

# Chapter 35

## Software Module Clustering Using Bio-Inspired Algorithms

**Kawal Jeet**

*D. A. V. College, India*

**Renu Dhir**

*Dr. B. R. Ambedkar National Institute of Technology, India*

### ABSTRACT

*Nature has always been a source of inspiration for human beings. Large numbers of complex optimization problems have been solved by the techniques inspired by nature. Software modularization is one of such complex problems that have been encountered by software engineers. It is the process of organizing modules of a software system into optimal clusters. In this chapter, some bio-inspired algorithms such as bat, artificial bee colony, black hole and firefly algorithm have been proposed for the cause of software modularization. The hybrid of these algorithms with crossover and mutation operators of the genetic algorithm has also been proposed. All the algorithms along with their hybrids are tested on seven benchmark open source software systems. It has been evaluated from the results thus obtained that the hybrid of these algorithms proved to optimize better than the existing genetic and hill-climbing approaches.*

### INTRODUCTION

Human beings are always fascinated by nature. They have always found inspiration from nature in solving complex research and engineering problems. It could be easily depicted from various new optimization algorithms that are bio-inspired. Genetic Algorithms (GA)(Goldberg, 2006), Bat algorithm (Bat)(Yang, 2010b), Artificial Bee Colony algorithm (Mala & Mohan)(Yan, Zhu, Zou, & Wang, 2012), Black Hole algorithm (BH)(Hatamlou, 2013), Firefly algorithm (Kononoff, Lehman, & Heinrichs; Yang, 2010a) are the names of few of them. GA has been inspired by Darwin's theory of survival of the fittest bat algorithm has been inspired by the technique followed by bats to locate its prey in dark, ABC is an algorithm that has been inspired by swarm behavior of honey bees; BH has been inspired by the gravity of black holes in universe where gravitational force is so strong that even light can't get escape and FF

DOI: 10.4018/978-1-7998-3016-0.ch035

has been inspired by the flashing characteristics of fireflies to attract other fireflies. These algorithms are widely used for solving various complex optimization problems ranging from routing in wireless networks to image optimization.

In this paper, some of the bio-inspired algorithms have been used for modularization of software systems. Software modularization is the process of allocating modules to subsystems or clusters. This allocation should lead to efficient modularization of the software system concerned. It should be done in such a way that there should be minimum coupling and maximum cohesion among the subsystems and clusters thus generated. It is particularly useful to

- Recover the modularization of the system when source code is the only means available to get information about the system.
- Identify the best possible package to which classes of a java project should be allocated to its actual delivery.
- Combine the classes that could be downloaded together.

Software modularization is particularly useful during the maintenance phase of the software development life cycle. Maintenance is the largest and most expensive phase of the software development lifecycle (IEEE Standard for Software Maintenance). During this phase, the software system undergoes continuous change and is enhanced for a number of reasons. It is generally accompanied by the turnover of the software engineers involved in its development. It ultimately leads to the decay of the original clustering of the modules of the system concerned. It is called software erosion. For the effective and efficient maintenance of the software system, it is important to re-identify the modularization of this system. It might be a difficult task, especially in the absence of original design documents.

In this proposed work, bio-inspired algorithms such as bat, ABC, BH, and FF have been used for solving the problem of software modularization. These algorithms work to find optimized results but might get stuck at local optima. Experiments have been conducted to use the hybrid combination of these algorithms with crossover and mutation operators of genetic algorithm for software module clustering. Adding these operators has an inherent advantage of adding randomization which saves these algorithms to get stuck at local optima. Although we observed good quality modularization but we recommend the resulting modularization to be approved by the experts. In other words, it could be used as guidance to the experts but could not replace them.

The rest of the chapter is organized as discussed here. The literature of the field of software modularization, as well as applications of bio-inspired algorithms, has been discussed. Further, a concept of search based software engineering has been discussed along with the skeleton of bio-inspired algorithms. Software modularization using these bio-inspired algorithms has been elaborated. The experimental setup which includes the information about the samples, encoding scheme and algorithmic parameters used has been given. Results are discussed and analyzed followed by the discussion about threats to the validity of this research work, conclusion, and future work.

## **BACKGROUND**

Wiggerts (Wiggerts, 1997) very well introduced clustering techniques that have been successfully applied to system modularization. Various other clustering techniques like Rigi (Müller, Orgun, Tilley, &

24 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/software-module-clustering-using-bio-inspired-algorithms/261054](http://www.igi-global.com/chapter/software-module-clustering-using-bio-inspired-algorithms/261054)

## Related Content

---

### Developing Context Sensitive BIM Based applications

Timo Hartmann (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 361-376).

[www.irma-international.org/chapter/developing-context-sensitive-bim-based/62453](http://www.irma-international.org/chapter/developing-context-sensitive-bim-based/62453)

### Topology-Aware Load-Balance Schemes for Heterogeneous Graph Processing

(2018). *Creativity in Load-Balance Schemes for Multi/Many-Core Heterogeneous Graph Computing: Emerging Research and Opportunities* (pp. 113-143).

[www.irma-international.org/chapter/topology-aware-load-balance-schemes-for-heterogeneous-graph-processing/195894](http://www.irma-international.org/chapter/topology-aware-load-balance-schemes-for-heterogeneous-graph-processing/195894)

### Archiving Nature's Heartbeat Using Smartphones

Jinglan Zhang, Paul Roe, Binh Pham, Richard Mason, Michael Towsey and Jiro Sumitomo (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications* (pp. 1896-1912).

[www.irma-international.org/chapter/archiving-nature-heartbeat-using-smartphones/62552](http://www.irma-international.org/chapter/archiving-nature-heartbeat-using-smartphones/62552)

### Identifying and Analyzing the Latent Cyber Threats in Developing Economies

Atul Bamrara (2018). *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 1044-1059).

[www.irma-international.org/chapter/identifying-and-analyzing-the-latent-cyber-threats-in-developing-economies/203547](http://www.irma-international.org/chapter/identifying-and-analyzing-the-latent-cyber-threats-in-developing-economies/203547)

### The Role of Compliance and Conformance in Software Engineering

José C. Delgado (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 103-133).

[www.irma-international.org/chapter/the-role-of-compliance-and-conformance-in-software-engineering/192874](http://www.irma-international.org/chapter/the-role-of-compliance-and-conformance-in-software-engineering/192874)