Search Based Software Engineering Techniques

Arushi Jain
M.tech Student
ITM University, Gurgaon

Aman Jatain
Assistant Professor
Amity University, Gurgaon,

ABSTRACT

Search Based Software Engineering (SBSE) is the field of Software Engineering that helps in solving the problems using metaheuristic approach rather than solving the problems manually i.e. it helps in providing the automated solution for the complex problems of Software Engineering. Search Based Optimization is used by SBSE to solve the problems in Software Engineering. It is important because this approach is applied to all phases of SDLC (Software Development Life Cycle Model) i.e. from requirements specification to operation and maintenance. The problems are considered as search problem in SBSE. The search problems are finding solution of SE problems in search space which means finding optimal solution in the area. This paper explains the basic of SBSE already applied and also signifies the future development in same field.

Keywords:

Search Based Software Engineering (SBSE), Software Engineering (SE), Search Based Optimization (SBO), Genetic Algorithm (GA), Stimulated Annealing (SA), Hill Climbing (HC).

1. INTRODUCTION

Search-Based Software Engineering (SBSE) is a name given to a body of work in which Search Based Optimization is applied to software engineering. In SBSE, the term 'Search' is used to refer metaheuristic Search Based Optimization Techniques (SBO) that are used. Operational Research (OR) techniques as well as metaheuristic 'search based' techniques are used for understanding SBSE. SBSE is an approach to apply metaheuristic search techniques like Genetic algorithm, Stimulated annealing and Tabu Search to software Engineering problems. This approach offers a suite of adaptive automated and semi-automated solutions in situations typified by large complex problem spaces with multiple competing and conflicting objectives. Observations were made that many activities in Software Engineering can be formulated as optimization problems. Due to computational complexity of these problems, exact optimization techniques were impractical for large scale Software Engineering problems. Hence to find a near optimal or good-enough solutions researchers and practitioners used metaheuristic search techniques.

Software Engineering can be divided into two types-Firstly, Black Box Optimization which is a typical combinatorial optimization problem. Secondly, white box problems where operations on source code are needed to be considered. SBSE involves number of stages. First a search space and this space is typically too large to be explored exhaustively and therefore a metaheuristic is applied to sample this space, Secondly, a metric (also called fitness function) is used to measure the quality of a potential solution. SBSE is applied to almost all the phases of Software Development Process. Software Testing has been one of the major applications of search techniques in Software Engineering. Search techniques have also been applied to other Software Engineering activities for instance Requirement Analysis, Software Design, Development and Maintenance. The most widely used optimization and search techniques used are Local Search, Stimulated Annealing (SA), Genetic Algorithms (GAs), and Genetic Programming (GP), Hill Climbing (HC), Greedy Replan Algorithm, Linear Programming (LP) techniques and Integer Linear Programming (ILP). The earliest attempt to apply optimization to a Software Engineering problem was reported by Miller and Spooner [1976] in the area of software testing. The term SBSE was first used by Harman and Jones in 2001. Harman and Jones identified two key ingredients for application of SBO in Software Engineering problems: The choice of representation of a problem and the definition of the fitness function.

This paper describes the various key optimization techniques that are being used and key ingredients for successful application of optimization techniques to problems in software engineering. It covers widely used techniques in detail. It gives an overview of the previous work on successful application of optimization in eight areas of software engineering activity. It provides a future work in optimization for software engineering.

2. OPTIMIZATION TECHNIQUES

The Search Based Optimization (SBO) techniques are used to solve the problems of Software Engineering (SE).These techniques are used throughout the life cycle model of SE i.e. from requirement elicitation to software maintenance. There are many search and optimization techniques that help in solving the SE problems. The key ingredient for the problem solving techniques is Fitness function which helps in guiding the good and the bad solution. The optimization techniques are categorized into two parts: Classical Technique and Metaheuristic search Techniques. Mostly the metaheuristic search techniques are used by author.

A) Classical Techniques

Linear programming is one of the optimization techniques which are used to provide optimal solution to the problems which clearly define the single objective to be optimized and set of constraints that can be expressed as linear equations. This technique is basically used for problems of resource and plant allocation [2]. Another technique in classical techniques is Branch and Bound. Bagnall et al. used this technique for the formulation of next release problem (NLP) [5].This approach is used to handle the exponential explosions that are mostly found in search problems. Barreto et al. used this branch and bound technique for project staffing constraint satisfaction problem. Del Grosso et al. used the combination of classical and metaheuristic techniques to test buffer overflow. The problems in SBSE are mostly NP hard so these problems cannot be solved using these classical optimization techniques. So there is a need for better approach to solve these problems. Metaheuristic search techniques provided the solution of this problem.

B) Metaheuristic Search Techniques

These techniques give the better solution because it provides heuristic information regarding the search area which leads to provide near optimal solution of the SE problems. Now there is the brief overview of the most widely used metaheuristic techniques-Hill Climbing, Stimulated Annealing and Genetic algorithm.

Hill Climbing

It is the most basic and commonly used metaheuristic technique. This technique follows the method of iteration. The current solution is compared with the neighboring solution or candidate solution. If the neighbor provide better solution then we move upward on the climb and the candidate solution becomes the current solution which is again compared with the neighboring solution and the iteration continue till there is no fitter neighbor [2] .The aim is to find the solution which is the fittest among all candidate solutions.

Hill climbing technique though is most easy technique but it provides local optimal solution and not globally optimal solution. This is random search technique because initial search starts with random point and then search goes on from that particular point. This technique covers only small portion of overall search space [1]. For every search, the process is restarted again and again which leads to the over utilization of the computational resources. This repeated restart process may be considered advantageous because repeatedly restart will produce optimal solution at different places [1] [2].

Simulated Annealing

Simulated annealing was first discovered by Kirkpatrick et al in 1983[6]. Simulated annealing are said to be a local search technique. It is one of the most widely used local searches. It is similar to hill climbing because it also attempts to improve one solution. It is based on neighborhood search and allows uphill moves. Simulated annealing has an explicit strategy to escape from local optima by extending local search methods [7]. It operates only on the current state and move into neighboring states. Local search have an advantage that is they use very little memory and they can find reasonable solutions in large or infinite (continuous) state spaces. By applying SA we take the help of random search technique to find the optimal solution of objective function from the perspective of probability. We can solve large scale combinatorial optimizations with the help of SA. [8]. It can be thought as of a variation of hill climbing. Experimental tuning is very important in simulated annealing.

For implementing simulated annealing generic and Problem-specific decisions need to be taken. It deals with highly non-linear problems. When the search space is discrete then simulated annealing is used. If the cost function is expensive to the computer then repeated annealing is very slow. Other complimentary methods like Branch and Bound are required to find an optimal solution as simulated annealing cannot tell this. Simulated annealing doesn't know what the energy landscape is for a particular problem. Simulated annealing statistically guarantees to find an optimal solution. For complex problems, simulated annealing is easy to code. It generally gives a good solution. Simulated annealing does not require vector calculations.

Genetic Algorithm

Genetic algorithm was invented in the early 1970s by John Holland. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are typically applied to discrete algorithms. Genetic algorithm is a field of artificial intelligence in computer science which is a search heuristic technique. It mimics the process of natural selection in a search heuristic technique. Genetic algorithms have easy implementation but there behavior is difficult to understand. Two things are required for a genetic algorithm: a genetic representation and fitness function. A genetic representation is needed for a solution domain and a fitness function is required for evaluating a solution domain. Solutions in genetic algorithms re represented in binary as strings of 0s and 1s and other encodings. A solution is represented by an array of bits. It is not too fast but it is a good heuristic for combinatorial problems. Applications of genetic algorithm are bioinformatics, chemistry, economics, pharmacometrics, manufacturing, mathematics, computational sciences, engineering and many other fields. Two basic operators and performance of genetic algorithm depends on crossover and mutation.

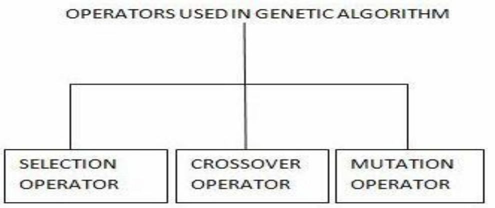

Figure 1

We start the algorithm with a set of solutions that are represented by chromosomes called as population. To form a new population, solutions from one population are taken and used. We can create a high quality solution with the help of genetic algorithm. It is used to solve a problem for which we have a very little knowledge about. Genetic algorithms have an application of finding a reasonable solution to a complex problem in a very effective way. Three operations of performed in a genetic algorithm are selection, genetic operation and replacement. To introduce variations that may be local or global, in the chromosomes mutation operator is used. Flipping of bits involve standard mutations. Crossover operator helps in differentiating genetic algorithm with other optimization techniques. They are known as unary operators where there is mutation on a single chromosome and binary operators operate on more than a single chromosome [9].

3. Fitness Function and Representation

The key ingredients for finding optimal solution of the SE problems are:

- Representation of the problem
- Fitness Function

Every Software Engineering problem has a suitable representation so the first ingredient is easily available. The representation of the SE problem is usually in the form of floating point number or binary code [4]. The fitness function is defined as a set of object oriented software metrics. The collection of these metrics is used to measure the efficiency and modifiability of software quality concept [3]. Fitness landscape is used by fitness function to determine which search technique is best suited for the SE problem and which is the best solution among the candidate solutions. Sometimes the fitness function is not easy to determine because the problem to be optimized may contain some qualities for which the determination of metrics is difficult [4].

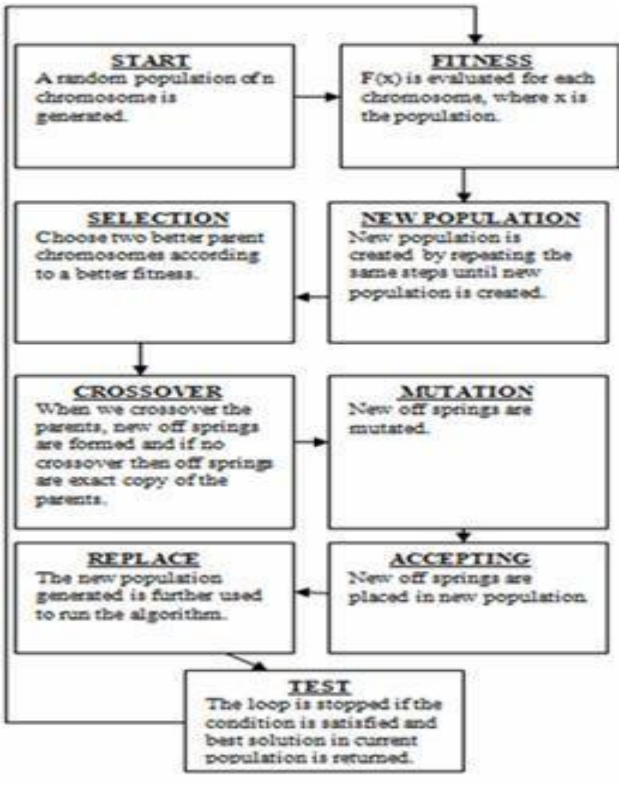
START
A random population of n chromosome is generated.
FITNESS
F(x) is evaluated for each chromosome, where x is the population.
SELECTION
Choose two better parent chromosomes according to a better fitness.
NEW POPULATION
New population is created by repeating the same steps until new population is created.
CROSSOVER
When we crossover the parents, new off springs are formed and if no crossover then off springs are exact copy of the parents.
MUTATION
New off springs are mutated.
REPLACE
The new population generated is further used to run the algorithm.
ACCEPTING
New off springs are placed in new population.
TEST
The loop is stopped if the condition is satisfied and best solution in current population is returned.

Figure 2

The fitness function is the key ingredient for applying the Search Based optimization techniques to the SE problems.

4. COMPARATIVE STUDY

Three optimization techniques described above are mostly used for solving SE problems. Hill climbing and Simulated Annealing provides local maxima whereas Genetic Algorithm provides global maxima. Hill climbing is the technique which is most easy to implement and produce very efficient result when there is limited time to perform search **[**15]**.** The problem with this technique is for every search we have to restart the search process [1]**.** Simulated annealing helps in overcoming the problem of restarting the search process by initially selecting the non fit candidate solution. It provides better solution. It has been surveyed that Hill climbing provides worst quality solution as compare to the simulated annealing technique. Simulated Annealing technique is time consuming and is slower than both Hill climbing and Genetic Algorithm [17]. Genetic Algorithm is the most used optimization technique because unlike both the technique it takes into consideration a set of candidate solution rather than a single candidate solution and hence provides a more optimal solution. Genetic algorithm is famous and advantageous because it is intrinsically parallel technique **[**17].

5. MULTI OBJECTIVE OPTIMIZATION

All the techniques read in literature deals with single objective having single fitness function. So the approach used till now was to combine all the criteria and then working on them to find one optimal solution [11]. The better approach is to combine several objectives so as to give the better result. This approach is known as Pareto optimization technique. This technique is important because now most of the Software Engineering problems are multi objective which cannot be solved using one fitness function. Moreover the importance of each objective cannot be determined when the objectives are large in number [16] [1].The set of solution of the multiple objective problems is called as Pareto optimal solution or Non dominated solution.[11].The search space in which the non-dominated solutions are there is called as Pareto Front. Pareto Front describes the overall structure of possible solution of the problem. The objectives in multi objective problems are considered to be achieved as separate goals [18].

6. TESTING IN SBSE

Search based optimization techniques were first applied to the field of search based software testing. [12].The quality of the software that is developed can be measured using software testing. We can find bugs in software by applying testing but it doesn't ensure bugs free software. Testing increases our confidence towards software being reliable and helps in saving time and cost of the software being developed [13]. Various other fields of testing in which SBSE is applied are regression testing, structural testing, integration testing, configuration testing and many others. Hill climbing approach has been applied to statistical testing to derive the optimal probability distributions. The flow chart depicts the decreasing order of the application areas in which SBSE is used. [14].

7. CONCLUSION

This paper gives a review about the metaheuristic search techniques that are applied in Search based Software Engineering. The paper shows the comparison between the search techniques Genetic algorithm, simulated annealing and hill climbing. Genetic Algorithm is proved to be the best algorithm because it focuses on solutions and fitness. They can be easily implemented and understandable in comparison to hill climbing and simulated annealing. Genetic Algorithms are best for combinatorial problems.

8. REFERENCES

[1] Mark Harman, Phill McMinn, Jerffeson eixeira de Souza, and Shin Yoo, "Search Based Software Engineering: Techniques", Taxonomy, Tutorial.

[2] Mark Harman, "The Current state and Future of Search Based Software Engineering".

[3] Mark Harman Baryan F. Jones, "Search Based Software Engineering" Information and Software Technology 43(2001)833-839.

[4] A.J Bagnall, V.J Rayward-Smith, I.M. Whitlley, "The next release problem", Information and Software Technology 43(2001) 883-890.

[5] Rob A. Rutenbar, "Simulated Annealing Algorithms: An overview", January 1989.

[6] Francisco Javier Rodriguez-Diaz, Carlos Garcia-Martinez, and Manuel Lozano, "A GA-Based Multiple Simulated Annealing" 2010, IEEE.

[7] Nansheng Pang, Yuan You, Yingling Shi, College of Business Management, China, "Application research on the Simulated Annealing in Balance Optimization of Multi-Resource Network Planning", Second international Symposium on Intelligent Information Technology Application. 2008, IEEE.

[8] K.S. Tang, K.F. Man, S. Kwong and Q. He, "Genetic Algorithms and their Applications", Signal Processing Magazine. November 1996, IEEE.

[9] Abdel Salam Sayyad, Hany Ammar. "Pareto-Optimal Search based Software Engineering", 2013 IEEE.

[10] Mark Harman and Afshin Mansouri, "Search Based Software Engineering", Introduction to the special issue of the IEEE Transactions on Software Engineering. Vol. 36, No. 6, November/December 2010.

[11] Rakesh Roshan, Rabins Porwal, Chandra Mani Sharma, "Review of Search Based Techniques in Software Testing" , International Journal of Computer Applications (0975-8887) Volume 51-No. 6, August 2012.

[12] Mark Harman, S. Afshin Mansouri and Yuanyuan Zhang. "Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications", April 9, 2009

[13] Himmat Singh, Aman Jatain, Hitesh Kumar Sharma "A Review on Search Based Software Engineering", International Journal Of Research in Information Technology.Vol.2, Issue 4,April 2014,pg:116-122.

[14] Mohamed W.Mkaouer, Marouane Kessentini, "Slim Bechick, and Daniel R. Tauritz. Prefernce-BasedMulti-Objective Software Modelling", 2013 IEEE.

[15] E-G.Talbi, P.Muntean, "Hill climbing, Simulated Annealing and Genetic Algorithms: A comparative Study and Application to the Mapping Problem", 1993 IEEE.

[16] Yuanyuan Zhang, Anthony Finkelstein and Mark Harman, "Search Based Requirements Optimization: Existing Work and Challenges".