وزارة التعليم العالي والبحث العلمي
جامعة الموصل
كلية علوم الحاسوب والرياضيات
قسم البرمجيات

تحسين تجميع الوحدات البرمجية باستخدام الخوارزميات الإرشادية

رسالة مقدمة
الى مجلس كلية علوم الحاسوب والرياضيات في جامعة الموصل
كجزء من متطلبات نيل شهادة ماجستير علوم في
البرمجيات

من قبل

علي حسين علي شحنة

إشراف
أ.د. دجان بشير طه علي

١٤٤٦ هـ ٢٠٢٥ م

المستخلص

تُعد مرحلة الصيانة في هندسة البرمجيات الأكثر إستهلاكاً للموارد، إذ غالبًا ما تشكل نسبة كبيرة من إجمالي التكاليف والجهود المبذولة في دورة حياة النظام. ومن الأساليب الأساسية لمعالجة تحديات الصيانة هي تجميع وحدات البرمجيات – Software Module Clustering) (SMC، وتُعد هذه التقنية إحدى ممارسات الهندسة العكسية، إذ تُستخدم في تحليل الأنظمة البرمجية التي تفتقر إلى التوثيق الكافي، بهدف تحسين قابلية الصيانة والفهم للنظام من خلال استخراج نموذج هيكلي من الشيفرة المصدرية. ومع ذلك، فإن مشكلة SMC تُصنف ضمن المشكلات الصعبة من الناحية الحسابية (NP–hard) ؛ لأنها تتطلب تحسين تكوين المجموعات لتقليل الأقتران بين المجموعات وزيادة التماسك داخل كل مجموعة.

تُقدم هذه الدراسة حل جديد لهذه المشكلة، إذْ تجمع بين ثلاث تقنيات متقدمة: خوارزمية تحسين الأرملة السوداء(Black Widow Optimization) ، خوارزمية التحسين المستندة الى التعليم والتعلم(Teaching–Learning–Based Optimization) ، فضلاً عن خوارزمية هجينة تدمج بين الطريقتين(BWO–TLBO Optimization). وقد تم تطبيق هذه التقنيات لتجميع وحدات البرمجيات، مع تقييم أدائها بناءً على معايير رئيسة تشمل :الاستقرار، وجودة التجميع(Modularity Quality – MQ) . بالإضافة إلى ذلك، فقد تم استخدام خوارزمية لوفين (Louvain) لتحديد المجتمعات في رسوم الاعتماديات بين الوحدات البرمجية (Module Dependency Graphs – MDG) بهدف تحديد العدد الأمثل للتجمعات. كما استكشفت الدراسة تأثير طرائق مختلفة لتهيئة الحلول الأولية جودة واستقرار عملية التجميع وهذه الطرائق هي الخرائط الفوضوية المنطقية كنظام لورنز(Lorenz System) ونظرية الفوضى(Chaos Theory)، فضلاً عن التهيئة العشوائية؛ وقد أُجريت التجارب على ثمانية تطبيقات مفتوحة المصدر مكتوبة بلغةPython ، ذات مستويات تعقيد متفاوتة. تساهم هذه الدراسة في توفير حل فعال وقابل للتوسّع لمشكلة SMC من خلال الاستفادة من تقنيات تحسين متقدمة وطرائق تهيئة فوضوية. يدعم الإطار المقترح استراتيجيات تجميع أفضل، مما يساعد في صيانة وتطوير هياكل البرمجيات المعقدة.

أظهرت نتائج الدراسة أن الخوارزمية الهجينة (BWO–TLBO) قدّمت أداءً متفوقاً مقارنةً بالخوارزميات الفردية TLBO وBWO، وذلك بالاعتماد على معيار جودة التجميع (Modularity Quality – MQ) بوصفه المقياس الرئيس لتقييم جودة المخرجات.

وبلغ متوسط قيمة MQ للخوارزمية الهجينة عبر جميع حالات الاختبار 6.6445، مقارنةً بـ 6.3183 لخوارزمية BWO و4.3757 لخوارزمية TLBO. كما سجّلت الخوارزمية الهجينة أدنى انحراف معياري، مما يدل على ثبات واستقرار أدائها عبر التجارب المختلفة.

أما فيما يتعلق بتهيئة الحلول الأولية، فقد أظهرت النتائج أن استخدام التهيئة الفوضوية، وخصوصًا نظام لورنز (Lorenz System) ، أدى إلى تحسين واضح في كل من متوسط MQ والانحراف المعياري، مقارنةً بأسلوب التهيئة العشوائية، مما يعزز من فعالية الاستكشاف الأولي في فضاء الحلول ويقلل من التشتت في النتائج.

Ministry of Higher Education and
Scientific Research
University of Mosul
College of Computer Science and
Mathematics
Department of Software

Enhancing Software Modules Clustering Using Heuristic Algorithms

**A Thesis Submitted to the Council of the College of
Computer Science and Mathematics
University of Mosul
as a Partial Fulfillment of Requirements
for the Degree of Master of Science
in Software
By**

Ali Hussein Ali

Supervised by

Prof. Dr. Dujan Basheer Taha

2025 A.D. **1446 A.H.**

Abstract

The maintenance phase in software engineering is considered the most resource-consuming, often accounting for a large proportion of the total costs and efforts throughout the system's life cycle. One of the fundamental approaches to addressing maintenance challenges is Software Module Clustering (SMC), a technique rooted in reverse engineering practices. It is employed to analyze software systems that lack sufficient documentation, aiming to enhance system maintainability and comprehensibility by extracting a structural model directly from the source code. However, the SMC problem is classified as computationally NP-hard, as it requires optimizing the configuration of clusters to minimize inter-cluster coupling and maximize intra-cluster cohesion.

This study presents a novel solution to the SMC problem by combining three advanced techniques: the Black Widow Optimization (BWO) algorithm, the Teaching-Learning-Based Optimization (TLBO) algorithm, and a hybrid BWO-TLBO optimization algorithm that integrates both methods. These techniques were applied to the clustering of software modules, with their performance evaluated based on key metrics, including stability and modularity quality (MQ). Additionally, the Louvain algorithm was used to identify communities within the Module Dependency Graphs (MDG) to determine the optimal number of clusters.

The study also explored the impact of different initial solution generation methods on the quality and stability of the clustering process, including chaotic maps such as the Lorenz system, chaos theory approaches, and random initialization. Experiments were conducted on eight open-source Python applications of varying complexity levels.

This study contributes an efficient and scalable solution to the SMC problem by leveraging advanced optimization techniques and chaotic initialization methods. The proposed framework supports superior clustering strategies, facilitating the maintenance and evolution of complex software architectures.

The study's results showed that the hybrid BWO-TLBO algorithm outperformed the standalone TLBO and BWO algorithms based on the modularity quality (MQ) metric, which served as the primary measure for

evaluating clustering output quality. The average MQ value achieved by the hybrid algorithm across all test cases was 6.6445, compared to 6.3183 for BWO and 4.3757 for TLBO. Furthermore, the hybrid algorithm recorded the lowest standard deviation, indicating the consistency and stability of its performance across different experiments.

Regarding initial solution generation, the results revealed that using chaotic initialization—particularly the Lorenz system—led to significant improvements in both the average MQ and the standard deviation, compared to random initialization. This finding highlights the enhanced effectiveness of early-stage exploration in the solution space and the reduction of result variability.