International Advance Journal of Engineering, Science and Management (IAJESM)
ISSN -2393-8048, July-December 2021, Submitted in September 2021, iajesm2014@gmail.com

Review of In Order to Address Nonlinear or Non-Differentiable Optimization Problems, The Genetic Algorithm Is Utilized as An Optimization Tool

Dr. Satish Kumar, Associate Professor, Department of Computer Science, Government College Ateli, Distt. Mohindergarh, Haryana, India s9467723723@gmail.com

ABSTRACT

An optimization method known as the genetic algorithm is used to handle nonlinear or non-differentiable optimization problems. It looks for a global minimum using ideas from evolutionary biology. It avoids local optima and looks for global fitness since it is the resilient solution when measured against fitness standards. However, there are still some instances and circumstances where solutions are offered based on local optimal values, which completely undermines the use of GAs for optimization. By altering a portion of the data depending on a predetermined probability, the mutation operator in GA maintains uniqueness in the newly formed data values/chromosomes. The presented work optimizes the conventional GA to prevent the population from stagnating at any local optima. It utilizes Mendelian inheritance theory in which genes are inherited by the offspring in such a way that new offspring always carries the properties of both of its parents, but only one parent's properties are dominant. Here the other parent's properties are just hidden & not absent. A modified version of this theory has been used in present work to improve the problem of local optima. Further, this modified version has been used to solve a very well known problems that is De Jong-1 function & Traveling Salesman Problem (TSP). The presented work solves the TSP with the help of Mendelian inheritance by considering modified selection and crossover operations while producing the generations. Further, the proposed concept has been applied on software test field to generate test cases. In research and software testing, it's a big task to produce automatic test cases. So to tackle with such issues, this work proposes Mendel inspired genetic algorithm for auto test case generation.

KEYWORD: Algorithms, Combinatorial optimization, Computer Science, Computer Science, Information Systems, Engineering and Technology, Evolutionary computation, Genetic algorithms

Genetic programming (Computer science), Learning classifier systems

INTRODUCTION

In the first conference of Software Engineering (SE), Fritz Banner defined software engineering as ―The establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines.‖ ―Software Engineering is concerned with all aspects of software production, not just coding, but everything from understanding the user needs to maintaining that software‖ (Sommerville, 2011). Here, engineering is to mean a systematic, disciplined and managed approach to the production of software. Whenever any search technique is being applied to the SE problems, it is known as Search-Based Software Engineering (SBSE) (Harman and Jones, 2001); whereas search technique means all of the techniques such as Calculus based, Enumerative, Guided Random, Hill Climbing, Simulated Annealing, and Genetic Algorithms (GA). The term ‗search' in SBSE has been proposed by Harman and Jones (Harman and Jones, 2001). This work described Search-Based Optimization (SBO) by relating it to software engineering. Though this work was not the first attempt in this field, it gained popularity because of its well-described and useful literature review (Jiang et al., 2007). For many decades, researchers have been continuously proposing many new techniques to get optimized solutions for many software engineering related problems, but still the field fascinates authors for finding some better results. To find a suitable search technique that can be applied to software engineering, it is necessary to figure out some basic understandings of software types and optimization. The software may be described as any program that runs on the computer, but this is not a proper description of the software. There can be a computer program that does not perform any useful function for either user or the system, such a piece of code cannot be called software (Garlan and Shaw, 1993). A computer program that provides desired functionality and performance

from the user perspective is known as software. In technical terms, the software has three essential components:

i) Instruction- It is the main computer program that when executed, provides desired functions and performance.

ii) Data Structure- It enables the program to manipulate information adequately.

iii) Document(s)- It describes the operation and use of the program. The 1st component of the definition states that software must provide desired functionality and performance.

2nd component, data structure is an essential part of any computer software. It maintains the data in such a manner that a program can easily access and process program logic and design as per user requirements (Henry and Kafura, 1981). The 3rd component, documents and user manuals are required to understand how the software works. Manuals are needed to understand the internal working of the software; design manuals are also essential for maintenance purpose. If something needs to be changed in the software after delivery, it would be extremely difficult to do so in the absence of these manuals. Without design manuals, it would be difficult to locate where the source code needs to be changed and how those changes would affect the performance of the overall system (Tesch, 2013) (Chapin et al., 2001) (Guttag, Horowitz and Musser, 1978).

SEARCH-BASED SOFTWARE ENGINEERING

Streamlining & testing in SE for utilizing conventional systems have been quite difficult these days. To solve this problem, SBSE methods are acquainted with understanding of the real world and enormous scale issues proficiently. ―SBSE is the name given to an assemblage of work in which SBO is applied to software engineering‖ (Harman and Jones, 2001). It helps in dealing with SE problems effectively and conventionally. New application zones inside software engineering are developing, and a collection of proof has now emerged that shows that the searchbased methodology is digging in for the long haul (Mauša, Grbac and Bašić, 2012). SBSE reformates SE issues as search issues. In search issue ideal solutions are looked for in a search space of possible solutions, guided by a fitness function which recognizes fitter and worst solution (McMinn, 2004). SBSE has been applied to numerous fields inside the general domain of software engineering, some of which have now adequately been developed to warrant their own reviews. For instance, there are reviews covering SBSE for prerequisites, plan and testing, just as general studies of the entire field of SBSE.

CHARACTERISTICS OF SEARCH BASED SOFTWARE ENGINEERING

The different qualities of SBSE are as under (Zhang, Finkelstein and Harman, 2008) (Harman et al., 2010):

1. Generality: SBSE is widely relevant as uncovered by numerous SBSE reviews. In case, the client has no real way to address a software issue, at that point, nobody can start with any methodology; therefore, problem formulation is typical starting stage for any arrangement approach and its not only true for SBSE. There should be a proper function to evaluate the fitness using which any one can start the experimentation, and there are a number of software engineering metrics which can be immediately used fitness functions.

2. Robustness: Calculation to improve SBSE are vigorous. Frequently, the solutions which are required have some constraints. Those who are novice with SBSE, can be exhausted in tuning the parameters to obtain the optimal results from SBSE approach. There is a perception that every individual will discover that solutions obtained are generally good using the correct decisions of parameters. This the reason that progress and improvement can be done by tuning the parameters as confirmed by the facts, anyone can easily find that all the parameter decisions to beat a simply arbitrary search. In such a scenario by using a search base approach for practically any parameter selection can help progress from the.

3. Scalability through Parallelism: SBO procedures are generally referred to as being ―embarrassingly parallel‖ on account of their capabilities for versatility using execution of fitness function calculation in parallel. It has been shown by a number of researchers in the field of SBSE that parallelism can be used in SBSE work to acquire the adaptability using fitness calculations. Researchers have also indicated that General -Purpose Graphical

Processing Gadgets (GPGPUs) can be used to achieve scaleup variables of 20 in comparison to single CPU based calculations.

APPLICATIONS OF SBSE

The different uses of Search Based Software Engineering are following (Harman, 2006) (Harman, Mansouri and Zhang, 2012):

i) Search Based Software Engineering for Software Testing: The principal area of software engineering to which SBO techniques are applied are Search-Based Software Testing (SBST), and it remains the most broadly contemplated zone.

ii) Search Based Software Engineering has been used in many areas of software testing like unit testing, regression testing, model testing, temporal, stress, exception, mutation and integration testing. A search-based testing has not been applied for testing measurements.

iii) Search Based Software Engineering for Software Design: Designing of software is a major research field in search based software engineering. Software design provides expert a very complex design space in which many competing and conflicting goals are be improved and various constraints and concerns must be balanced. All things considered, the software design and configuration is a characteristic area for SBSE research. Structure level Search Based Software Engineering is the subject of an ongoing overview, mirroring its developing significance.

iv) Search Based Software Engineering for Predictions and Modeling in SE: Prediction and Modeling are the additional areas for which Search Based Software Engineering can be used for conflicting imperatives & objectives and can be used to add the best result to a lot of information, using fitness criteria and fitness.

OPTIMIZATION

For understanding the concept of optimization, awareness of simulation, regression and modelling are required. Mathematically, it is the process of finding conditions that give the maximum and minimum values of function; in that case, maximize the problem to find extremum, i.e., maximum and minimum; of course simultaneously, and it is evident that at the same time both are not possible to achieve. So, for some cases, it tends for maximum and for some, it could represent minimum. Optimization is essential for a variety of problems to simplify daily life problems. In addition to science and technology, optimization has a significant impact on the industry and management sectors. The role of optimization comes into the picture while minimizing and maximizing the objective function (Törn and Žilinskas, 1989) (Fletcher, 2013).

For example, the business world always looks for maximizing the profit and minimizing the loss. In such cases, optimization can be used. The meaning remains the same for almost all types of optimization applications and the factors that change are objective functions and decision variables. Basically, if past and current values of a variable are known, then it is possible to easily predict the sequence of all these futures values with 100% certainty in the deterministic processes. Basically, in the deterministic process, if initial conditions are known, then it is a bit easier to focus on the whole future horizon of that variable in this kind of processes. In deterministic methods, there is no error term, or the error term is zero for the whole horizon of this process.

CONVENTIONAL OPTIMIZATION AND SEARCH TECHNIQUES

Followings techniques are generally used for Optimization and Search (Hillier and Hillier, 2003):

i) Deterministic (information is sure): In each search step, it advances toward the total arrangement by settling on a deterministic choice, for example, Simplex strategy, Quasi-Newton calculations, tabu search and numerous other ordinary calculations.

ii) Stochastic Algorithms: These settle on an irregular choice at each search step e.g., random search, Simulated annealing, hill climbing, genetic algorithms, ant algorithms etc. There are two cases (Sarker, Kamruzzaman and Newton, 2003); either the accessible data is deficient, unsure, or it is unstructured because of being corrupt due to the irregular noise. The ideal data is accessible; however, because a vast search space finding a perfect solution isn't achievable,

hence an irregular component is used, which directs the search for finding the ideal solution. The stochastic calculations are based on the state-space portrayal of the search space.

iii) State Space: In order to perform the random search, hill-climbing, GA, ACO, the problem is defined as state space problem. The arrangement of every conceivable state, i.e., the courses of action of the components or segments to be utilized in taking care of an issue, frames the space of states for the issue. This is known as the state space.

SEARCH TECHNIQUES AND METHODS

In this section, various searching techniques have been discussed with the essential workout concept (Savchenko, 2016), e.g., calculus-based guided random search and enumerative techniques. the classification of various search techniques, which is further divided into calculus-based technique, guided random search technique and enumerative techniques. A lot of research on all methods is available in the literature, but researchers found guided random search more attractive.

CALCULUS-BASED TECHNIQUES

In calculus-based techniques, it is required to find out a derivative or second derivative and then apply those derivatives in a very intelligent manner, such as the use of the steepest descent method or Newton‘s method in nonlinear programming. Now it‘s a big task to find the best functional value for maxima and minima (Wuerl, Crain and Braden, 2003).

ENUMERATIVE TECHNIQUES

These techniques are enumerative in nature and categorized as another class of search techniques. The further classification is depth-first search, breadth-first search, and dynamic programming methods (Hajela and Lin, 1992).

GUIDED RANDOM SEARCH (META-HEURISTICS)

The Meta-heuristic technique is guided random search techniques, and these are all nature inspired. The techniques include hill-climbing, simulated annealing (SA), Tabu search and evolutionary algorithms. Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) are part of evolutionary algorithms. As it happens over generations, so these are called evolutionary algorithms (Voß et al., 2012).

HILL CLIMBING

It is a nearly straightforward local search technique that attempts to enhance a solitary competitor solution, starting from a randomly chosen initial stage. The neighbouring search space is assessed from the momentum position. In this, a fitter hopeful solution is discovered, and the search moves to that point. If at any point, no better solution is found in the area, the algorithm ends. The strategy has been classified as "Hill Climbing" because the procedure is compared to the moving of slopes on the surface of the fitness value (Tsamardinos, Brown and Aliferis, 2006).

SIMULATED ANNEALING

SA was first discussed by Kirkpatrick et al. in 1983 (Kirkpatrick, Gelatt and Vecchi, 1983). SA is said to be a local search technique. It is a standout amongst the most generally utilized local inquiries. It is like hill-climbing since it likewise endeavours to make move one solution at a time. It depends on the neighbourhood search and permits tough moves. SA has a specific procedure to escape from local optima by expanding neighbourhood seek strategies. It works just on the current state and moves into neighbouring states. Neighbourhood seeks have leverage that it utilizes almost no memory and can discover sensible arrangements in large or boundless (persistent) state spaces.

OPTIMIZATION BASED ON THEORY OF EVOLUTION

Various species in the nature evolve by adopting their environment. Evolution of the species is based on the theory of Darwin. Around in 1859, Charles Darwin gave the theory of evolution, based on ―Natural Selection‖. The theory stress on that species those successfully adopted the environment evolve and other seizes to survive. It states the species and their offspring will always have the best traits of their parents. This gives foundation to survival of fittest. It claims that the chromosome or attributes that are fittest will get reproduced in the offspring. Mendel in 1866 while demonstrating his experiment on peas plants showed that in the inheritance even so far invisible traits being recessive can also reappear. In 1936, R.A. Fisher reconstructed the

same experiment and showed that in 2nd and third generation recessive genes can remerge in the phenotype. Based on the Darwin Principle, Genetic Algorithms were introduced to solve problems for optimization.

GENETIC ALGORITHM

Genetic Algorithm is meta heuristic search technique based on the theory of evolution given by Charles Darwin. It uses the principle of natural selection allowing only fittest individual to involve in the reproduction of next generation. Off-springs are produced in that process which makes part of the next generation. The process is iterated to produce further generations.
The whole process involves following steps:
i) Initial Population
ii) Using fitness function
iii) Selection
iv) Crossover for exploiting best traits
v) Mutation for exploring search area

CONCLUSION

As auto test case generation is a very well-known issue in software testing. Here, proposed scheme that utilizes the Mendel operator in GA for generating test cases automatically has been presented. Also, a comparison of the performance of the proposed technique by using a traditional GA for test case generation has been presented. The obtained results prove the comparable performance of the Mendel inspired GA for autotest case generation.

REFERENCE

Abid Hussain, Yousaf. S. M. and Asim. N. (2018) _Optimization Through Genetic Algorithm with a New and Efficient Crossover Operator‘, International Journal of Advances in Mathematics (IJAM), Volume 201(1), pp. 1-14.
AbouElhamayed, Ahmed. F.M, Abdarhman. S. S, Tarek. T. S, Cherif. U, and Ahmed. H. (2016) _An enhanced genetic algorithm-based timetabling system withincremental changes‘, in 2016 11th International Conference on ComputerEngineering & Systems (ICCES). IEEE, pp. 122–127.
Afzal, W., Torkar, R. and Feldt, R. (2009) _A systematic review of search-based testing for non-functional system properties‘, Information and Software Technology (IST). Elsevier, 51(6), pp. 957–976.
Agrawal, Amritanshu, Fu, W. and Menzies, T. (2018) _What is wrong with topic modeling? And how to fix it using search-based software engineering‘, Information and Software Technology (IST). Elsevier, 98, pp. 74–88.
Ahmed, Haytham. M. A., Eltantawy, Ayman. B. and Salama, Magdy. M. A. (2016) _A planning approach for the network configuration of AC-DC hybrid distribution systems‘, IEEE Transactions on Smart Grid. Institute of Electrical & Electronics Engineers (IEEE), 9(3), pp. 2203–2213.
Aibinu, A. M. Salau, H.B. Rahman, N.A. Nwohu, M.N. andAkachukwu, C.M (2016) _A novel Clustering based Genetic Algorithm for route optimization‘, Engineering Science and Technology, an International Journal (JESTECH). Elsevier, 19(4), pp. 2022–2034.
Al-Dulaimi, Buthainah. F. Ali, and Hamza A (2008) _Enhanced Traveling salesman problem solving by genetic algorithm technique (TSPGA)‘, World Academy of Science, Engineering and Technology (WASET), 38, pp. 296–302.
Albouy, A. and Stuchi, T. J. (2004) _Generalizing the classical fixed-centres problem in a non-Hamiltonian way‘, Journal of Physics A: Mathematical and General. IOP Publishing, 37(39), p. 9109
Alhanjouri, M. A. (2017) _Optimization Techniques for Solving Traveling Salesman Problem‘, Optimization Techniques for Solving Traveling Salesman Problem. International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), 7(3).
Alshahwan, N. and Harman, M. (2011) _Automated web application testing using search based software engineering‘, in Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering. Institute of Electrical & Electronics Engineers (IEEE) Computer Society, pp. 3–12.
Angus, D. (2007) _Crowding population-based ant colony optimisation for the multiobjective Traveling salesman problem‘, in 2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making. Institute of Electrical & Electronics Engineers (IEEE), pp. 333–340.
Anvik, J., Hiew, L. and Murphy, G. C. (2006) _Who should fix this bug?‘, in Proceedings of the 28th international conference on Software engineering. Association of Computing Machinery (ACM), pp. 361–370.