Selection and Prioritization of Test Cases by using Bees Colony

Shahbaa I. Khaleel Ragad waleed khaled
shahbaaibrkh@uomosul.edu.iq
College of Computer Sciences and Mathematics
University of Mosul

Received on: 15/8/2013 **Accepted on:11/11/2013**

ABSTRACT

In this research, the bees swarm intelligence was studied to appointment it to serve software engineering. And that was performed through using Artificial Bees Colony ABC Algorithm in selection of test cases for the software written by C++ language in an automatic way since to enable the corporation which develops the software to save time, effort and costs that required for testing phase and regression testing activity, which is always evaluated by 50% of the product cost. The proposed work can reduce test cases that are used in the tests of software and in regression testing activity ,also will make prioritization to the test cases, that are produced by the best selection process, by using Greedy Algorithm and Genetic Algorithm. the proposed work was applied practically on some programs – that differ in number of lines of code-.the result that appeared reduce number of test cases and make test cases in certain ordering that assists testing and regression testing for the software in safe mode and short time .

Keywords: swarm intelligence , Artificial Bees Colony, test cases , prioritization, Regression Testing Activity.

اختيار حالات الاختبار وتعيين أسبقيتها باستخدام مستعمرة النحل

شهباء إبراهيم خليل ***رغـد وليد خالد***
كلية علوم الحاسوب والرياضيات
جامعة الموصل

تاريخ استلام البحث : 2013\8\15 **تاريخ قبول البحث : 2013\11\11**

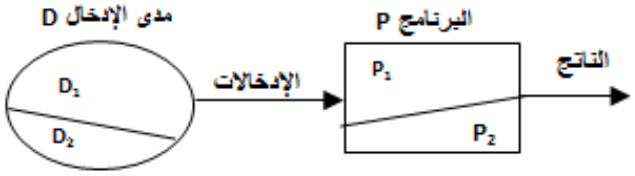
الملخص

في هذا البحث تمت دراسة ذكاء سرب النحل بغية توظيفه في خدمة هندسة البرمجيات , وقد أُنجز ذلك من خلال استخدام خوارزمية مستعمرة النحلة الاصطناعية Artificial Bees Colony ABC في انتقاء واختيار حالات الاختبارTest Case Selection للمسارات الأساسية للبرمجيات المكتوبة بلغة C++ وبشكل تلقائي Automaticallyمما سيوفر للمؤسسة المطورة للبرمجيات الوقت والكلفة لمرحلة الاختبار Testing Phase وفعالية الاختبار التراجعيRegression Testing Activity التي تقدر عادة بنحو 50% من كلفة إنتاج المنتج .ويقوم العمل على التقليل من حالات الاختبار المستخدمة في عملية الاختبار والاختبار التراجعي للبرمجيات وتعيين الأسبقية لحالات الاختبارTest Case Prioritization باستخدام خوارزمية اختيار الأفضل Greedy Algorithm والخوارزمية الجينيةGenetic Algorithm . وقد تم تطبيق البحث عملياً على عدة برامج, المختلفة في عدد الأسطر البرمجية, وأظهرت النتائج تقليل من عدد حالات الاختبار ووضع حالات الاختبار في ترتيب معين, الذي يساعد على تسهيل اختبار البرمجيات والاختبار التراجعي , في وقت قصير.

الكلمات المفتاحية : سرب الذكاء ، مستعمرة النحل الاصطناعي ، حالات الاختبار ، تحديد الأسبقيات،فعالية الاختبار التراجعي.

1. المقدمة

احد المشاكل في الاختبار هو مطالبة الزبون بالاختبار الكامل الذي يعني بعدم وجود أي خطأ غير مكتشف في نهاية عملية الاختبار وهذا شبه مستحيل في اغلب الأنظمة وذلك لعدة أسباب منها, قد يكون مدى الإدخال للبرنامج كبيراً لحد ما يصعب أن يستخدم بالكامل في اختبار النظام بالإضافة الى ان الكلفة العالية من ناحية المصادر والوقت بالإضافة الى الكلفة المادية. البرنامج ربما تكون له حالات كثيرة , الإدخال الصحيح والإدخال



غير الصحيح والإدخال المشروط بوقت حيث يكون صحيحاً في وقت معين وغير صحيح بوقت أخر, فمدى الإدخال للنظام يكون كبيراً جداً لا يستخدم بالكامل في اختبار البرنامج .الحل لهذه المسألة يكمن في اختيار مجموعة جزئية من مدى الإدخال لكي يتم الاختبار بها والشكل يوضح ذلك[6].

شكل (1) جزء من مدى الإدخال يفحص جزء من سلوك البرنامج

ليكن مدى الإدخال للبرنامج P هو D, يتم اختيار مجموعة هي D_1 التي هي جزئية منD , لاختبار البرنامج P من الممكن ان D_1 تختبر فقط الجزء P_1 الذي هو جزء من P, وفي هذه الحالة الأخطاء الموجودة في الجزء الثاني P_2 لا يتم اكتشافها . بإختيار مجموعة من مدى الإدخال, D_1, مهندس الاختبار يحاول استنتاج خصائص للبرنامج P بكامله وذلك بملاحظة سلوك الجزء الأول P_1 من سلوك البرنامج P باستخدام مجموعة الإدخال D_1 , ولذلك اختيار مجموعة من مدى الإدخال يجب ان يجري بعناية وطريقة نظامية وبذلك يكون الاستنتاج دقيق وكامل قدر الإمكان[6]. إن اختيار مجموعة صغيرة من حالات الاختبار تشكل عقبة أمام المختبِر وكفاءة الاختبار[4]. فانطلاقا من هذه النقطة يتم في هذا البحث تغطية مسألة اختيار مجموعة من حالات الاختبار ووضعها بترتيب حسب أسبقيتها في محاولة لتقليل كلفة عملية الاختبار وتكلفة الاختبار التراجعي Regression Testing.

2. الاعمال السابقة

اقترح Bharti suri وآخرون 2011 تقليل مجموعة الاختبار التراجعي باستخدام تقنية مهجنة بين تقنية أفضلية مستعمرة النحل BCO والخوارزمية الجينية GA وقد تناول بها طريقة لتقليل تكلفة الاختبار التراجعي بالاعتماد على تقنية تقليل مجموعة حالات الاختبار وذلك باختيار مجموعة جزئية من حالات الاختبار المتوفرة التي تغطي كل الأخطاء المكتشفة في اقل وقت للتنفيذ ولكن لم يطبق البحث بشكل عملي أي لم يتم أتمتته , اقترح Zheng Li واخرون 2007 استخدام خوارزميات البحث search Algorithms لتعيين الأسبقية لحالات الاختبار في الاختبار التراجعي, تقوم فكرة البحث على استخدام بعض خوارزميات البحث مثل Hill Climbing, Genetic Algorithm والخ، في تعيين الأسبقية لحالات الاختبار. قدمت Sangeeta واخرون 2011 استخدام الخوارزمية الجينية في تعيين الأسبقية لحالات الاختبار للاختبار الساكن, اقترحوا تعيين أسبقية حالات الاختبار المشتقة من الشفرة المصدرية باستخدام المفهوم الأساس لمقياس تدفق المعلومات information flow metric

IF والخوارزمية الجينية. أما siripong & jirapun فقد اقترحا 2010 طريقتين جديدتين الأولى تعيين الأسبقية لحالات الاختبار التي تملك نفس الوزن والطريقة الثانية هي تعيين الأسبقية لعدد من مجاميع الاختبار, مجموعة الاختبار هي عبارة عن عدد من حالات الاختبار. Arvinda & Shubhra 2011 استخدام الخوارزمية الجينية لتعيين الأسبقية لحالات الاختبار باستخدام مقياس تغطية الشفرة Code Coverage ولكن لم يطبق البحث بشكل عملي أي لم تتم أتمتته.

3. اختبار البرمجيات والاختبار التراجعي

عملية اختبار البرمجيات هي عملية تعريض نظام برمجي لعدد من الاختبارات في محاولة لكشف الأخطاء والتأكد من انه يؤدي الوظيفة التي صمم من اجلها. ومرحلة الاختبار تمثل احد اكبر مراحل تطوير البرمجيات كلفةً. وعندما يتم إجراء تعديل على البرنامج يجب أن يعاد اختباره مرة ثانية للتأكد أن الإصدار الجديد لا يحتوي على أخطاء ولا يؤثر على النسخة القديمة, ولهذا الغرض فأن الاختبارات السابقة التي أُجريت على النسخة القديمة يُعاد تنفيذها على النسخة المعدّلة للبرنامج وهذا ما يدعى بالاختبار التراجعي Regression Testing [2].

فالاختبار التراجعي هو عملية إعادة تنفيذ الاختبارات على النسخة المعدّلة للنظام البرمجي للتأكد من صحة النسخة المعدّلة وكذلك التأكد من أن التغيير على البرنامج لا يؤثر على الوظيفة التي تؤديها النسخة السابقة[2][1]. والاختبار التراجعي هو احد أهم فعاليات مرحلة الصيانة maintenance phase في التطوير البرمجي , والاختبار التراجعي هو فعالية مكلفة حيث أنها تستغرق وقتاً كبيراً بالإضافة الى الجهد وتستهلك نصف كلفة مرحلة الصيانة[9][3]. ولحل هذه المسألة هناك تقنيات الاختبار التراجعي التي تشمل :[7][2]

1. تقنية إعادة الاختبار Retest Technique : وهي من أسهل تقنيات الاختبار التراجعي تقضي بإعادة الاختبار باستخدام كل حالات الاختبار المتوفرة للتحقق من صحة النسخة المعدّلة , وهي تعد من أكثر التقنيات أمناً ولكنها حالة خاصة فقط عندما يكون حجم مجموعة الاختبار صغيراً إلا أنها تكون مكلفة عندما تتطلب المسألة وقت ومصادر[2].

2. تقنية تقليل مجموعة الاختبار Test Suite Reduction:هذه التقنية تحاول بشكل دائم تقليل حجم مجموعة الاختبار وذلك بحذف حالات الاختبار الزائدة التي لا تساهم بأي تغطية, هذه التقنية تُحدث انخفاض طفيف في فعالية الكشف عن الخطأ[2] .

3. تعيين الأسبقيات لحالات الاختبار Test Case Prioritization تقنية تحتفظ بمجموعة الاختبار بالكامل ولكن تحاول ترتيب حالات الاختبار بطريقة ما لتقليل الوقت الذي يلزم لتطبيق متطلبات الاختبار وهناك عدة معايير تستخدم لعمل الأسبقية لحالات الاختبار منها الاعتماد على معايير التغطية. كل هذا يساعد في إيجاد الأخطاء مبكرا خلال عملية الاختبار, وبسبب ان هذه التقنية تقوم بإعادة تنفيذ كل حالات الاختبار فهي تعتبر آمنة[2]. في هذا البحث تم استخدام هذه التقنية ولكن بعد عملية الاختيار لحالات الاختبار.

4. اختيار حالات الاختبار regression test case selection : وهي تقنية تحدد مجموعة جزئية من مجموعة الاختبار لاختبار التغيير الذي طرأ على البرنامج. هناك عدد من الطرق والتقنيات المستخدمة وهي:[2]

- تقنية التقليل للحد الأدنى minimization technique :تهدف لتحديد اصغر مجموعة من حالات الاختبار التي تحقق بعض معايير التغطية. بعض التقنيات تستخدم تقليل مجموعة الاختبار بشكل كبير ولكن قد يؤدي الى ظهور مشكلة فقدان بعض الأخطاء .[7][2]

- تقنية التغطية Coverage technique:هذه التقنية تشبه تقنية التقليل للحد الأدنى حيث تستخدم معيار تغطية معين لاختيار الحالات. اختلافها عن تقنية التقليل للحد الأدنى أنها لا تحاول تقليل مجموعة الاختبار[7][2].
- تقنية الأمان safe technique : وهي تهدف لاختيار كل حالة اختبار تنتج إخراج مختلف لنسخة البرنامج المعدلة عن الإخراج الذي كانت تعطيه النسخة القديمة عند الاختبار, وهذه التقنية لا تعتمد بالدرجة الأساس على معيار تغطية, قد يكون أداء هذه التقنية ضعيفاً تحت ظروف معينة فمثلا عند وجود اختلاف كبير بين النسخة القديمة والنسخة المعدلة أو عندما يكون الاختبار على أجزاء كبيرة من الشفرة المصدرية[7][2].

5. التقنية المدمجة: هذه التقنية تكون عبارة عن دمج بين التقنيات السابقة [7].

ان اختيار احد هذه الاستراتيجيات يعتمد على التطبيق وحالته فإذا تطلب التطبيق أن يتم الاختبار في اقصر وقت ممكن يتم اختيار تقنية التقليل للحد الأدنى minimization technique اما اذا كان التطبيق لا يتحمل وجود خطأ قد يظهر لاحقا فيتم استخدام تقنية الأمان.[2]

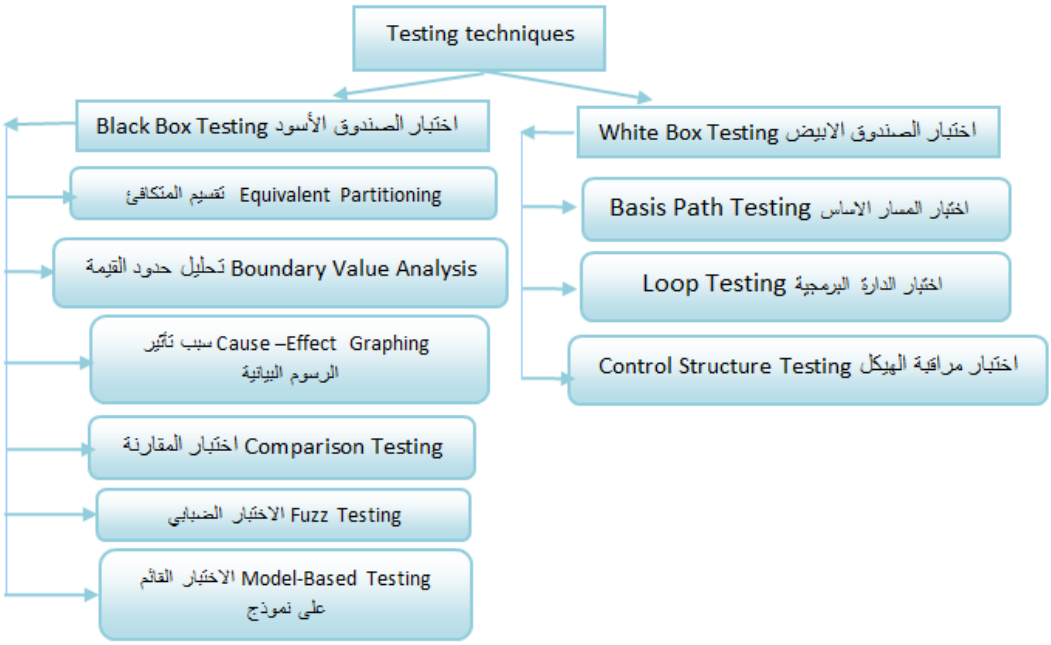
في هذا البحث أُستخدمت تقنية اختيار حالات الاختبار باستخدام خوارزمية مستعمرة النحلة الاصطناعية ABC بالاعتماد على معيار تغطية الشرط Condition Covered, استخدمت تقنية التقليل للحد الأدنى لبرنامج إعطاء تقدير الطالب بالاعتماد على المعدل المدخل والنسخة المعدلة حساب المعدل أما برنامج الحاسبة العادية والنسخة المعدلة له هي الحاسبة العلمية استخدمت تقنية التغطية فقط حيث لا يتم تقليل حالات الاختبار. وبعد الحصول على الحالات المطلوبة يتم تعيين أسبقية prioritization لكل حالة اختبار بالاعتماد على معيار عدد الشروط المغطاة.

4. اختبار الصندوق الأبيض

هو طريقة لتصميم حالات الاختبار والذي يستخدم مراقبة هيكل التصميم الإجرائي أي الشفرة المصدرية لاشتقاق حالات الاختبار, باستخدام طرائق اختبار الصندوق الأبيض يستطيع مهندس البرمجيات اشتقاق حالات الاختبار التي [5][4]:

1. تضمن كل المسارات المستقلة أن يتم فحصها على الأقل مرة واحدة .
2. اختبار كل القرارات المنطقية في حالة كونها صائبة أو خاطئة .
3. تنفيذ كل الدارات في حدودها .
4. تختبر هياكل البيانات الداخلية للتأكد من صحتها.

استخدام اختبار الصندوق الأبيض الذي يعتمد على الشفرة المصدرية يتم بإستخدام مخطط سيطرة التدفق Control Flow Graph CFG لتوضيح المسارات الأساسية للبرنامج حيث يتم اختيار المسارات في مخطط سيطرة التدفق بالاعتماد على الجمل الشرطية if, for loop, while loop, switch case المتواجدة في البرنامج [6]. وهناك نوعان آخران من الاختبار يدعى الاول اختبار الصندوق الأسود والثاني اختبار الصندوق الرمادي [5][4]. وفي هذا البحث تم اعتماد اختبار الصندوق الأبيض , كما موضح في الشكل التالي:



شكل (2) تصنيف تقنيات الاختبار

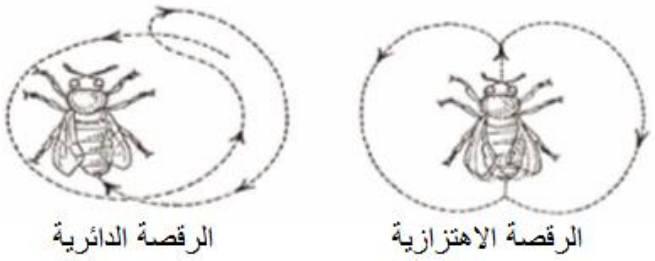
5. سلوك النحل في الطبيعة

تتكون خلية النحل في الطبيعة من الملكة والذكور والعاملات حيث ان لكل منها وظيفة خاصة بها تؤديها لخدمة الخلية فالذكر مسؤول عن عملية التزاوج والملكة مسؤولة عن عملية وضع البيض أما النحل العامل فهو مسؤول عن خدمة الخلية فبعضها يهتم بالبيض والآخر يهتم في الدفاع عن الخلية والبعض الأخر مسؤول عن الغذاء[12].

وهنا في هذا البحث يتم التركيز على السلوك الذي تسلكه النحلة للحصول على الغذاء.حيث ان مجموعة من النحل العامل مسؤول عن الغذاء فتقوم بضع نحلات من هذه المجموعة في البحث عن مصادر للغذاء كبستان من الأزهار أو مجموعة من الأشجار للحصول على الرحيق أو أقراص من العسل المكشوف أو حبوب لقاح وعندما تجد النحلة مصدراً للغذاء تقوم بالعودة الى الخلية لإخبار بقية النحل عن المصدر وذلك بأداء بعض الحركات والتي تدعى رقصات النحل ويتم الإخبار عن موقع واتجاه وكمية الغذاء وجودته من خلال تلك الرقصات, تتغير هذه الرقصة بزيادة المسافة الى مصدر الغذاء، وتم التعرف على أنواع مختلفة من هذا السلوك ومنها[12].

- الرقصة الدائرية dance Round
- الرقصة الاهتزازية (هز الذنب) Waggle tail dance

وكل نوع منهم يمكن أن يكون به درجات أو اختلافات توضح سلوك معين ,كما في الشكل(3).



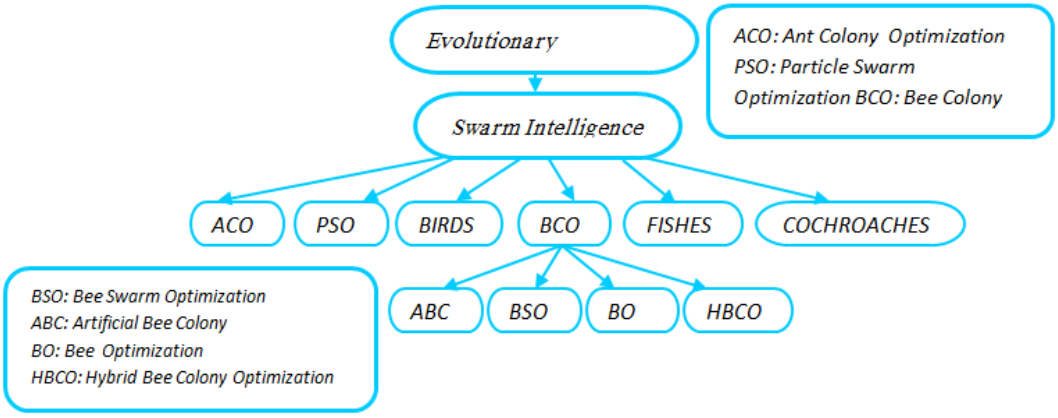
شكل (3) صورة توضيحية للحركات التي تؤديها النحلة

تقوم النحلة في الرقصة الدائرية بالجري دائرياً حيث يتم عمل عدة دوائر بواسطة النحلة مع عمل وقفات عند إعطاء عينات من الرحيق للنحل الذي يتعلم ويتبع الرقصة. في حالة النحلة العاملة أي الشغالة الراجعة بحبوب اللقاح للخلية يمكن للنحل المتتبع لها الحصول على المعلومات عن المصدر بالملامسة المباشرة لكتل حبوب اللقاح في رجلي الشغالة الراقصة. يوضح الرقص مكان المصدر في مدى مسافة معينة ولأن المصدر يكون قريباً فلا تعطى الرقصة الدائرية فكرة عن الإتجاه, تلتقط النحلة المتتبعة للرقصة الرائحة ويتم إعطاؤها عينة في حالة الرحيق أو العسل المسروق وبالتالي تعلم الطعم وتركيز السكر وهنا تغادر الشغالات الخلية للبحث عن الغذاء دون معرفة للاتجاه. ويتم البحث في كل الاتجاهات[12] .

عند وجود الغذاء بعيدا عن الخلية يتم تطوير الرقصة الدائرية الى الرقصة الاهتزازية أو رقصة هز الذنب Wagtailوهي على شكل رقم 8 بغرض إمداد نحل الخلية بمعلومات عن مسافة واتجاه الغذاء وذلك بالجري الدائري في إتجاه ثم الجري الدائري في الإتجاه الآخر مع هز الجسم اثناء الجري المستقيم، وتتم هذه الرقصات على قرص موضوع أفقياً يسمى منطقة الرقص وهو محدد المكان في الخلية. يقوم النحل باستخدام الشمس ليقيس زوايا اتجاهات الغذاء ، ولا يلزم للنحل رؤية الشمس للإستدلال على اتجاه مصدر الغذاء إذ يمكنه رؤية الضوء المستقطب والذي لا يراه الإنسان وذلك في مجال الأشعة فوق البنفسجية والذي يراه النحل جيداً. فإذا رقصت نحلة فوق القرص بزاوية° 20 يمين الخط الرأسي مع الشمس فمعنى هذا ان الطيران يكون بزاوية ° 20 يمين الشمس وهكذا. في هذه الحالة فان الـ° 360 درجة يمكن ان يتم توضيحها أو عملها فوق سطح القرص ويتم توضيح المسافة وعند تحرك الشمس يرجع النحل للخلية ليبلغ النحل بالتغيير الحاصل في الإحداثيات ويتعلم النحل هذه الأمور في أول أيام حياة الشغالة البالغة[12] .

6. نظام مستعمرة النحل Bees colony System

واحدة من نظم تصوير العمل الجماعي والعمل التعاوني هي Bees colony optimization BCO, وهو المجال الذي برز للباحثين في مسائل الأمثلية لأنه يوفر الحل للمشاكل الصعبة NP-hard problem وأسلوب تقنية BCO من الأسفل إلى الأعلى في النمذجة bottom up modeling حيث تشكل عناصره الحل الشامل عن طريق الاستفادة المثلى من الحلول المحلية. أفضلية مستعمرة النحل BCO هو صنف خاص من ذكاء السرب Swarm Intelligence SIحيث العمال (الأعضاء /الوكلاء) للمجموعة هم نحل العسل. يتواصلون عن طريق آلية تسمى "الرقص الاهتزازي " وتبادل المعلومات المهمة بخصوص موقع مصدر الغذاء الغني. والشكل (4) يوضح التمثيل الهرمي للخوارزميات التطويرية[13].



شكل (4) التمثيل الهرمي للخوارزميات التطويرية

7. سلوكيات النحل التي تم تقليدها اصطناعياً

سلوك البحث عن الطعام ، والتعلم، وحفظ المعلومات ومشاركتها هي خصائص للنحل والتي أصبحت مؤخرا واحدة من مجالات البحث الأكثر إثارة للاهتمام في ذكاء السرب. الدراسات على نحل العسل هي في اتجاه متزايد خلال السنوات القليلة الماضية. ويتم تصنيف الخوارزميات بشأن الخصائص السلوكية لنحل العسل الى:

- السلوكيات الغذائية.
- سلوكيات التزاوج.
- مفهوم الملكات[19] .

8. خوارزمية مستعمرة النحلة الاصطناعية ABC

خوارزمية مستعمرة النحلة الاصطناعية ABC اقترحت من قبل karaboga في عام 2005 لمعايير الأفضلية الحقيقية، وهي خوارزمية أمثلية تحاكي السلوك الغذائي لمستعمرة النحل. يتكون نموذج الحد الأدنى من ذكاء السرب في اختيار الغذاء في مستعمرة نحل العسل التي تحاكيه خوارزمية ABC من ثلاثة أنواع من النحل : النحل العامل employed bees والنحل المشاهد onlookers والنحل الكشفي scout bees [14]. نصف مستعمرة النحل تتكون من العاملين، ويشمل النصف الآخر النحل المشاهد. النحل العامل هو المسؤول عن استغلال مصادر الرحيق وإعطاء المعلومات إلى النحل المشاهد الذي ينتظر في الخلية عن جودة موقع مصدر الغذاء الذي يحاولون استغلاله. النحل المشاهد يتخذ قرار بشأن مصدر الغذاء للإستغلال استناداً إلى المعلومات التي يتشاركها مع النحل العامل. النحل الكشاف يبحث عشوائياً في البيئة من أجل العثور على مصدر الغذاء الجديد اعتماداً على الدافع الداخلي أو على أساس القرائن الخارجية المحتملة[14]. في الخوارزمية ABC اقترح karaboga موقع لمصدر الغذاء الذي يمثل حلاً ممكناً لمسألة الأفضلية ، وكمية الرحيق من مصدر الغذاء يتوافق مع ربحية الحل (اللياقة) المرتبطة بها. يتم استغلال كل مصدر غذاء من قبل نحلة عاملة واحدة فقط. وبعبارة أخرى، فإن عدد النحل العامل هو مساوٍ لعدد مصادر الغذاء الموجودة حول الخلية (عدد من الحلول في المجتمع) والنحل العامل الذي قام بالتخلي عن مصدر الغذاء يصبح الكشاف[14].

تعتبر خوارزمية ABC من أفضل الطرق التي تستخدم لاختبار البرمجيات والتي تتميز بكفاءتها من حيث استهلاكها لأقل وقت ممكن وسهولة تنفيذها مقارنة بالخوارزمية الجينية. فهناك بعض المشاكل التي تواجهها الخوارزمية الجينية تم حلها من قبل خوارزمية ABC حيث ان الخوارزمية الجينية تتضمن عدم وجود تخزين أي

التذّكر memorization, والتأخر في التقارب للحل, كما ان الخوارزمية الجينية لا تدعم الحل الأمثل العام حتى وان وصلت إليه, لكن ABC نموذج أساسه تحسين مجموعة الاختبار حيث يولد حلول قريبة من الحل الأمثل العام ويغطيه ضمن اقل عدد ممكن من تنفيذ الاختبار, ولذلك يمكن القول أن ABC هي من أفضل الطرق في بيئة اختبار البرمجيات [10].

1.8. الخوارزمية الأساسية لمستعمرة النحلة الاصطناعية Artificial Bees Colony (ABC)

1. إنتاج مواقع لمصدر الغذاء الأولية: إذا تم اعتبار فضاء البحث هو البيئة التي توجد بها الخلية التي تحتوي على مواقع لمصدر الغذاء ، تبدأ الخوارزمية بإنتاج مواقع لمصادر الغذاء بشكل عشوائي والتي تطابق الحلول في فضاء البحث. ويتم إنتاج مصادر الغذاء الأولية بشكل عشوائي ضمن نطاق حدود المعايير[14].

$$x_{ij}=x_j^{min}+rand(0,1)\left(x_j^{max}-x_j^{min}\right) \quad ,i=1,\dots,SN \quad ,j=1,\dots,D. \qquad \dots\dots(1)$$

SNهو عدد من المصادر الغذائية وD هو عدد قيم الأمثلية أو الأبعاد لفضاء البحث [14],[15] وبالإضافة إلى ذلك ، العدادات التي تخزن عدد التجارب من الحلول يتم إعادة تعيين قيمتها في هذه المرحلة إلى 0 بعد التهيئة الابتدائية للقيم، يتعرض المجتمع من مصادر الغذاء (حلول) لتكرار دورات من عمليات البحث من النحل العامل employed bees، والنحل المشاهدonlooker Bees والنحل الكشفي scout Bees , معايير الإنهاء لخوارزمية ABC يمكن التوصل إلى عدد محدد من الدورات MCN أو الوصول الى ما يطابق اقل نسبة خطأ[14].

2. إرسال النحل العامل employed Bees إلى مواقع مصدر الغذاء: كما ذكر سابقاً ، كل نحلة عاملة ترتبط مع موقع واحد لمصدر الغذاء فقط . وبالتالي فإن عدد المواقع لمصادر الغذاء هو مساوٍ لعدد النحل العامل employed Bees,النحلة العاملة تُحدث التعديل على الموقع لمصدر الغذاء (الحل) في ذاكرتها اعتماداً على المعلومات المحلية (المعلومات البصرية) وتجد مصدر الغذاء المجاور ، ومن ثم تقييم جودتها. وفي خوارزمية ABC، يتم إيجاد مصدر الغذاء المجاور حسب المعادلة التالية.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \qquad \dots\dots(2)$$

المنطقة المجاورة لمصدر الغذاء تتمثل بـ x_i ، مصدر الغذاء v_i تتعين قيمته بالتغيير على قيمةx_i. j هو عدد صحيح عشوائي في النطاق [1, D] وk هو مؤشر يتم اختياره عشوائياً ونطاقه [1….SN] ويختلف عن i ، ϕ_{ij} وهو رقم عشوائي حقيقي في النطاق[-1,1], [14] .

الفرق بين القيم x_{ij} و x_{kj} يتناقص ، والاضطراب في موقع x_{ij} يتناقص وهكذا. وكما في طرائق البحث عن الحل الأمثل في فضاء البحث، طول الخطوة يتناقص تكيفياً. إذا كانت القيمة التي تنتجها هذه العملية تتجاوز حدوده المحددة سلفاً، يمكن إعادة تعيين القيمة إلى قيمة مقبولة. يتم تعيين القيمة اذا تجاوزت حدودها إلى حدود هي:

If $X_i > X_i^{max}$ then $X_{i=} X_i^{max}$; If $X_i < X_i^{min}$ then $X_{i=} X_i^{min}$

[14] بعد إنتاج v_{ij} ضمن الحدود ، يمكن تعيين قيمة الملاءمة (اللياقة) للحل v_{ij} في مسألة الحد الادنىminimization problem بإستخدام المعادلة التالية:

$$fitness_i = \left\{ {}^{1}/_{(1+f_i)} \quad if\ f_i \geq 0 \quad \middle| \quad 1 + abs(f_i) \quad if\ f_i < 0 \right\} \qquad \dots\dots(3)$$

حيث f_i هي دالة الكلفة costing function للحل v_{ij} .لمسألة الحد الأقصى maximization problem، دالة الكلفة يمكن استخدامها مباشرة بوصفها دالة لياقة fitness function . يتم تطبيق عملية اختيار الأفضل greedy selection processبين x_i و v_i ، ثم يتم اختيار أفضل واحد اعتمادا على قيم اللياقة fitness التي تمثل كمية الرحيق من مصادر الغذاء x_i و v_i إذا كان مصدر في v_i هو أعلى من x_i من حيث الربحية ، والنحلة العاملة تخزن في ذاكرتها الموقع الجديد وتنسى القديم. وإلا يتم الاحتفاظ بالموقع السابق في الذاكرة. إذا لم يتم تحسين x_i ، كما يتم زيادة العداد الذي يحسب التجارب بـ 1 ، وإلا تتم إعادة تعيينه إلى 0.

3. حساب قيم احتمال المشاركة في الاختيار الاحتمالي: بعد ان يكمل كل النحل العامل employed Bees عمليات البحث الخاصة بهم، يقوم بمشاركة المعلومات المتعلقة بكمية الرحيق ومواقع مصادر الغذاء مع النحل المشاهد onlooker في المنطقة المخصصة للرقص. هذا النوع من التفاعل المتعدد هو ميزة خوارزمية النحل الاصطناعي ABC حيث ان النحلة المشاهدة تقوم بتقييم معلومات الرحيق المأخوذة من كل النحل العامل وتختار موقع مصدر الغذاء بالاعتماد على الاحتمالية التي تتعلق بكمية الرحيق. هذا الاختيار الاحتمالي يعتمد على قيمة اللياقة للحلول في المجتمع. طريقة الاختيار القائم على اللياقة قد يكون باستخدام عجلة الروليت ، أو نظام تحديد آخر. في خوارزمية ABC الأساسية , طريقة الاختيار باستخدام عجلة الروليت وكما موضح في المعادلة التالية:[14]

$$p_i = \text{fitness}_i \Big/ \sum_{i=1}^{SN} \text{fitness}_i \quad \text{...(4)}$$

في هذه الطريقة للاختيار الاحتمالي ، تزداد كمية الرحيق لمصادر الغذاء (لياقة الحلول)، وبالمقابل عدد نحل المشاهدين تزداد زيارتهم لتلك المصادر أيضا. هذا هو ميزة ردود الفعل الإيجابية من خوارزمية ABC .[14]

4. اختيار موقع مصدر الغذاء من قبل النحل المشاهد على أساس المعلومات المقدمة من النحل العاملات: يتم توليد عدد حقيقي عشوائي ضمن نطاق [0،1] لكل مصدر. إذا كانت القيمة الاحتمالية P_i في المعادلة 4 المتعلقة بذلك المصدر هي أكبر من هذا الرقم العشوائي فالنحل المشاهد يقوم بالتعديل على هذا الموقع لمصدر الغذاء عن طريق استخدام المعادلة 2 كما هو الحال بالنسبة للنحلة العاملة. بعدها يتم تقييم المصدر بتطبيق اختيار الأفضل Greedy Selection فالنحل المشاهد إما يخزن الموقع الجديد مع نسيان القديم أو يبقي الموقع القديم .إذا لم يتم تحسين الحل $_i$ ، يتم زيادة العداد الذي يحسب التجارب بـ 1، وإلا تتم إعادة تعيينه إلى0. وتتكرر هذه العملية حتى يتم توزيع جميع النحل المشاهد على مواقع مصادر الغذاء[14] .

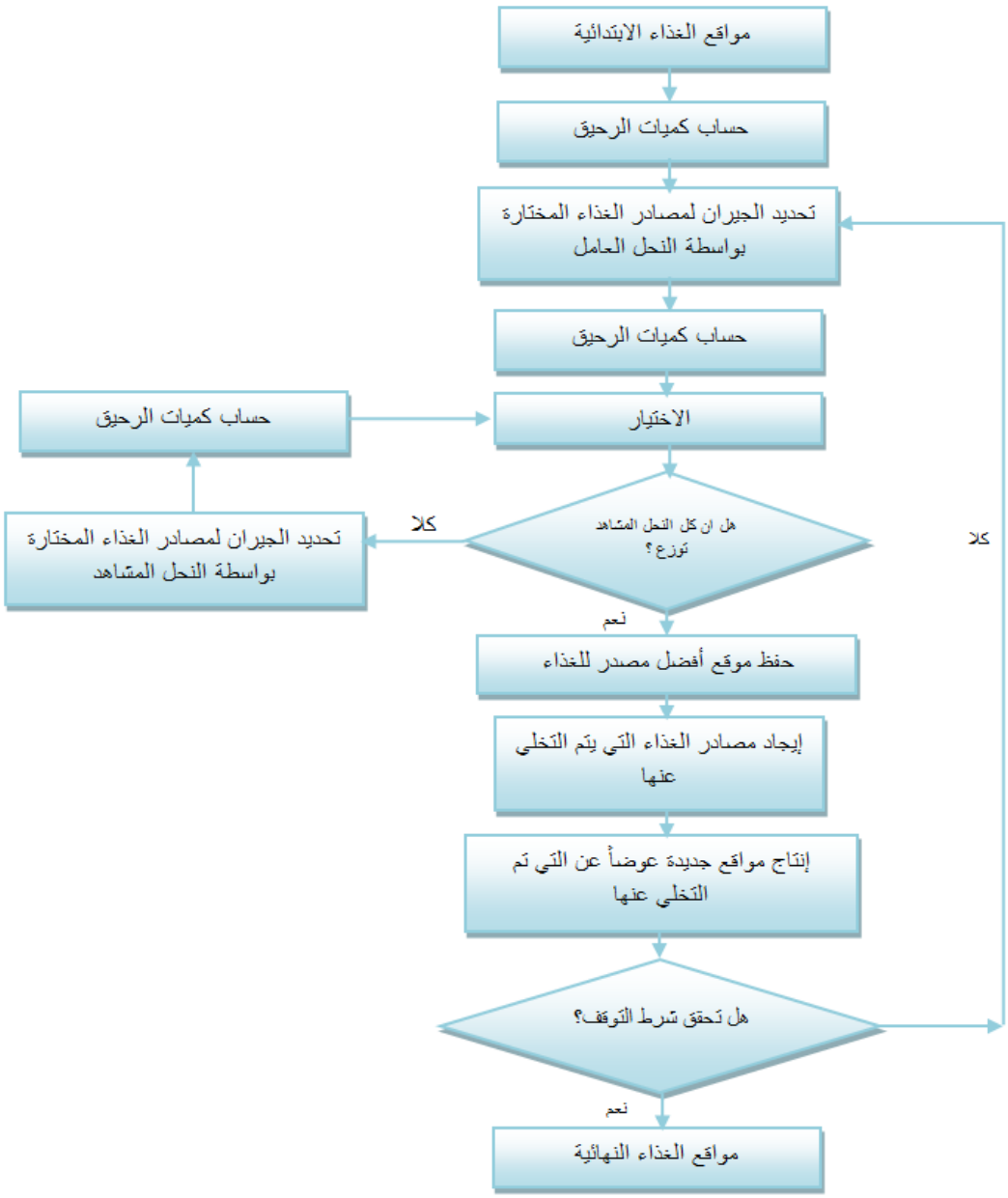
5. معايير التخلي عن المصدر : الحد واستخدام النحل الكشاف: في الدورة الواحدة ،بعد ان تكمل كل من النحل العامل والنحل المشاهد عمليات البحث الخاصة بهم, فأن الخوارزمية تدقق لمعرفة إذا كان هناك أي مصدر مستنفذ يمكن التخلي عنه. من أجل أن تقرر التخلي عن المصدر، يستخدم العداد الذي تم تحديثه خلال البحث. عداد التجارب trail إذا كانت قيمة العداد أكبر من معيار السيطرة على خوارزمية ABC، المعروفة باسم الحد limit ، فلأن المصدر المرتبط بهذا العداد من المفترض ان يكون مستنفذ ويتم التخلي عنه. كما هو معروف عن خوارزمية ABC الأساسية توظف معيار واحد فقط للسيطرة ، وهو ما يسمى الحد، ومعادلته هي:

$$\text{limit} = (SN * MCN) \quad \text{..(5)}$$

حيث MCN هي عدد التكرارات للخوارزمية و SN هو عدد من المصادر الغذائية أو النحل العامل [16].

يتم استبدال مصدر الغذاء الذي تخلت عنه النحلة بمصدر غذاء جديد اكتشف من قبل النحلة الكشافة والذي يمثل آلية ردود الفعل السلبية وخاصية التقلبات في التنظيم الذاتي لخوارزمية ABC. ويتم محاكاة ذلك من خلال

إنتاج موقع جديد بشكل عشوائي واستبداله بالموقع الذي تم التخلي عنه. لو نفرض أن مصدر الغذاء x_i تم التخلي عنه، تقوم النحلة الكشافة عشوائياً باكتشاف مصدر جديد للغذاء ليحل محل x_i ويمكن تعريف هذه العملية كما في المعادلة 1 في خوارزمية ABC الأساسية ، فمن المفترض أن مصدر واحد فقط يمكن أن تستنفذ في كل دورة، وفقط نحلة عاملة واحدة يمكن أن تكون نحلة كشافة. إذا كان هناك أكثر من عداد واحد يتجاوز قيمة الحد"limit " يمكن ان يتم اختيار الأعلى من بين العدادات بشكل برمجي. وتظهر جميع هذه الوحدات والتفاعلات التي بينها في المخطط الانسيابي التالي[14] .



الشكل (5) المخطط الانسيابي لخوارزمية مستعمرة النحلة الاصطناعية

2.8. الخوارزمية المقترحة لمستعمرة النحل الاصطناعي Artificial Bees Colony (ABC)

1. إنتاج مواقع لمصدر الغذاء الأولية: قبل الشروع بإستخدام الخوارزمية يتم توليد حالات اختبار للبرنامج بشكل عشوائي(ضمن مدى معين يحدده المختبِر tester بالاعتماد على البرنامج) ويتم إدخال البرنامج الى عملية تحليل parsing لتحديد المسارات الأساسية للبرنامج, الموضحة في الجدول (4) والجدول (5). تعتبر هذه المسارات الناتجة هي الإدخال للخوارزمية .إذا تم اعتبار فضاء البحث هو البيئة (التي توجد بها الخلية) التي تحتوي على مواقع لمصدر الغذاء ، تبدأ الخوارزمية بأخذ المسارات التي تم الحصول عليها كمجتمع ابتدائي وتعيين القيم الابتدائية المستخدمة في الخوارزمية مثل حجم الخلية SN حيث يتم تقسيم الخلية الى نصفين نحل عامل ونحل مشاهد ويتم تعيين نسبة قليلة جدا للنحل الكشاف , قيمة عدد التجارب trail وهو عبارة عن عداد لعدد المرات التي لا يتغير فيها الحل ويتم إعادة تعيين هذا العداد بالقيمة 0, أيضاً يتم تحديد قيمة العامل D الذي يعبر عن أبعاد فضاء البحث (حسب طبيعة المسألة ويتم تحديد القيمة وغالباً يستخدم العدد 30 كقيمة لهذا العامل) وقد اعتمد في هذا البحث على قيمة التعقيد الدائري للبرنامج Cyclomatic complexity وعدد العقد لأطول مسار في البرنامج حيث يتم حساب القيمتين عند استدعاء البرنامج في عملية الـ parsing , تم جمع القيمتين واعتمادهما كقيمة للعامل D وفي حالة كون البرنامج بسيطاً وقيمة التعقيد الدائري للبرنامج Cyclomatic complexity صغيرة يتم ضرب القيمتين بدل جمعهما حسب المعادلتين التاليتين :

D =cyc+max(no.node) .....(6)

D =cyc*max(no.node) .....(7)

حيث ان cyc تمثل التعقيد الدائري للبرنامج Cyclomatic complexity وأن no.node يمثل عدد العقد لأطول مسار في البرنامج.

يتم تحديد عدد التكرارات للخوارزمية MCN وعدد الشروط المغطاة في كل مسار Condition Covered التي تستخدم في حساب الملائمة Fitness علما أن عدد الشروط المغطاة في كل مسار ConditionCovered تحسب عند استدعاء البرنامج في عملية تحليل البرنامج parsing. بعد التهيئة الابتدائية للقيم ، يتعرض المجتمع (حلول) لتكرار دورات من عمليات البحث من قبِل النحل العامل employed bees، والنحل المشاهد onlooker Bees والنحل الكشفي scout Bees.

2. إرسال النحل العامل employed Bees إلى مواقع مصدر الغذاء: كل نحلة عاملة ترتبط مع حالة اختبار واحدة فقط .وبالتالي ، فإن عدد حالات الاختبار هو مساوٍ لعدد النحل العامل employed bees, النحلة العاملة تُحدث التعديل على الحل بالاعتماد على المعادلة التالية. وتجد الحل الجديد

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

j هو عدد صحيح عشوائي في النطاق [1, D] و k هو مؤشر يتم اختياره عشوائياً ونطاقه [1….SN] ويختلف عن i ، ϕ_{ij} هو رقم عشوائي حقيقي في النطاق [-1,1] .إذا كانت القيمة التي تنتجها هذه العملية تتجاوز حدود محددة سلفاً ، يمكن إعادة تعيين القيمة إلى قيمة مقبولة ضمن تحديد أعلى قيمة مسموح بها واقل قيمة مسموح بها .بعد إنتاج v_{ij} ضمن الحدود، يمكن تعيين قيمة الملاءمة (اللياقة) للحل v_{ij} بإستخدام المعادلة التالية:

$$fitness_i = (1/(1+f_i)) * C\_covered \quad .....(8)$$

حيث fi هي دالة الكلفة costing function للحل v_{ij}, C_covered تمثل عدد الشروط المغطاة من قبِل كل حالة اختبار. يتم تطبيق الاختيار الأفضل greedy selection process بين x_i و v_i ، ثم يتم اختيار الأفضل اعتماداً على قيم اللياقة fitness إذا كانت الملاءمة للمسار في v_i هي أعلى من x_i من حيث الربحية ،فالنحلة

العاملة تستبدل القيم القديمة بالقيم الجديدة . وإلا فيتم الاحتفاظ بالحل السابق .إذا لم يتم تحسين i ، يتم زيادة العداد الذي يحسب التجارب بـ 1 ، وإلا تتم إعادة تعيينه إلى 0.

حساب قيم احتمال المشاركة في الاختيار الاحتمالي: بعد أن يكمل كل النحل العاملemployed bees عمليات البحث الخاصة بهم ويحدّث على الحل، يقوم بإعطاء الناتج الى النحل المشاهد onlooker Bees .هذا النوع من التفاعل المتعدد هو ميزة خوارزمية مستعمرة النحلة الاصطناعية ABC , النحلة المشاهدة تقوم بتقييم المعلومات المأخوذة من كل النحل العامل وتختار موقع الحل بالاعتماد على الاحتمالية. هذا الاختيار الاحتمالي يعتمد على قيمة اللياقة للحلول في المجتمع كما موضح في المعادلة 9:

$$pi = a * fitness(i)/max(fitness) + b \quad \text{.....(9)}$$

حيث a,b تقع قيمتهما في نطاق [0−1] , [21].

3. اختيار موقع الحل من قبل النحل المشاهد على أساس المعلومات المقدمة من النحل العاملات: يتم توليد عدد حقيقي عشوائي ضمن نطاق [0−1] لكل مسار .إذا كانت القيمة الاحتمالية P_i في المعادلة 4 المتعلقة بذلك المسار هي أكبر من هذا الرقم العشوائي فالنحل المشاهد تقوم بالتعديل على هذا المسار عن طريق استخدام المعادلة 2 كما هو الحال بالنسبة للنحلة العاملة. بعدها يتم تقييم الحل الجديد v_i بتطبيق الاختيار الأفضلgreedy selection والنحل المشاهد إما يخزن الحل الجديد مع نسيان القديم أو يبقي الحل القديم. إذا لم يتم تحسين الحل i ، يتم زيادة العداد الذي يحسب التجارب بـ 1، وإلا فيتم إعادة تعيينه إلى 0. وتتكرر هذه العملية حتى يتم توزيع جميع النحل المشاهد على المسارات.

4. الحد واستخدام النحل الكشاف: في الدورة الواحدة ، بعد ان تكمل كل من النحل العامل والنحل المشاهد عمليات البحث الخاصة بهم ، الخوارزمية تدقق لمعرفة إذا كان هناك أي مصدر مستنفذ التي يمكن التخلي عنه. من أجل أن تقرر التخلي عن المصدر ، يستخدم العداد الذي تم تحديثه خلال البحث. عداد التجارب trail إذا كانت قيمة العداد أكبر من معيار السيطرة على خوارزمية ABC، المعروفة باسم الحد limit، المصدر المرتبط بهذا العداد من المفترض أن يكون مستنفذ ويتم التخلي عنه باستخدام المعادلة (1):

يتم خزن المسارات التي لم يتم التغيير عليها وحالات الاختبار وكذلك عدد الشروط المغطاة Conditions Covered لكل حالة اختبار في ملفات كلٌ على حِدّى وتستخدم هذه المعلومات في عملية تعيين الأسبقية لكل حالة اختبار .

9. خوارزمية اختيار الأفضل

تعمل خوارزمية اختيار الأفضلgreedy بمبدأ العنصر الذي يملك أعلى وزن يؤخذ أولاً ويتبعه العنصر الذي يملك ثاني أعلى وزن وهكذا لغاية زيارة كل العناصر وإتمام العملية, ولكن في بعض الأحيان ينشأ حل شبه امثل , خوارزمية اختيار الأفضل greedy تسعى لتقليل الكلفة المتوقعة للوصول الى الهدف. من ميزاتها أنها سهلة وجذابة بسبب أنها غير مكلفة (رخيصة) من ناحية تنفيذها والوقت المستغرق في تنفيذها [8].

10. الخوارزمية الجينية

تعد الخوارزمية الجينية أحد أساليب الذكاء الاصطناعي، إذ برزت أهمية استخدام هذا الأسلوب في حل مسائل معقدة (كبيرة الحجم تمتلك كماً هائلاً من الحلول البديلة) خلال زمن مناسب، والحل الناتج من تطبيق

الخوارزمية الجينية يكون في أغلب الأحيان حلاً قريباً إلى المثالي، ويؤمن هذا الأسلوب عند تطبيقه بحثاً ذكياً بين عدد هائل من الخطط البديلة [18].

1.10. عناصر الخوارزمية الجينية المستخدمة في البحث

الإدخال الى الخوارزمية الجينية هي حالات الاختبار التي تم انتقاؤها بعد استخدام خوارزمية ABC, باستخدام الخوارزمية الجينية يتم الحصول على حالات الاختبار التي تحقق المسارات الأساسية ومُتخذة ترتيب معين الذي يمثل الأسبقية لحالات الاختبار. تتألف الخوارزمية الجينية من عدد من العناصر وفيما يأتي توضيح لها:

1. المجتمع الابتدائي initial Population : يتكون المجتمع من الأفراد , تمثيل كل فرد في المجتمع بكروموسوم [18]. يتمثل المجتمع في هذا البحث بمواقع حالات الاختبار ، كل كروموسوم هو عبارة عن مجموعة من مواقع لحالات الاختبار يتم وضعها في الكروموسوم بشكل عشوائي وقد استخدم حجم المجتمع في هذا البحث 50 فرداً .

2. التقييم Evaluation: يتم في الخوارزمية الجينية تخصيص قيمة الصلاحية Fitness Value لكل فرد سواء كان مولداً في المجتمع الابتدائي أو من خلال الأجيال اللاحقة [18]، وقد استخدمت المعادلة التالية في حساب الملائمة لكل كروموسوم :

Fitness=sum(C_covered)/no.testcases ……(10)

حيث ان C_covered هي تغطية الشروط لحالات الاختبار وno.testcases عدد حالات الاختبار التي تم اختيارها بواسطة خوارزمية ABC

3. التشفير Encoding : يقصد بالتشفير عملية إيجاد تمثيل مناسب للحل ، إذ يعد التشفير عاملاً أساساً لنجاح الخوارزمية الجينية [18]، وقد استخدم تشفير الإعداد الصحيحة Integer Encoding في هذا البحث. حيث أن كل كروموسوم هو عبارة عن مجموعة من المواقع لحالات الاختبار يتم وضعها في الكروموسوم بشكل عشوائي.

4. الانتقاء Selection : يقصد بالانتقاء عملية اختيار الأباء من المجتمع لأجل التقاطع وإنتاج جيل جديد، فبعد اختيار طريقة التشفير المناسبة يصبح القرار التالي الذي يجب اتخاذه هو كيفية إجراء الانتقاء. إن طريقة الانتقاء تحدد الكيفية التي سيتم بها انتقاء الأفراد الذين سيبقون للمرحلة المقبلة [18], وقد استخدم الانتقاء بالاعتماد على عجلة الروليت اي اختيار الفرد الذي يملك أعلى ملاءمة في انتقاء الأب الأول parent1 أما الأب الثاني parent2 فقد تم انتقاؤه باختيار الفرد الذي يملك اقل ملاءمة.

5. التقاطع أو التزاوج Crossover : بعد أن تمت عملية اختيار مقطعين من مقاطع الآباء بالجيل الحالي باستخدام إحدى طرق الاختيار، تبدأ عملية التقاطع ، وهي عملية تكوين مقطع جديد ليمثل أحد مقاطع الجيل اللاحق[18]. وقد استخدم التقاطع ذو النقطة الواحدة one point crossover في البحث ، إذ يتم اختيار موقع القطع بصورة عشوائية وإنتاج الكروموسومين الجديدين من الأبوين المختارتين وكانت نسبة التقاطع crossover rate في المجتمع هي ما يقارب 0.8 .

11. اختيار(انتقاء) حالات الاختبار وتعيين أسبقيتها

تم استلهام الفكرة من التقنيات المستخدمة في الاختبار التراجعي regression testing التي تستخدم لاختيار حالات الاختبار التي تم ذكرها أنفاً. عمليتي الاختبار والاختبار التراجعي من العمليات المكلفة(مصادر .

جهد . كلفة مادية – وقت) في تطوير البرمجيات فهناك باحثون يقولون أن نسبة عملية الاختبار تتراوح بين 40 – 70% من وقت وكلفة عملية تطوير البرمجيات ولمحاولة تقليل هذه الكلفة اتجهت الأنظار نحو حالات الاختبار فاتجه الباحثون للتقليل من حالات الاختبار ومحاولة جدولتها ووضعها حسب أسبقيات معينة بالاعتماد على عدة معايير[11] . حالات الاختبار سواء كانت لعملية الاختبار أو عملية الاختبار التراجعي يتم اختيارها بالاعتماد على خوارزمية مستعمرة النحلة الاصطناعية ABC , يبدأ العمل بتدريب الخوارزمية بعد أن تم تهيئتها بالقيم الابتدائية اللازمة (حجم المستعمرة colony size, فضاء البحث search space ,عدد التكرارات epoch, التعقيد الدائري Cyclomatic complexity, عدد الشروط المغطاة لكل حالة اختبار Condition covered حيث يكون الإدخال هو المسارات الأساسية للبرنامج (التي تم استخلاصها بالاعتماد على مخطط سيطرة التدفق والذي يتم تمثيل بياناته بملف بصيغة XML).

الإدخال : البرنامج (الشفرة المصدرية) المكتوب بلغة C++.

إنشاء XML

- تحويل البرنامج الى صيغة XML وذلك لعدم القدرة على التعامل مع مخطط CFG بشكل مباشر فيتم تحويل البرنامج الى صيغة الXML لتسهيل التعامل معه وكما موضح بالشكل (7) والشكل(8).

توليد المسارات :

- يتم توليد مسار معين لكل حالة اختبار بالاعتماد على مخطط تدفق السيطرة CFG الموضح في الشكل(9) والشكل (10).

استخدام الخوارزمية :

- باستخدام خوارزمية ABC المقترحة يتم الحصول على عدد من حالات الاختبار التي تحقق المسارات الأساسية.
- باستخدام الخوارزمية الجينية وخوارزمية اختيار الأفضل يتم وضع حالات الاختبار حسب أسبقية معينة تسهل عملية التنفيذ وتقلل من الجهد والكلفة.

وقد استخدمت خمسة برامج مكتوبة بلغة C++ وهي متفاوتة في عدد الأسطر البرمجية .وكما موضح في الجدول(1) :

جدول (1) البرامج المستخدمة ووصفها

البرامج	عدد الأسطر البرمجية	التعقيد الدائري (Cyclomatic complexity)	وصف البرنامج
Pass.cpp	33	10	بإعطاء معدل الطالب البرنامج يقوم بإرجاع تقدير ذلك الطالب
avarge.cpp	39	13	بإعطاء درجات الطالب البرنامج يعطي معدل وتقدير الطالب
Ifelse.cpp	38	6	البرنامج يصف قائمة لمجموعة من

			الأكلات لمطعم ما بإستخدام جمل إذا الشرطية (if) ويقوم البرنامج بإرجاع وصف معين عن الوجبة المختارة.
Calculator.cpp	62	2	البرنامج يقوم بإنجاز المهام الأساسية للحاسبة
Scientificcal.cpp	160	10	البرنامج يقوم بإنجاز المهام الأساسية ومهام ثانوية أخرى للحاسبة (الحاسبة العلمية)

في هذا البحث يتم الحد من تكلفة عمليتي الاختبار والاختبار التراجعي وذلك بإنتقاء حالات الاختبار وتعيين الأسبقية لها فبرنامج avarge.cpp هو النسخة المعدّلة عن برنامج pass.cpp وأيضا Scientificcal.cpp هو النسخة المعدّلة عن برنامج Calculator.cpp , ففي عملية الاختبار للبرامج Pass.cpp, Calculator.cpp,Ifelse.cpp تم تقليل حالات الاختبار لها باستخدام خوارزمية مستعمرة النحلة الاصطناعية ABC المقترحة وكذلك للاختبار التراجعي أُستخدم البرنامجين avarge.cpp وScientificcal.cpp حيث تم تقليل حالات الاختبار لهما باستخدام خوارزمية مستعمرة النحلة الاصطناعية ABC المقترحة, وقد تم في بادئ الأمر استخدام خوارزمية ABC القياسية إلا أنها لم تخدم العمل. فتم اللجوء الى إجراء تغيير في المعادلات رغبة لتحقيق نتائج أفضل, يتم تدريب الخوارزمية للحصول على مجموعة من حالات الاختبار ومن ثم تبدأ عملية الاختبار للخوارزمية ينتج عنها مجموعة من حالات الاختبار التي تحقق المسارات الأساسية والجدول (2) يوضح النتائج:

جدول (2) النتائج التي تم الحصول عليها باستخدام خوارزمية ABC

البرامج	الوقت المستغرق للتدريب		الوقت المستغرق للفحص بالثواني	عدد التكرارات للتدريب	عدد مرات التنفيذ لحين الحصول على النتيجة المطلوبة	عدد النحل المستخدمة في التدريب	عدد حالات الاختبار المستخدمة خلال التدريب (نصف عدد النحل)	أبعاد فضاء البحث
	بالثواني	بالدقائق						
Passdevelop.xml	42.6757	0.7113	0.8129	6	3	200	100	18
Avarge.xml	68.162175	1.1360	0.5519	7	7	200	100	24
Ifelse.xml	52.108690	0.8685	0.4673	8	2	50	25	14
Calculator.xml	100.133207	1.6689	0.6717	6	5	200	100	10
Scientificcal.xml	297.191128	4.9532	7.9375	6	10	310	155	24

بعد الانتهاء من عملية الاختيار Selection تبدأ عملية تعيين الأسبقية prioritization لحالات الاختبار فإما باستخدام خوارزمية اختيار الأفضلGreedy Algorithm أو باستخدام الخوارزمية الجينية Genetic Algorithm. والجدول(3) يوضح النتائج التي تم التوصل إليها.

جدول (3) النتائج التي تم الحصول عليها باستخدام خوارزمية ABC والخوارزمية الجينية وخوارزمية اختيار الأفضل

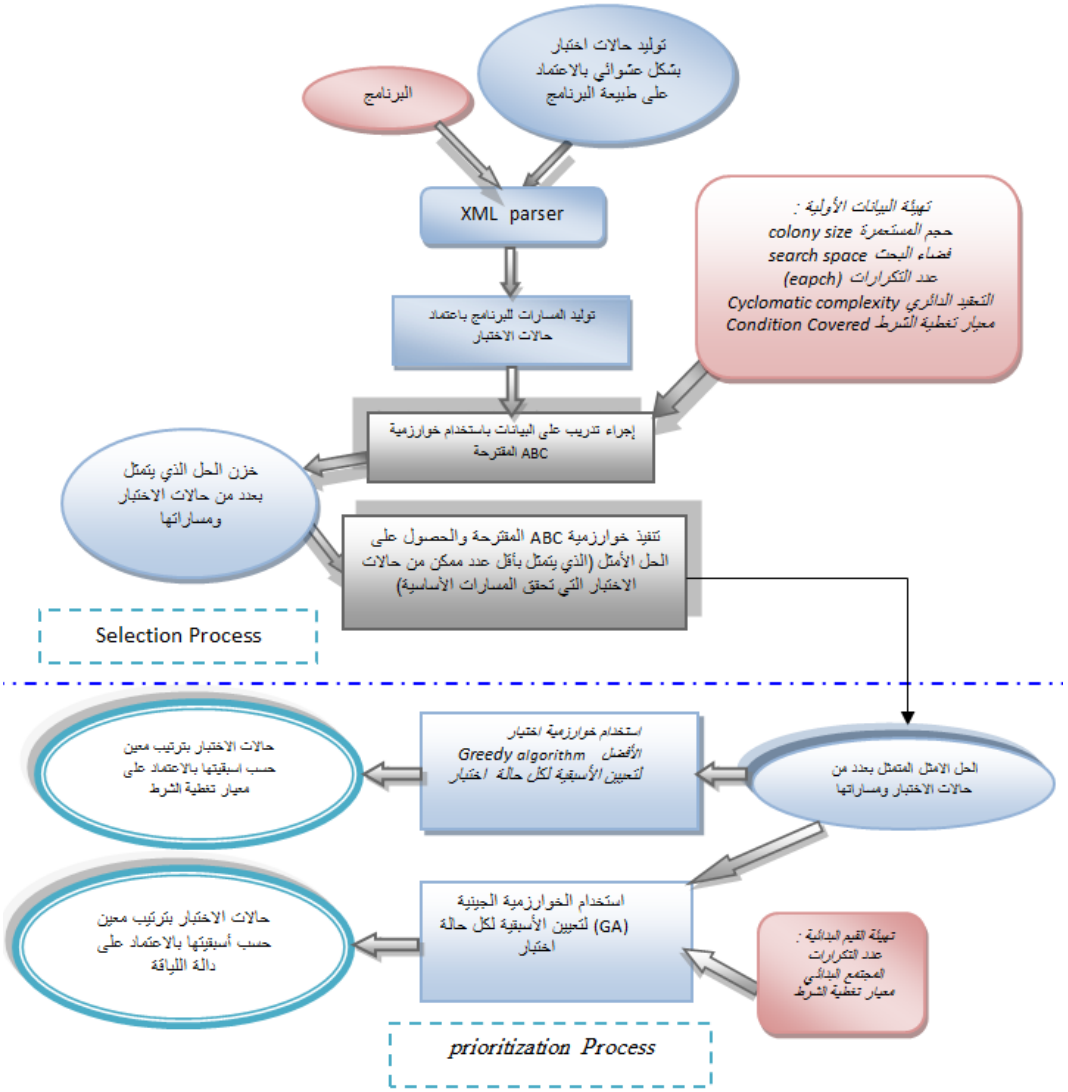
البرامج	عدد حالات الاختبار قبل التدريب	عدد حالات الاختبار بعد التدريب	عدد حالات الاختبار بعد الفحص	عدد حالات الاختبار الناتجة بعد عمل الأسبقية باستخدام طريقة اختيار الأفضل greedy algorithm	عدد حالات الاختبار الناتجة بعد عمل الأسبقية باستخدام الخوارزمية الجينية	عدد التكرارات للخوارزمية الجينية
Pass.xml	100	26	22	19	22	50
Avarge.xml	100	39	34	33	31	100
Ifelse.xml	25	16	14	6	6	10
Calculator.xml	100	27	19	12	12	10
Scientificcal.xml	155	63	56	26	26	80

النتائج تظهر الحصول على جميع المسارات وضمن وقت قليل. ولتقيم النتائج يتم حساب تغطية الشرط عن طريق المعادلة التالية:

$$Condition\ Coverage = \frac{number\ of\ Excuted\ Condition}{number\ of\ total\ conditions} * 100\% \quad \text{.....(11)}$$

حيث أن number of executed Condition هي عدد الشروط التي يتم تغطيتها في مجموعة الاختبار الناتجة, وnumber of total Conditions هي مجموع الشرط الموجودة في البرنامج[20]. بما ان مجموعة الاختبار الناتجة بعد عملية الاختيار لحالات الاختبار تحقق جميع المسارات الأساسية, فهذا يعني انه تم المرور على جميع الشروط في البرنامج دون استثناء وتكون نسبة التغطية للبرنامج هي 100% وبذلك يتم تقييم النتائج.

ولتوضيح العمل بشكل أفضل تم تمثيله بمخطط يوضح العمل بالشكل(6):



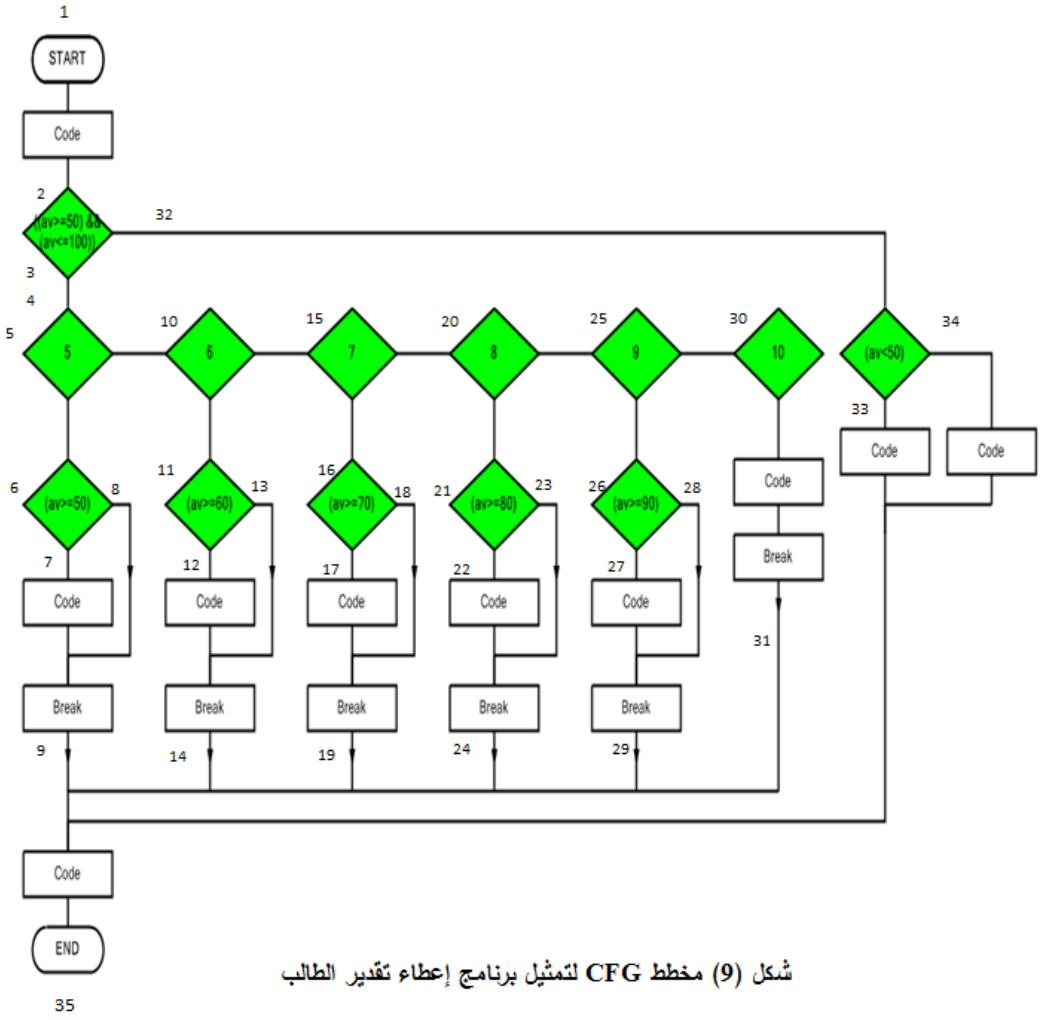
شكل (6) مخطط توضيحي للعمل

```
<?xml version="1.0" encoding="GB2312"?>
- <Function EndPos="501" BegPos="54" Text="void main">
    <Code EndPos="72" BegPos="56" Text="int av; cin>>av;"/>
  - <IF EndPos="486" BegPos="75" Text="((av>=50) && (av<=100))">
    - <YES EndPos="429" BegPos="106" Text="Yes">
      - <SWITCH EndPos="429" BegPos="106" Text="(av/10)">
        - <Case_Item EndPos="171" BegPos="129" Text="5">
          - <IF EndPos="161" BegPos="136" Text="(av>=50)">
            - <YES EndPos="161" BegPos="149" Text="Yes">
                <Code EndPos="161" BegPos="149" Text="cout<<"pass";"/>
              </YES>
              <NO EndPos="-1" BegPos="-1" Text="No"/>
            </IF>
            <Break EndPos="171" BegPos="166" Text="break;"/>
          </Case_Item>
        - <Case_Item EndPos="220" BegPos="176" Text="6">
          - <IF EndPos="210" BegPos="183" Text="(av>=60)">
            - <YES EndPos="210" BegPos="196" Text="Yes">
                <Code EndPos="210" BegPos="196" Text="cout<<"middle";"/>
              </YES>
              <NO EndPos="-1" BegPos="-1" Text="No"/>
            </IF>
            <Break EndPos="220" BegPos="215" Text="break;"/>
          </Case_Item>
        - <Case_Item EndPos="270" BegPos="227" Text="7">
          - <IF EndPos="260" BegPos="234" Text="(av>=70)">
            - <YES EndPos="260" BegPos="248" Text="Yes">
                <Code EndPos="260" BegPos="248" Text="cout<<"good";"/>
              </YES>
              <NO EndPos="-1" BegPos="-1" Text="No"/>
            </IF>
            <Break EndPos="270" BegPos="265" Text="break;"/>
          </Case_Item>
        - <Case_Item EndPos="324" BegPos="277" Text="8">
          - <IF EndPos="314" BegPos="284" Text="(av>=80)">
            - <YES EndPos="314" BegPos="298" Text="Yes">
                <Code EndPos="314" BegPos="298" Text="cout<<"verygood";"/>
              </YES>
```


شكل (7) مقطع من برنامج إعطاء تقدير الطالب بصيغة XML

```
<?xml version="1.0" encoding="GB2312"?>
- <Function EndPos="711" BegPos="54" Text="void main">
    <Code EndPos="140" BegPos="56" Text="int av,mark; int su=0,mcount=0; cout<<'\n'<<"inter grade of student:"; cin>>mark;"/>
  - <WHILE EndPos="218" BegPos="143" Text="((mark!=-1) && (mark<=100) )">
    - <Circle EndPos="218" BegPos="179" Text="Circle">
        <Code EndPos="215" BegPos="180" Text="su=su+mark; cout<<'\n'; cin>>mark;"/>
      </Circle>
    </WHILE>
    <Code EndPos="264" BegPos="221" Text="av=su/4; cout<<'\n'<<"the avarage is:"<<av;"/>
  - <IF EndPos="696" BegPos="267" Text="((av>=50) && (av<=100))">
    - <YES EndPos="629" BegPos="298" Text="Yes">
      - <SWITCH EndPos="629" BegPos="298" Text="(av/10)">
        - <Case_Item EndPos="363" BegPos="321" Text="5">
          - <IF EndPos="353" BegPos="328" Text="(av>=50)">
            - <YES EndPos="353" BegPos="341" Text="Yes">
                <Code EndPos="353" BegPos="341" Text="cout<<"pass";"/>
              </YES>
              <NO EndPos="-1" BegPos="-1" Text="No"/>
            </IF>
            <Break EndPos="363" BegPos="358" Text="break;"/>
          </Case_Item>
        - <Case_Item EndPos="412" BegPos="368" Text="6">
          - <IF EndPos="402" BegPos="375" Text="(av>=60)">
            - <YES EndPos="402" BegPos="388" Text="Yes">
                <Code EndPos="402" BegPos="388" Text="cout<<"middle";"/>
              </YES>
              <NO EndPos="-1" BegPos="-1" Text="No"/>
            </IF>
            <Break EndPos="412" BegPos="407" Text="break;"/>
          </Case_Item>
        - <Case_Item EndPos="462" BegPos="419" Text="7">
          - <IF EndPos="452" BegPos="426" Text="(av>=70)">
            - <YES EndPos="452" BegPos="440" Text="Yes">
                <Code EndPos="452" BegPos="440" Text="cout<<"good";"/>
              </YES>
              <NO EndPos="-1" BegPos="-1" Text="No"/>
            </IF>
```

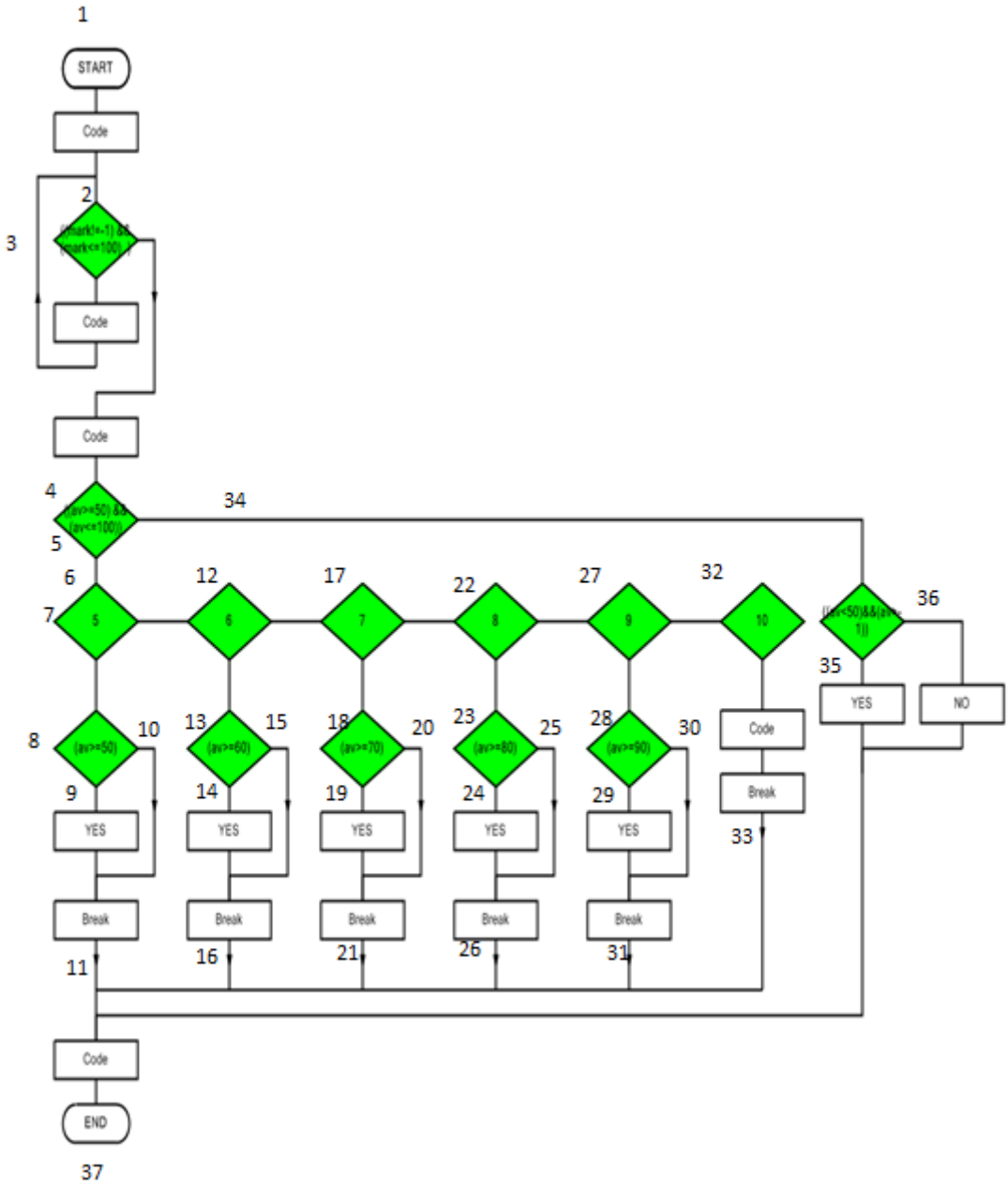

شكل (8) مقطع من برنامج حساب معدل الطالب بصيغة XML



شكل (9) مخطط CFG لتمثيل برنامج إعطاء تقدير الطالب

جدول (4) المسارات الأساسية لبرنامج إعطاء تقدير الطالب

البرنامج	المسارات الأساسية للبرنامج								
Pass.cpp	1	2	32	33	35				
	1	2	3	4	25	26	27	29	35
	1	2	3	4	10	11	12	14	35
	1	2	3	4	5	6	7	9	35
	1	2	3	4	15	16	17	19	35
	1	2	3	4	20	21	22	24	35
	1	2	32	34	35				
	1	2	3	4	30	31	35		



شكل (10) مخطط CFG لبرنامج حساب المعدل

جدول (5) المسارات الأساسية لبرنامج حساب معدل الطالب

البرنامج	المسارات الأساسية للبرنامج
Avarge.cpp	1 2 3 4 5 6 7 8 9 11 37
	1 2 3 4 5 6 12 13 14 16 37
	1 2 3 4 5 6 17 18 19 21 37
	1 2 3 4 5 6 22 23 24 26 37
	1 2 3 4 5 6 27 28 29 31 37
	1 2 3 4 5 6 32 33 37
	1 2 3 4 34 35 37
	1 2 3 4 34 36 37

12. الاستنتاجات

باستخدام خوارزمية مستعمرة النحلة الاصطناعية التي تم التطوير عليها وذلك باقتراح معادلتين لحساب فضاء البحث بالاعتماد على التعقيد الدائري وعدد العقد لأطول مسار للبرنامج كخطوة تطويرية أولى وشمل التطوير أيضا معادلة اللياقة حيث تم إدخال معيار عدد الشروط على معادلة اللياقة الأصلية واسهم ذلك في تحسين وتسريع من أداء الخوارزمية كما وتم اعتماد معادلة مختلفة عن المعادلة المستخدمة في الخوارزمية القياسية لحساب احتمالية اختيار النحل المشاهد , المعادلة المستخدمة في حساب الاحتمالية التي يوجد بها عاملان وهما a,b اللذان يقعان في نطاق [1,0]. هذه الخطوات التطويرية أسهمت إسهاما ملفتا في التسريع من الوصول الى الحل وكانت النتائج ممتازة حيث تظهر النتائج في مرحلة الاختبار للخوارزمية انه يتم الحصول على الحل المطلوب في ثواني أو أجزاء الثانية. أما تعيين الأسبقية لحالات الاختبار فقد أثبتت الأبحاث[7][8] أن إعطاء حالات الاختبار حسب ترتيب معين يسهم كثيرا في تقليل الوقت المستغرق لاختبار البرمجيات. وفي هذا البحث لتعيين الأسبقية لحالات الاختبار التي تم اختيارها بواسطة خوارزمية الـ ABC فقد تم استخدم خوارزمية اختيار الأفضل والخوارزمية الجينية, خوارزمية اختيار الأفضل تقوم بترتيب حالات الاختبار حسب أعلى قيمة في تغطية الشرط وصولاً الى اقل قيمة, أما الخوارزمية الجينية فقد كانت دالة اللياقة تعتمد على معيار تغطية الشرط مقسوم على عدد حالات الاختبار واستخدم التقاطع crossover ذو النقطة الواحدة بنسبة 0.8 وقد تم الحصول على أفراد تحتوي على كل حالات الاختبار موضوعة في ترتيب معين.

13. الأعمال المستقبلية

تم تطبيق إحدى تقنيات ذكاء السرب مع تقنيات الاختبار التراجعي وتعتبر هذه انطلاقة الى استخدام تقنيات ذكاء السرب مع تقنيات الاختبار التراجعي, وقد تم الاقتراح التالي:

- لقد تم استخدام خوارزمية مستعمرة النحلة الاصطناعية ABC التي تعتبر احد خوارزميات ذكاء السرب, ويحتوي ذكاء السرب على خوارزميات أخرى. من المقترح أن يتم استخدام تلك الخوارزميات في توليد واختيار حالات الاختبار ومقارنة النتائج فيما بينها.
- لتوجه الى اختيار حالات الاختبار, تقليل وتعيين أسبقية حالات الاختبار المستخدمة في الاختبار التراجعي لتطبيقات الويب باستخدام التقنيات الذكائية وذلك رغبة في تقليل الوقت والجهد المستهلكين في الاختبار خصوصاً وان تطبيقات الويب في تطور متسارع.

المصادر

[1] Reddy,N & Kumar,S., (2011),An Effective Approach To Regression Test Optimization Technique, India, , Indian Journal of Computer Science and Engineering (IJCSE).

[2] Hansson, E. (2004) ,Coverage Based Regression Test Selection– Selecting Regression Tests for a JVM, Master's Thesis in Computer Science at the School of Computer Science and Engineering .

[3] Yoo,S. & Harman,M. (2010), Regression testing minimization , selection and prioritization :a survey, published by John Wiley & Sons,

[4] Pressman.R.S.,(2001), A Practitioner's Approach Fifth Edition, published by McGraw-Hill Higher Education A Division of The McGraw-Hill Companies, Americas, New York .

[5] Acharya S.,& Pandya V.(2009), Bridge between Black Box and White Box – Gray Box Testing Technique, International Journal of Electronics and Computer Science Engineering www.ijecse.org.

[6] NAIK K.,(2008),Software Testing And Quality Assurance Theory And Practice, Published by John Wiley & Sons, Inc., Hoboken, New Jersey Published simultaneously in Canada.

[7] Suri B., Mangal I. & Srivastava V.,(2010) , Regression Test Suite Reduction using an Hybrid Technique Based on BCO And Genetic Algorithm, Issue of International Journal of Computer Science & Informatics (IJCSI)

[8] Li Z. , Harman M., and Hierons.R.M,(2007),Search Algorithms for Regression Test Case Prioritization , IEEE transactions on software engineering.

[9] Malhotra R. , Kaur A. and Singh Y.,(2010), A Regression Test Selection and Prioritization Technique, India, Journal of Information Processing Systems.

[10] Singh T.& Sandhu M.K.,(2012) , An Approach in the Software Testing Environment using Artificial Bee Colony (ABC) Optimization , International Journal of Computer Applications .

[11] Roongruangsuwan S. & Daengdej J.,(2005-2010),Test Case Prioritization Techniques, journal of theoretical and applied information technology 2005-2010, www.jatit.org .

[12] www.elnadaBee.com

[13] Kaur A., Goyal S.,(2011), A Survey on the Applications of Bee Colony Optimization Techniques , India , International Journal on Computer Science and Engineering (IJCSE).

[14] Akay B, Karaboga D.,(2010), A modified Artificial Bee Colony algorithm for real-parameter optimization, Turkey, Elsevier, journal homepage: (www.elsevier.com\locate\ins).

[15] Akbari R., Mohammadi A. and Ziarati K.,(2009) ,A novel bee swarm optimization algorithm for numerical function optimization , Shiraz, Iran, Elsevier www.elsevier.com.

[16] KARABOGA D.,(2005), An Idea Based On Honey Bee Swarm For Numerical Optimization (TECHNICAL REPORT-TR06), Turkey.

[17] Al-Obaidy. S .I.,(2004) , Designing A Genetic Fuzzy Model for the Evaluating of Hospital Efficiency, Master's Thesis in Computer Science and Mathematical University of Mosul .

[18] Kharruffa S. H. ,(2005) , Using Genetic Hybrid Algorithm In The Classification Of Satellite Images, Master's Thesis in Computer Science and Mathematics Sciences University of Mosul .

[19] Baykasoùlu A., Özbakır L. ,and Tapkan P.,(2007), Artificial Bee Colony Algorithm and Its Application to generalized Assignment Problem, Itech Education and Publishing, Vienna, Austria, www.i-techonline.com

[20] British Computer Society Specialist Interest Group in Software Testing (BCS SIGIST) ,(2001), Standard for Software Component Testing, http://www.testingstandards.co.uk/ .

[21] Tankasala G. R., (2012), Artificial Bee Colony Optimisation for Economic Load Dispatch of a Modern Power system, International Journal of Scientific & Engineering Research.