Optimal Test Sequence Generation in State Based Testing Using Cuckoo Search

Praveen Ranjan Srivastava, Birla Institute of Technology & Science, Pilani, India
Ashish Kumar Singh, Birla Institute of Technology & Science, Pilani, India
Hemraj Kumhar, Birla Institute of Technology & Science, Pilani, India
Mohit Jain, Birla Institute of Technology & Science, Pilani, India

ABSTRACT

The present work describes a method for increasing software testing efficiency by identifying the optimal test sequences in the state machine diagram. The method employs a Meta-heuristic algorithm called Cuckoo Search to investigate best paths in the diagram. It tries to provide a technique for exhaustive coverage with minimal repetition which ensures all transitions coverage and all paths coverage at least once with minimal number of repetitions of states as well as transitions. The algorithm works by maximising an objective function which focuses on most error prone parts of the program so that critical portions can be tested first. State machine diagram is given as input, and Cuckoo Search is performed to generate a list of test sequences as output.

Keywords: Cuckoo Search, Fitness Function, Levy Flights, Random Walk, Software Testing, State Based Testing, State Model, Test Sequence Generation

1. INTRODUCTION

For the proper functioning of societies whether national or international, software engineering is the fundamental ingredient. It is almost impossible to run the modern world without software. Software systems are abstract and intangible. Throughout the 1970s and 1980s, a variety of new software engineering techniques and methods (Dijkstra, 1970; Parnas, 1972) were developed, such as structured pro- gramming, information hiding and object oriented development (Pressman, 2001).

The increasing use of software in the organisations and the costs associated with a software failure are motivating forces for well-planned software development followed by extensive testing. Generally a software development organisation incurs a cost of 30 to 40 percent in software testing of the total project development effort (Pressman, 2001). Given such a scenario, study of software testing is of utmost importance.

DOI: 10.4018/jaec.2012070102
Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test (Karner, 2006). It is an integral part of software quality assurance as it is concerned with finding faults in software. Usually, the objective of software testing is to find minimal set of test cases such that, as many faults as possible are uncovered. As mentioned previously, software testing is a very lengthy process and requires a significant amount of time (Alander, Mantere, & Turunen, 1998).

Over the years many researchers have proposed (Mayers & Glenford, 2004; Edgar, 1989) different methods for software testing. With the increasing popularity of object oriented programming languages, the research in the field of object oriented software testing has been done increasingly. As the size of the commercial softwares is usually large, automated testing tools are required (Srivastava, Ray, & Raghurama, 2009). Although a lot of research has been done in the field of software testing, optimal test sequence generation in an automated way is still an area of concern for such testing tools.

Empirical studies have shown that state based testing methods are more effective than random testing (Offutt et al., 2003). Also, the current research in software testing using Meta-heuristic techniques have shown more promising results than traditional techniques (Doungsa-ard et al., 2005; Li & Lam, 2004; Li, Zhang, & Liu, 2009; Rathore et al., 2011; Srivastava et al., 2010a, 2010b).

In view of the findings the proposed approach is a meta-heuristic method based on state transition diagram of the software under test. A study (Gandomi, Yang, & Alavi, 2011) of various structural optimization problems indicate that cuckoo search algorithm fairs better than other algorithms. Therefore, the proposed approach employs Cuckoo Search algorithm to find the best sequences in a state diagram. In 2009 Xin-She Yang and Suash Deb developed an optimization algorithm named Cuckoo search (CS) (Yang & Deb, 2009), which was inspired by obligate brood parasitism of some Cuckoo species that lay their eggs in the nests of birds of other species. In addition Levy Flight (Lecardi, 2005) distribution is used to emulate the random walk characteristic of the Cuckoo bird. The algorithm uses an objective function which focuses on most error prone parts of the program so that critical portions can be tested first. The approach tries to provide a technique for exhaustive coverage with minimal repetition.

This paper is structured as follows. Section 2 discusses the previous work in the field of software testing. Section 3 describes the Cuckoo Search algorithm and other techniques used in the proposed approach. Section 4 presents proposed approach for test sequence generation. Section 5 gives the experimental research by providing an example. Section 6 describes the analysis of the suggested approach for test sequence generation, and finally Section 7 concludes the paper.

2. BACKGROUND

At a very primary level, software testing can be divided into two broad categories namely white box testing and black box testing (Pressman, 2001). State based testing is a type of black box testing. There has been increasing level of research in the testing of object oriented programs using finite state machines. Test generation methods based on FSMs include “tour” (Naito & Tsunoyama, 1981), the “W-method” (Chow, 1978) and the “Partial W-Method” (Fujiwara et al., 1991). The research paper (Turner & Robson, 1993) draws similarities between finite state automata and state transition diagram and describes a method for testing using state machine diagram. The disadvantage with mentioned approaches is that most of these specification based testing techniques use manual methods that cannot be generalized or automated.

The approach of Delha Sokenou (2006) provides a way for test sequences coverage of state diagrams. The state diagrams are first represented as sequence diagrams and then test sequences are generated. It is assumed that the sequence and state models are consistent.
Fuzzy COTS Selection for Modular Software Systems Based on Cohesion and Coupling under Multiple Applications Environment
[www.igi-global.com/article/fuzzy-cots-selection-modular-software/74850?camid=4v1a](http://www.igi-global.com/article/fuzzy-cots-selection-modular-software/74850?camid=4v1a)