ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 7, July 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

A Review Paper on Genetic Algorithm

Jyoti Kumari[1]
Department of Computer Science
Rayat Bahra Institute of Mgt. &Techs,
Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat, India

Ashok Kumar Dubey[2]
Department of Computer Science
Rayat Bahra Institute of Mgt. &Tech,
Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat, India

Abstract: Genetic algorithm is a technique used for estimating computer models based on methods adapted from the field of genetics in biology. To use this technique, one encodes possible model behaviors into ''genes". After each generation, the current models are rated and allowed to mate and breed based on their fitness. In the process of mating, the genes are exchanged, crossovers and mutations can occur. The current population is discarded and its offspring forms the next generation. In this paper we will provide review of different problems that can be solved by Genetic Algorithm approach.

Keywords: Genetic Algorithm, Crossover operators, Mutation Operators.

I. INTRODUCTION

Genetic Algorithms (GA's) are adaptive methods which may be used to solve search and optimization problems. They are based on the genetic processes of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection and survival of the fittest, first clearly stated by Charles Darwin in The Origin of Species [DAV91]. By mimicking this process, genetic algorithms are able to find solutions to real world problems, if they have been suitably encoded. For example, GA's can be used to design bridge structures, for maximum strength/weight ratio, or to determine the least wasteful layout for cutting shapes from cloth. They can also be used for online process control, such as in a chemical plant, or load balancing on a multi-processor computer system.

The basic principles of GA's were first laid down rigorously by Holland in 1975. [SIV08] GA's simulate those processes in natural world which are essential to evolution. Exactly which biological processes are essential for evolution, and which processes have little or no role to play is still a matter for research but the basics are clear. In nature, individuals in a population compete with each other for resources such as food, water and shelter. Also, members of the same group often compete to catch the attention of a mate. Those individuals which are most successful in surviving and attracting mates will have relatively larger number of offspring's. Poorly performing individuals will produce few or even no offspring at all [MIT96]. This means that the genes from the highly adapted individuals will spread to a growing number of individuals in each following generation.

The combination of good characteristics from different ancestors can sometimes produce super fit offspring, whose fitness is greater than that of either parent. In this way, species evolve to become more and better suited to their environment. Genetic algorithms use a direct analogy to natural behavior. They work with a population of individuals, each representing a possible solution to a given problem [SIV08]. Each individual is assigned a fitness score according to how good a solution to the problem it is. The highly fit individuals are given opportunities to reproduce, by cross breeding with other individuals in the population. This produces new individuals as offspring, which share some features taken from each parent. The least fit members of the population are less likely to get selected for reproduction, and so die out.

In this paper we will provide review of different problems that can be solved by Genetic Algorithm approach.

II. Literature Review

The genetic algorithm belongs to the family of evolutionary algorithms, along with genetic programming, evolution strategies, and evolutionary programming. Evolutionary algorithms can be considered as a broad class of stochastic optimization techniques. An evolutionary algorithm maintains a population of candidate solutions for the problem at hand. The population is then evolved by the iterative application of a set of stochastic operators. The set of operators usually consists of mutation, recombination, and selection or something very similar. Let's discuss some of the papers published that is related to the Genetic Algorithm in the past.

Bryant [BRY00], here in this paper compared the results, which come after applying many different crossover and mutation operators devised for the traveling salesman problem and it is concluded that operators that use heuristic information or a matrix representation of the graph give the best results. Genetic algorithms are an evolutionary technique that uses crossover and mutation operators to solve optimization problems using a survival of the fittest idea. They have been used successfully in a variety of different problems, including the traveling salesman problem. In the traveling salesman problem the aim is to find a tour of all nodes in a weighted graph so that the total weight is minimized. The traveling salesman problem is NP-hard but has many real world applications so a good solution would be useful.

Enrietch [ERA00] in this paper "Evolutionary algorithms" introduced that Evolutionary Algorithms are stochastic optimization techniques based on the principles of natural evolution. An overview of these techniques is provided with the general functioning of EA's, and gives an outline of the main families into which they be divided. Subsequently, it analyzes the different components of an EA, and provides some examples on how these can be instantiated. In the end it finished with a glimpse of the numerous applications of these techniques. Various techniques are Evolutionary Programming, Evolutionary strategies, Genetic programming and Genetic Algorithms. The basic differences between these paradigms lie in the nature of the representation schemes, the reproduction operators and selection methods.

Madureira [MAD02], suggested GA for the resolution of real world scheduling problems, and proposed a coordination mechanism. Because of frequently changing dynamic environments, providing efficient production management and timely delivery are one of the hard to solve problems. Scheduling is to allocate a set of machines to perform a set of jobs within a certain time period, and the goal of scheduling is to find an appropriate allocation schedule which maximizes certain performance measure. For the implementation issues, the solutions are encoded by natural representation, and the order crossover operator is used. They used the inversion mechanism as mutation operator. Finally, Madureia et al. solved dynamic scheduling problem using a set of static scheduling schemes by GA, and they showed the feasibility of GA in Job-Shop scheduling problem.

Sandstrom [SAN02] suggested the GA which is applied for assigning task priorities and offset to guarantees that real time timing constraints. Assigning timing constraint to task is not trivial problem in real-time system. They showed how timing constraints be mapped to attributes of periodic tasks running on standard preemptive RTOS (Real-Time Operating System). They used GA because of the GA's ability to generate a result that satisfies a subset of the timing constraints in cases where it is impossible to fulfill all constraints. GA, the mechanism of natural selection, gradually improves individuals timing constraints assignment in a population. It's been tested on many test cases and showed good result.

Molga [MOL05] suggested in this paper on "Test functions for optimization needs" provides the review of literature benchmarks (test functions) commonly used in order to test optimization procedures dedicated for multidimensional, continuous optimization task. Special attention has been paid to multiple-extreme functions, treated as the quality test for opposing optimization methods (GA, SA, TS etc.). Quality of optimization procedures (those already known and those newly proposed) are frequently evaluated by using common standard literature benchmarks. There are several classes of such test functions, all of them are continuous, which are, first is unimodal, convex, multidimensional, second is multimodal, two-dimensional with a small number of local extremes, third is multimodal, two-dimensional with huge number of local extremes and the last is multimodal, multidimensional with huge number of local extremes. Where, class first contains nice functions as well as malicious cases, causing poor or slow convergence to single global extreme. Class second is intervening between first and third and last is used to test quality of standard optimization procedures in the unfriendly environment, namely that having few local extremes with single global one. Classes third and fourth are recommended to test quality of intelligent resistant optimization methods.

Omar [OMA06], proposed a method of solving job-shop scheduling using GA. They generated an initial population randomly including the result obtained by some well known priority rules such as shortest processing time and longest processing time. From there the population will go through the process of reproduction, crossover and mutation to create a new population for the next generation until some stopping criteria defined were reached. In the paper, the numbers of generations are used as stopping criterion. In crossover and mutation, the critical block neighborhood is used and the distance measured to help evaluate the schedules. Result has shown that the implementation of critical block neighborhood and the distance measured can lead to the same result obtain by other methods.

Dr. Rakesh Kumar [RKU10] suggested in his paper on “Genetic Algorithm approach to Operating system process scheduling problem”. Scheduling in operating systems has a significant role in overall system performance and throughput. An efficient scheduling is vital for system performance. The scheduling is considered as NP hard problem .The power of genetic algorithm is used to provide the efficient process scheduling. The aim is to obtain an efficient scheduler to allocate and schedule the process to CPU.

H. Nazif [HNA09] suggested in his paper on “A Genetic Algorithm on Single Machine Scheduling Problem to Minimize Total Weighted Completion Time”. In this paper, he addresses a single machine family scheduling problem where there are multiple jobs. Each job is characterized by a processing time and an associated positive weight, are partitioned into families and setup time is required between these families. For this problem, he propose a genetic algorithm using an optimized crossover operator designed by an undirected bipartite graph to find an optimal schedule which minimizes the total weighted completion time of the jobs in the presence of the sequence independent family setup times.

Snehal Kamalapur [SKA06] suggested in his paper on “Efficient CPU Scheduling: A Genetic Algorithm based Approach “that Operating system's performance and throughput are highly affected by CPU Scheduling. The scheduling is considered as an NP problem. An efficient scheduling improves system performance. In her paper she presents and evaluates a method for process scheduling. In this paper, she discussed the use of genetic algorithms to provide efficient process scheduling. And evaluate the performance and efficiency of the proposed algorithm in comparison with other deterministic algorithms and in a way that optimises some performance by simulation.

S.Ramya [SRA10] suggested in his paper on “Window Constrained Scheduling of Processes in Real Time CPU Using Multi Objective Genetic Algorithm “a new approach to window constrained scheduling, suitable for weakly-hard real-time systems. The originally developed algorithm, called Virtual Deadline Scheduling (VDS) that attempts to guarantee m out of k deadlines are serviced for real-time jobs such as periodic CPU tasks. VDS is capable of generating a feasible window constrained schedule that utilizes 100% of resources. However, when VDS either services a job or switches to a new request period, it must update the corresponding virtual deadline. This updation is a bottleneck for the algorithm, which increases the time complexity. Further, when VDS tries to solve the problem of delay the number of context switches increases. Context switching and delay are two conflicting criteria. By using Multi Objective Genetic Algorithm a trade off can be achieved between the context switching and the delay. We design our algorithm in such a way that it also overcomes the problem of updation, which is an additional overhead in the original VDS algorithm.

Birch [BIR09], in this paper presents four different versions of computationally efficient genetic algorithms by incorporating several different local directional searches into the GA process. These local searches are based on using the method of steepest descent (SD), the Newton-Rampson method (NR), a derivative-free directional search method (DFDS), and a method that combines SD with DFDS. Some benchmark functions, such as a low-dimensional function versus a high-dimensional function, and a relatively uneven function versus a very uneven function, are employed to illustrate the improvement of these proposed methods through a Monte Carlo simulation study using a split-plot design. A real problem related to the multi-response optimization problem is also used to illustrate the improvement of these proposed methods over the traditional GA and over the method implemented in the Design-Expert statistical software package. Our results show that the GA can be improved both in accuracy and in computational efficiency in most cases by incorporating a local directional search into the GA process.

III. CONCLUSION

The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. Genetic Algorithms has been widely studied, experimented and applied in many fields in engineering worlds. In this paper we have studied different problems that can be solved by Genetic Algorithm approach.

References

1. [GOL89] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Boston: Addison-Wesley.
2. [DAV91] Davis, L. (1991). Handbook of Genetic Algorithm. Von Nostrand Reinhold, Newyork.
3. [MIT96] Mitchell, Melanie (1996). An Introduction to Genetic Algorithms. MIT Press.
4. BRY00] Bryant, Kylie (2000). Genetic Algorithms and the Traveling Salesman Problem, in Proceedings of 1st GNT Regional Conference on Mathematics, Statistics and Applications.
5. [COE00] Coello, Carlos (2000). An updated survey of GA-based multiobjective optimization techniques, ACM Computing Surveys, vol. 32, no. 2, pp. 109-143.
6. [HAN00] Hanne, Thomas (2000). Global multiobjective optimization using evolutionary algorithms. Journal of Heuristics, vol. 6, no. 3, pp. 347-360.
7. [MAD02] Madureira, A., Ramos, C., & Silva, S. Carmo, (2002). A Coordination Mechanism for Real World Scheduling Problems using Genetic Algorithms. Evolutionary Computation, in Proceedings of the 2002 CEC, vol. 1, pp. 175 –180.
8. [MOL05] Molga, M. & Smutnicki, C. (2005). Test functions for optimization needs, in Proceedings of 4th Conference on Genetic Algorithms.
9. [OMA06] Omar, M., Baharum, A., & Hasan, Y. Abu (2006). A Job-Shop Scheduling Problem (JSSP) Using Genetic Algorithm ,in Proceedings of 2nd IMT-GT Regional Conference on Mathematics and Statistics.
10. [INA06] Inazawa, H. & Kitakaze, K. (2006). Locus-Shift Operator for Function Optimization in Genetic Algorithms. Complex Systems Publications, Inc.
11. [SKA06] Snehal Kamalapur (2006).Efficient CPU Scheduling: A Genetic Algorithm based Approach. IEEE, pp.206-207.
12. [DEH08] Dehuri, S. et al. (2008). Application of elitist multi-objective genetic algorithm for classification rule generation, Applied Soft Computing, pp. 477–487.
13. [SIV08] Sivanandam, S. N. & Deepa, S. N. (2008). Introduction to Genetic Algorithms. Springer.
14. [BIR09] Birch, J. B. & Wan, W. (2009). An Improved Genetic Algorithm Using a Directional Search. Tech report presented at Virginia Polytechnic Institute and State University, Blacksburg.
15. [HNZ09] H. Nazif (2009). A Genetic Algorithm on Single Machine Scheduling Problem to Minimise Total Weighted Completion Time. European Journal of Scientific Research, Vol.35 No.3, pp.444-452.
16. [RKU10] Dr. Rakesh Chawla (2010). Genetic Algorithm approach to Operating system process scheduling problem. International Journal of ngineering Science and Technology, pp. 4247-4252.
17. [SRA10] S. Ramya [2010] "Window Constrained Scheduling of Processes in Real Time CPU Using Multi Objective Genetic Algorithm"International Journal of Computer Applications Volume 1, pp.86-90.