objective function equal to zero. Whereas direct search method seeks local optima by hopping on the function and moving in a direction related to local gradient. The main disadvantages of calculus-based search are, firstly, a tendency for the search to get trapped on local maxima. All moves from the local maxima seem to decrease the fitness of the solution. Secondly, the application of such searches depends on the existence of derivatives, which require continuous functions. The nature of placing constraints on optimization problems tends to create a discontinuous solution space. These methods are lack in robustness.

Figure 1 shows single-peak function graph. Single-peak function is easy for calculus based method

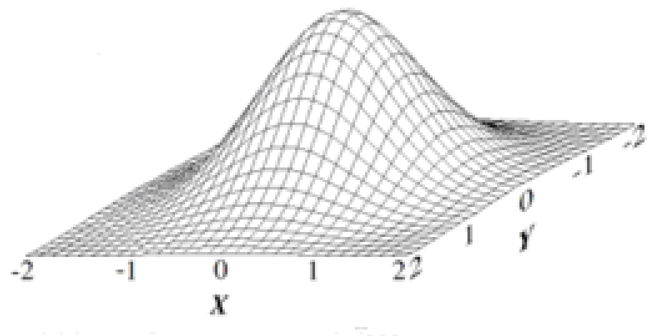

Figure 1: Single-peak function graph

The best example of calculus based search method is hill climbing.

Hill Climbing: Hill climbing is a mathematical optimization technique which belongs to family of local search. It is an iterative search algorithm. Random search and gradient search may be combined to give an "iterated hill-climbing" search. It starts with an arbitrary solution to a problem, then attempts to find a better solution by changing single element of the solution. It is good for finding local optimum but it is not guaranteed to find the best possible solution. Once one peak has been located, the hill climb is started again, but with another randomly chosen starting point. However, since each random trial is performed in isolation, no overall idea of the shape of the domain is obtained, and there is no method for eliminating trials with low probability of success. No method is made to improve good solutions, and all solutions are treated identically.Figure 2 shows multi-peak function graph which shows dilemma which hill is to climb.

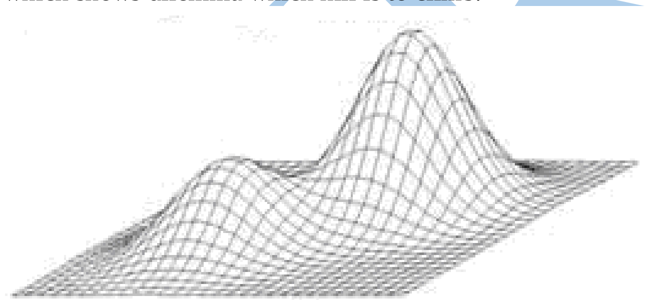

Figure 2: Multi-peak function graph

B. Enumerative Search methods these schemes have been considered in many shapes and sizes. These schemes have straight forward idea. Within a finite search space, the algorithm starts looking at objective function values at every point in space. But these schemes lacks in efficiency. The best example of these methods is Dynamic programming.

Dynamic Programming is a method for solving multi-step control problems, but can only be used where the overall fitness function is the sum of the fitness functions for each stage of the problem. Since there is no interaction between stages, the process cannot use prior stages to improve the solution. It is a method of solving complex problems by breaking them into simpler sub problems.

C. Random Search: Random search have achieved increasing popularity. These methods have potential to solve large scale problems. These methods are simple to implement. But random searches in the long run are not expected to do better. This was the traditional approach to solving difficult functions where complete enumeration was time prohibitive. Without a mechanism to evaluate the solutions and systematically improve them, this method is largely hit or miss. This method can take an unpredictable amount of time to produce acceptable results.

D. Simulated Annealing: Simulated Annealing was invented by Kirkpatrick in 1982, and is essentially a modified version of the random search and hill-climbing combination. Starting from a random point in the search space, a random move is made. If this move takes us to a higher point, it is accepted; otherwise it is accepted only with probability p(t), where t is time. The function p(t) begins close to 1, but gradually reduces towards zero - an analogy with the cooling of a solid. Therefore, initially any moves are accepted, but as the "temperature" reduces, the probability of accepting a negative move is lowered. Negative moves are essential if local maximums are to be escaped, but too many negative moves would lead the search away from the maxima. This method basically introduced the idea of 'momentum', to allow the search to escape local maxima. Stimulated annealing is often used when the search space is discrete. For certain problems it is more efficient than the exhaustive enumeration, provided het goal is merely to find good solution but not best solution. Figure 3 represents stimulated annealing method.

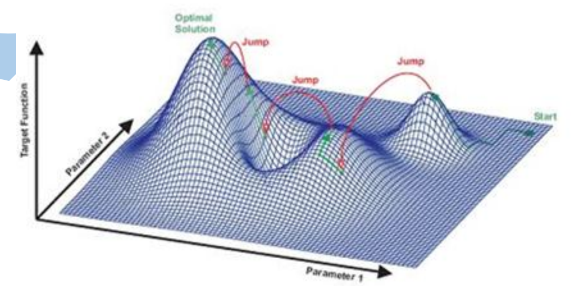

Figure 3: Stimulated Annealing Method

E. Stochastic Search Technique: GA are among several types of optimization methods that use a stochastic approach to randomly search for good solutions to a specified problem, including Simulated Annealing, Hill Climbing and numerous variations. It gives better results than various traditional methods. GA belongs to the class of probabilistic algorithms. GA is more robust than existing various search methods.

Genetic Algorithms are adaptive methods which may be used to solve search and optimization problems [4]. They are based on the genetic processes of biological evolution. GA is a powerful technique in optimization problem problems. GA is a search algorithm based on a simple idea from biology "Survival of the Fittest". They have been used for many different applications including scheduling, predicting the stock market and creating the art etc. Genetic Algorithms start with an initial random population, and subsequently allocate more trials to regions of the search space and found to have high fitness. GA can be combined with hill-climbing techniques to speed up the search process, but are able to 'jump' from local maxima because the elements that promote a good solution are being mixed up in subsequent iterations.

III. GA IMPLEMENTATION APPROACH

Genetic algorithm is fairly simple. For each generation it performs some basic operations. A simple genetic algorithm describes the following cycle:

1. Generation of random (n) chromosomes that form the initial population.
2. Assessment of each individual of the population.
3. Verification of the termination criteria.
4. If verify termination criterion - cycle ending.
5. Selection of n/2 pairs of chromosomes for crossover.
6. Reproduction of chromosomes with recombination and mutation.
7. New population of chromosomes called new generation.
8. Go back to step 2. [5]

The cycle is repeated from one generation to other and after so many generations, the probability of getting optimized solution increases. So by varying the population size and number of generations, GA is able to find the best possible solution and can be used on variety of applications. For example,

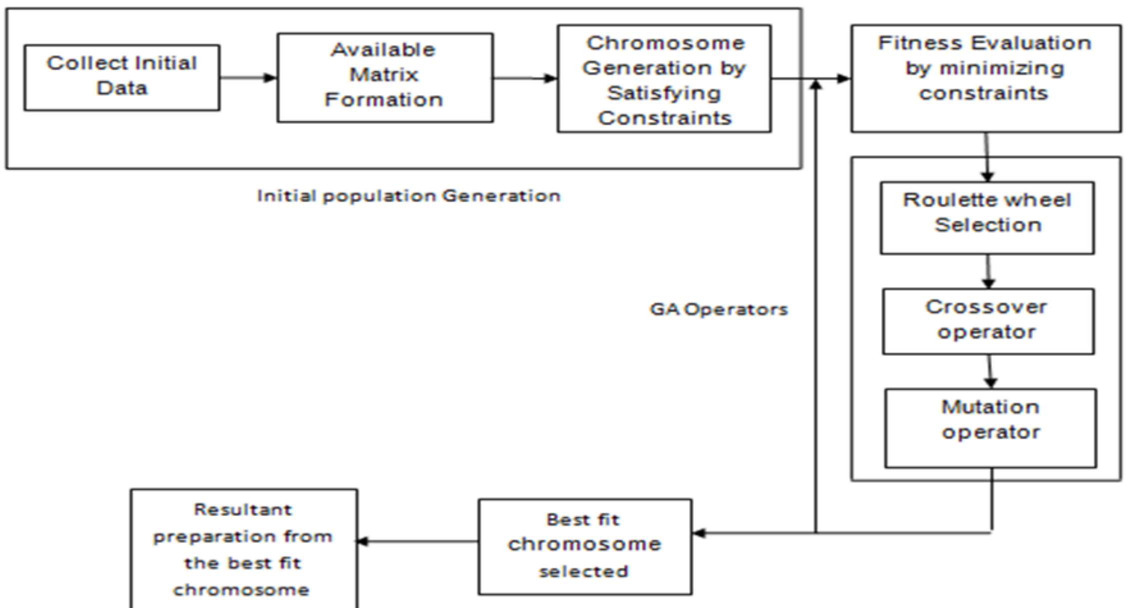

Figure 4: Block Diagram of optimization using GA

IV. COMPARISON OF GA FROM OTHER SEARCH METHODS

Goldberg (1989) presents four fundamental characteristic that differ GA from other traditional optimization and search procedures [2]. These characteristics are:

1) GA works with encoding of the parameter set, not the parameters themselves: GA require natural parameter set to be coded as finite length of strings. Coding can be done in several ways.
2) GA searches from a population of points, not from a single point: In many optimization methods, we move from single point to next in space using some transition rule to find the next point. This point to point method may locate false peaks in multimodal search spaces. But GA works from a rich database of points simultaneously, climbing many peaks in parallel. Thus probability of finding false peak reduces.
3) GA uses objective function information, not derivatives or other auxiliary knowledge: Many search methods require auxiliary information to work properly. GA has no need for such information. To perform effective search in GA for better and better results, only objective function values are needed which are associated with individual strings.
4) GA uses probabilistic transition rules, not deterministic rules: GA use probabilistic transition rules to guide their search. GA use random choice as a tool to guide search.

V. CONCLUSION AND FUTURE SCOPE

We studied various optimization problems and also studied various search method for their solution with their benefits and do representation of genetic algorithm and another techniques for solving these problems in this paper. We also presented the genetic algorithm implementation approach and presented how it is different from classical approaches. We can also extend our study to non-conventional but popular approaches like bacterial forging optimization algorithm etc. And, can solve these problems using MATLAB toolboxes and analysing the performance under different simulation criterion.

VI. REFERENCES

[1] D.E Goldberg, "Genetic Algorithms in search, Optimization and machine learning", Addison-Wisley-1989.

[2] S.Hassan paur, B. Asadi, P.Jargar, " Improving routing realibility in wireless area sensors network using genetic algorithms", 978-1-4244-8728-8/11, 2011 IEEE.

[3] L. De Giovanni, F.Pezzella, "An Improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling problem" 200 (2010) 395–408, 2010 EJOR

[4] J. E. Smith, et. al., "Operator and parameter adaptation in genetic algorithms," Soft computing: a fusion of foundations, methodologies and applications, vol. 92, pp. 81–87, 1997.

[5] Randy L. Haupt, et. al., "Practical genetic algorithms", 2004 by John Wiley & Sons, Inc., ISBN 0-471-45565-2.

[6] Bajeh,A.O,et. al.,"Optimization : A comparative study of Genetic algorithms and Tabu search algorithms", International journal of computer applications(0975-8887), vol-31, october 2011.