



جامعة الموصل
كلية علوم الحاسوب والرياضيات

توليد المجموعة المثلى لمسارات الاختبار باستخدام خوارزمية مستعمرة النمل العكسية

عبد الله هاني أحمد

رسالة ماجستير

هندسة البرمجيات

بإشراف

د. دجان بشير طه

أستاذ مساعد

المستخلص

مع التطور الحاصل في استخدام البرمجيات في مجالات الحياة المختلفة فقد ازداد الإهتمام بجودة البرمجيات للتأكد من أن البرمجيات التي تم تطويرها تتوافق مع المتطلبات الخاصة بها. يعدُّ اختبار البرمجيات أحد أهم فعاليات ضمان الجودة للبرمجيات. إن مرحلة الاختبار هي أهم مراحل تطوير البرمجيات كما انها المرحلة الأكثر كلفة إذ إنها تتطلب الكثير من الوقت والجهد للتأكد من أن البرنامج يؤدي العمل المطلوب منه. خلال السنوات الأخيرة اهتمت هندسة البرمجيات بمجال الاختبار الآلي للبرمجيات للتقليل من كلفة عملية الاختبار والحصول على نتائج أكثر دقة بالمقارنة مع الاختبار اليدوي. تعدُّ عملية توليد حالات الاختبار بصورة آلية إحدى أهم مجالات البحث في مجال الاختبار الآلي للبرمجيات.

تم في هذا البحث بناء أداة برمجية تعمل على توليد حالات الاختبار المثلى للبرامج المكتوبة بلغة Java وإجراء عملية الاختبار بصورة آلية بدلاً من عملية الاختبار اليدوي. تم برمجة الأداة باستخدام لغة Java وقد تم توظيف الذكاء الاصطناعي في العمل عن طريق استخدام خوارزمية مستعمرة النمل العكسية (Anti-ant Colony Algorithm) لتوليد مجموعة المسارات المثلى التي تغطي جميع الجمل البرمجية للبرنامج قيد الاختبار وتم توليد بيانات الإدخال التي تضمن تنفيذ كل مسار من مجموعة المسارات المثلى للبرنامج وإجراء التنفيذ الحقيقي للبرنامج قيد الاختبار وعرض نتائج التنفيذ.

تم تطبيق عمل الأداة المقترحة على مجموعة من الدوال البرمجية لتوليد حالات الاختبار المثلى للدوال وإجراء عملية الاختبار بصورة آلية، وقد أثبتت الأداة المقترحة قدرتها على توليد المسارات المثلى للبرنامج قيد الاختبار إذ تم توليد أقل عدد ممكن من المسارات التي تغطي جميع الجمل البرمجية للبرنامج قيد الاختبار. وقد نجحت الأداة المقترحة في تقليل التكلفة والوقت والجهد المطلوب في عملية الاختبار إلى درجة كبيرة بالمقارنة مع الاختبار اليدوي وبعض طرائق الاختبار الآلي وتم تحقيق نسبة تغطية عالية في عملية الاختبار تجاوزت 90% من البرنامج قيد الاختبار.

**University of Mosul
College of Computer Sciences
And Mathematics**



Generation of Optimal Testing Paths set using Anti –Ant Colony Algorithm

Abdullah Hani Ahmed

**M.Sc./Thesis
Software Engineering**

**Supervised By
Dr. Dujan Basheer Taha
Assistant Prof.**

Abstract

With the development of the use of software in various areas of life, the importance of software quality has increased to ensure that the software developed is compatible with its requirements. Software testing is one of the most important quality assurance activities for software and it is one of the most important stages of software engineering. Software testing is the most expensive process in software development because it takes a lot of time and effort to make sure that the program performs the required work. In recent years, software engineering has focused on the field of automated software testing to reduce the cost of the testing process and to obtain more accurate results compared to manual testing. The process of automatic generation of test cases is one of the most important areas of research in the field of software automated testing.

In this research, a software tool was created to generate the optimal test cases for programs written in Java language and to perform the testing process automatically instead of manual testing, the proposed tool is programmed using Java language. The artificial intelligence has been employed in this work. The anti-ant colony algorithm was used to generate optimal paths set covering all statements of the program under test. Input data were generated to ensure that each path of the program's optimal paths was implemented. The real execution of the program is done and finally then tool displays the results of the program execution.

The proposed tool has been applied to a set of software functions to generate the optimal test cases for these methods and to perform the testing process automatically. The proposed tool demonstrated its ability to generate the optimal paths of the program under test as the minimum number of paths that cover all code statements of the program under test has been generated. The proposed tool has significantly reduced the cost, time and effort required in the testing process compared to the manual test and some automated testing methods and a high coverage rate was achieved in the testing process that exceeded 90% of the program under test.