

# Chapter 16

## Test Data Generation for Branch Coverage in Software Structural Testing Based on TLBO

**Updesh Kumar Jaiswal**

*Ajay Kumar Garg Engineering College, Ghaziabad, India*

**Amarjeet Prajapati**

*Jaypee Institute of Information Technology, India*

### **ABSTRACT**

*Test data generation is forever a core task in automated software testing (AST). Recently, some meta-heuristic search-based techniques have been examined as a very effective approach to facilitate test data generation in the structural testing of software. Although the existing methods are satisfactory, there are still opportunities for further improvement and enhancement. To solve, automate, and assist the test data generation process in software structural testing, a teaching learning based optimization (TLBO) algorithm is adapted in this chapter. In this proposed method, the branch coverage convention is taken as a fitness function to optimize the solutions. For validation of the proposed method, seven familiar and benchmark software programs from the literature are utilized. The experimental results show that the proposed method, mostly, surpasses simulated annealing, genetic algorithm, harmony search, particle swarm optimization, ant colony optimization, and artificial bee colony.*

### **1. INTRODUCTION**

Software projects are being produced and developed at a rapid pace due to the impressive advancements in computer science and the notable upsurge in hardware devices. Retaining software quality has become a new challenge as a result of these rapid advances. Software testing is widely regarded as a prominent step in ensuring software quality. Researchers evaluate and scrutinize the software engineering process, modifying it to limit the number of faults that could lead to software-failure (Hetzl, 1984).

DOI: 10.4018/979-8-3693-3502-4.ch016

Structural testing mainly focuses on branches, statements, and paths coverage. Compared to functional testing, structural testing is a more cost-beneficial way to uncover program flaws, hence it has been widely used and learned (Ince, 1987).

Manual testing of the software has recently been impractical and ineffective due to the expansion and evolution of software projects, as well as the dealing with new platforms. As a result, researchers have brought up the subject of AST. The following benefits of AST over manual testing are listed (Ince, 1987): decreased testing time, decreased testing expense, and improved reliability.

In terms of AST, significant advancement and innovations have occurred in both industrial and educational areas (Habib et al., 2023). According to the findings of the studies, automated test data creation in the case of AST is admitted as one of the primary difficulties that has drawn the interest of many academicians. Even though several techniques have been proposed thus far to provide test data for AST, the issue still persists and hasn't been fully fixed.

Applications of nature-inspired heuristic search algorithms have been extensively studied in recent years for NP-hard optimization problems (Boukhelif et al., 2023). Heuristic search algorithms demonstrate strong capabilities in finding optimal solutions compared to random techniques (Harman & McMinn, 2018). An ideal solution can be found in a solution space by conducting local and global searches under the direction of fitness functions and through continuous evolution. The automatic test case generation method based on heuristic search algorithms has also garnered significant interest in the field of testing automation.

A number of search-based approaches have been used in software testing recently, including simulated annealing (Cohen et al., 2003; Tracey et al., 1998), genetic algorithms (Lin & Yeh, 2001; Minohara & Tohma, 1995; Pargas et al., 1999), harmony search (Mao, 2013), particle swarm optimization (Ahmed & Zamli, 2011; Chen et al., 2010; Kaur & Bhatt, 2011; Mao, 2014), ant colony optimization (Mao et al., 2018; Mao et al., 2014), and artificial bee colony (Aghdam & Arasteh, 2021). According to related works, the two biggest challenges in this field are convergence-speed and execution-time.

The Teaching Learning Based Optimization (TLBO) algorithm does not demand any particular parameters for self-execution (Rao et al., 2011). The TLBO algorithm works with only a few common regulating factors, such as population size and number of generations; thus, in this paper, TLBO was applied to tackle test data generation challenges, and branch coverage was used as a fitness function to optimize the proposed solutions. The key contributions of the suggested method are as follows: For the first time, test data for AST was generated using a TLBO technique. The proposed technique averaged 99.99% branch coverage. Our strategy yielded 99.98% success rate. The suggested approach reduced execution time to 0.16 ms by increasing convergence speed to 4.01.

The subsequent portions of this paper are arranged as follows: Section 2 provides a brief description of the fundamental TLBO algorithm and a summary of relevant research on automated test data creation. Section 3 presents the proposed methodology. In Section 4, the results of the experimental study are reported, and the suggested approach is contrasted with approaches from earlier studies. Section 5 concludes with some final thoughts and recommends future study areas.

## **2. BACKGROUND**

Evolutionary Algorithms and Swarm Intelligence-based algorithms are two key groups of the population-based-heuristic algorithms. The Genetic Algorithm (GA), Artificial Immune Algorithm (AIA),

11 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/test-data-generation-for-branch-coverage-in-software-structural-testing-based-on-tlbo/346335](http://www.igi-global.com/chapter/test-data-generation-for-branch-coverage-in-software-structural-testing-based-on-tlbo/346335)

## Related Content

---

### Adaptive Awareness of Hospital Patient Information through Multiple Sentient Displays

Jesus Favela, Monica Tentori, Daniela Segura and Gustavo Berzunza (2009). *International Journal of Ambient Computing and Intelligence* (pp. 27-38).

[www.irma-international.org/article/adaptive-awareness-hospital-patient-information/1370](http://www.irma-international.org/article/adaptive-awareness-hospital-patient-information/1370)

### Artificial Intelligence in the fields of Financial Management and Hospitality

Mohammad Badruddoza Talukder, Firoj Kabir and A. K. Haghi (2025). *Utilizing AI and Machine Learning in Financial Analysis* (pp. 531-554).

[www.irma-international.org/chapter/artificial-intelligence-in-the-fields-of-financial-management-and-hospitality/368346](http://www.irma-international.org/chapter/artificial-intelligence-in-the-fields-of-financial-management-and-hospitality/368346)

### AI-Powered Task-Based Learning for Cross-Border Higher Education

Rabia Khatoon (2025). *Bridging Global Divides for Transnational Higher Education in the AI Era* (pp. 141-164).

[www.irma-international.org/chapter/ai-powered-task-based-learning-for-cross-border-higher-education/361839](http://www.irma-international.org/chapter/ai-powered-task-based-learning-for-cross-border-higher-education/361839)

### An Antiwindup Approach to Power Controller Switching in an Ambient Healthcare Network

Michael J. Walsh, John Barton, Brendan O'Flynn, Martin J. Hayes, Cian O'Mathuna and Seyed Mohammad Mahdi Alavi (2011). *International Journal of Ambient Computing and Intelligence* (pp. 35-55).

[www.irma-international.org/article/antiwindup-approach-power-controller-switching/54446](http://www.irma-international.org/article/antiwindup-approach-power-controller-switching/54446)

### KStore: A Dynamic Meta-Knowledge Repository for Intelligent BI

Jane Campbell Mazzagatti (2009). *International Journal of Intelligent Information Technologies* (pp. 68-80).

[www.irma-international.org/article/kstore-dynamic-meta-knowledge-repository/2452](http://www.irma-international.org/article/kstore-dynamic-meta-knowledge-repository/2452)