Software Effort Estimation Using Multi Expression Programming

Najla Akram Al-Saati Taghreed Riyadh Alreffaee
dr.najla_alsaati@uomosul.edu.iq *taghreed_reyad@uomosul.edu.iq*
College of Computers Sciences and Mathematics / University of Mosul
Received on: 19/9/2013 **Accepted on: 12/2/2014**

ABSTRACT

The process of finding a function that can estimate the effort of software systems is considered to be the most important and most complex process facing systems developers in the field of software engineering. The accuracy of estimating software effort forms an essential part of the software development phases. A lot of experts applied different ways to find solutions to this issue, such as the COCOMO and other methods. Recently, many questions have been put forward about the possibility of using Artificial Intelligence to solve such problems, different scientists made several studies about the use of techniques such as Genetic Algorithms and Artificial Neural Networks to solve estimation problems. This work utilizes one of the Linear Genetic Programming methods (Multi Expression programming) which apply the principle of competition between equations encrypted within the chromosomes to find the best formula for resolving the issue of software effort estimation. As for to the test data, benchmark known datasets are employed taken from previous projects, the results are evaluated by comparing them with the results of Genetic Programming (GP) using different fitness functions. The gained results indicate the surpassing of the employed method in finding more efficient functions for estimating about 7 datasets each consisting of many projects.

Keywords: Effort Estimation, Multi Expression Programming, Genetic Programming.

تخمين الجهد البرمجي باستخدام البرمجة المتعددة التعابير

نجلاء اكرم الساعاتي تغريد رياض جارالله الرفاعي
كلية علوم الحاسوب والرياضيات / جامعة الموصل

تاريخ استلام البحث : 2013\9\19 **تاريخ قبول البحث : 2014\2\12**

الخلاصة

تعد عملية ايجاد دالة لتخمين جهد الانظمة البرمجية من اهم واعقد العمليات التي تواجه مطوري الانظمة في حقل هندسة البرمجيات ، حيث ان الدقة في تخمين الجهد تشكل جزءا اساسيا من مراحل تطوير البرمجيات. لقد قام العديد من الخبراء بتطبيق مختلف الطرائق لايجاد حلول لهذه المسالة ومنها طريقة الكوكومو وغيرها من الطرائق. وتم في الاونة الاخيرة طرح العديد من الاسئلة حول امكانية استخدام الذكاء الاصطناعي لحل هذه المشكلة ، حيث قدم مختلف العلماء دراسات عديدة حول استخدام تقنيات مثل الخوارزمية الجينية والشبكات العصبية الاصطناعية لحل مسائل التخمين. وقد تم في هذا البحث استخدام احدى طرائق البرمجة الجينية الخطية وهي طريقة البرمجة المتعددة التعابير (Multi Expression Programming MEP) والتي تتضمن تطبيق مبدأ التنافس بين معادلات مشفرة داخل الكروموسومات لايجاد المعادلة الافضل في حل مسالة تخمين جهد البرمجيات. اما فيما يختص بالبيانات فقد تم استخدام مجاميع بيانات قياسية ومعروفة لمشاريع سابقة وجرى تقييم النتائج المستحصلة من خلال مقارنتها مع نتائج طريقة البرمجة الجينية (Genetic Programming GP) وباستخدام طرق مختلفة لدالة اللياقة. وقد اظهرت النتائج تفوق الطريقة المستخدمة في ايجاد دوال اكثر كفاءة في تقييم الجهد لما يقارب 7 مجاميع من البيانات المتضمنة لعدة مشاريع.

الكلمات المفتاحية: *تخمين الجهد ، البرمجة المتعددة التعابير ، البرمجة الجينية.*

1. المقدمة

مع التطور السريع الذي شهدته البرمجيات في عصرنا الحديث أصبحت البرمجيات تشكل اساسا يعتمد عليه في معظم الصناعات والمصانع وذلك لما توفره من دقة وسرعة في انجاز العمل كما إن استخدام هذه البرامج اصبح يشكل مصدر تطور وربح للشركات من حيث اختصار الوقت والكلف المبذولة.

غالبا ما يشكل التخمين الدقيق للحجم والكلفة والجهد والجدول الزمني للبرمجيات التحدي الاكبر الذي يواجه مطوري البرمجيات في الوقت الحاضر ، حيث ان تأثيره الأساسي على ادارة تطوير البرمجيات يكمن في التأثير المباشر لكل من تحت التخمين "underestimates" وفوق التخمين "overestimates" في احداث الضرر في الشركات البرمجية. وقد تم اقتراح عدة نماذج من قبل باحثين مختلفين لتخمين الجهد وهناك ايضا بعض الدراسات التي اجريت لتخمين الجهد البرمجي في المراحل المبكرة لتوضيح أهميته[1][5] .

لذا يقع على منظمات البرمجيات معرفة الكيفية التي تمكنها من تطوير مشاريعها وكذلك كمية الجهد المطلوبة لعملية التطوير [14]، حيث ان تخمين الجهد هو فعالية حرجة ومهمة لعملية تخطيط ومراقبة تطوير مشاريع البرمجيات وكذلك لتسليم المنتج بالوقت والميزانية المحددتين [9].

تعتبر آلية تخمين الجهد البشري من اهم تحديات هندسة البرمجيات التي يعاني منها مدراء المشاريع والتي تؤثر بدورها على كلفة المشروع ، ونتيجة لذلك تبرز الكفاءة للادارة التي تستخدم تخمين الجهد لتقييم المشاريع وادارة عمليات التطوير بشكل اوضح واكبر. وكذلك فان دقة تخمين جهد البرمجيات يعد واحداً من اهم التحديات التي تواجه مطوري البرمجيات، حيث تعتبر الطرق التقليدية مثل طريقة الكوكومو (COCOMO) محدودة بعدم قدرتها على ادارة الحيرة والانطباعات التي تحيط مشاريع البرمجيات في مراحل مبكرة من عملية التطوير لذلك لجأ الباحثون في الفترة الاخيرة الى استخدام الطرائق الحاسوبية الذكية مثل الشبكات الاصطناعية والخوارزميات الجينية وكذلك البرمجة الجينية والمنطق المضبب.

يتناول هذا البحث دراسة حول امكانية استخدام احد الطرائق الخطية المطورة من البرمجة الجينية وهي طريقة البرمجة المتعددة التعابير ((Multi Expression Programming (MEP) [23] والتي تتعامل مع الكروموسوم بشكل خطي وليس كشجرة كما هو متعارف عليه في البرمجة الجينية. وسيتم استخدامها في ايجاد المعادلة الانسب لحساب الجهد بدقة، وستقارن النتائج مع نتائج لباحثين استخدموا البرمجة الجينية (Genetic Programming(GP)) في تخمين الجهد وباستخدام عدد من مجاميع البيانات القياسية.

2. الدراسات السابقة

تم تقديم العديد من البحوث في مجال تخمين جهد البرمجيات وذلك باستخدام طرق مختلفة ومتنوعة وقد اختلفت النتائج باختلاف الطرق ، وفيما يلي بعض من هذه الاعمال:

في عام (2001) قام الباحث (Dolado) باستخدام طريقة البرمجة الجينية (GP) لاستكشاف دالة لحساب الكلفة وقد قورنت النتائج مع بيانات سابقة [8]. وفي العام نفسه قام (Burgess & Lefley) بمحاولة لاستخدام البرمجة الجينية ومقارنة النتائج مع طرائق سابقة باستخدام بيانات قياسية معروفة [6].

وفي العام (2003) استخدم الباحثان (Lefley, Shepperd) طريقة البرمجة الجينية لتحسين عملية تخمين جهد البرمجيات بالاعتماد على مجاميع عامة للبيانات [19]. اما في عام (2004) فقد اقترح الباحثون (Ohsugi,

.et al) طريقة للتخمين بالاعتماد على مايسمى بالـ (Collaborative Filtering) واسترجاع البيانات المفقودة كاحد تقنيات التخمين باستخدام البيانات الناقصة او المعابة (Defective Data) [22] .

طبقت فكرة الخوارزميات الجينية (Genetic Algorithms) في العام (2006) من قبل الباحثان (Huang, Chiu) لقياس الجهد للبرمجيات من خلال الاوزان غير المتكافئة والاوزان الخطية واللاخطية [10]. اما في العام (2008) فقد ظهرت فكرة (Bayesian Network Models) من قبل (Mendes, Mosley) كدراسة مقارنة لتخمين كلف الويب [20].

واستخدم الباحثان (Tsakonas, Dounias) في عام (2009) البرمجة الجينية لاعطاء تعبير رياضي ذي دقة عالية وليساعد في ايجاد الجهد المخمن باعطاء علاقة بين مايسمى بخواص المشروع والعمل المطلوب وقد تم استخدام بيانات الـ (COCOMONASA dataset) و(COC81 dataset) [29].

وفي عام (2010) قام الباحثان (Patil & Yogi) بجمع معلومات من 10 فرق بمن فيهم متدربو المشروع عن الصعوبات التي واجهتهم في مرحلة تطوير المشاريع ، واخذت جميع عوامل المخاطر بنظر الاعتبار حيث وجد ان ادارة المخاطر بصورة فعالة مع دقة التخمين تساعد المدراء على تفادي الـ (Deadline) للمشروع [25].
واستخدم (Sheta ,Al-Afeef) في العام ذاته البرمجة الجينية لتطوير نموذج رياضي لتخمين الجهد باستخدام متغيرين (LOC وMethodology) لتطوير علاقة بينهما [27].

في عام (2012) قام الباحثون (Ziauddin, Tipu, and Zia) بايجاد نموذج لتخمين الجهد لمشاريع الـ (Agile Software Projects) واستخدموا الطرق التقليدية وبيانات تجريبية مؤلفة من 21 مشروع [31]. وقام الباحثون (Singh and Misra) في نفس العام باستخدام طرق لتخمين جهد البرمجيات مبنية على اسس تقنيات الـ (Soft Computing) والتي وفرت الحل لهذه المشكلة ، وقد تم استخدام الشبكات العصبية الاصطناعية بوجود بيانات ناسا (NASA Project Data) التي تألفت من 85 مشروع [28]. وكذلك قامت الباحثة اسراء زهير في عام (2012) بتقديم تحليل لاداء الشبكات العصبية ومقارنتها مع طريقة الـ (COCOMO) وبينت نتائج التخمين تفوق الشبكات العصبية على الطرق التقليدية [26].

وتم مؤخرا في عام (2013) اقتراح استخدام طريقة الـ (Function Point FP) مع الـ (Data flow Diagram) من قبل الباحث (Arnuphaptrairong) لحل مسألة الحصول على معلومات التخمين في مراحل مبكرة من تطوير البرمجيات ، حيث اعتمدت معظم نماذج التخمين على المعلومات التي تستخلص في مراحل متاخرة من تطوير البرمجيات [2].

3. تخمين الجهد (Effort Estimation)

يعرف الجهد في مجال هندسة البرمجيات على انه الوقت الكلي الذي يجب على اعضاء فريق التطوير انهاء المهمة المطلوبة خلاله ، ويعبر عنه عادة بوحدات قياس مثل (شخص– يوم ، شخص – شهر أو شخص – سنة). ولهذه القيمة اهميتها الخاصة كونها الاساس لقيم اخرى ذات صلة بمشاريع البرمجيات مثل حساب الكلفة او الوقت الكلي المطلوب لانتاج منتج معين. وهنالك عدة اسباب لتخمين الجهد منها[32]:

1. **ادارة المشروع:** ان تخطيط وادارة المشروع تكون من مهام مدير المشروع وهاتان الفعاليتان تستوجبان تخمين الجهد بموجب مراحل خاصة لغرض اكمال المشروع.

2. الادراك والفهم من قبل فريق التطوير: لكي ينجز فريق التطوير عمله بكفاءة ، فانه من الضروري ان يفهم اعضاء الفريق ان لكل فرد اهميته ودوره في انجاز العمل بصورة خاصة وكذلك فهم وادراك الفعاليات المطلوبة منهم بصورة عامة.

3. الموافقة على المشروع: لابد ان يكون هناك قرار يتخذ من قبل طرف معين في المنظمة على اعطاء موافقة لبدء انطلاق المشروع ولكن يجب ان يسبق هذا القرار بتخمين الجهد المطلوب والذي يكون لازما لاكمال المشروع بنجاح.

وقد اجريت عمليات مسح مستقلة لتقييم أهمية تقدير الجهد في مجال تطوير البرمجيات ، وأفادت هذه العمليات بأن 70-85% من المستطلعين وافقوا على أهمية تقدير الجهد. وفي الوقت الحالي أصبح إيجاد نماذج جيدة لتخمين الجهد احد أكثر الأهداف أهمية في مجتمع هندسة البرمجيات [26][5] . يمكن ان تصنف الطرائق المستخدمة في تخمين الجهد الى ثلاثة اصناف او اكثر[29] :

1. التناظر التاريخي (Historical Analogy): عند توفر بيانات تاريخية سابقة مشابهة فان هذه البيانات (المسجلة والمقيدة والتابعة لمشاريع سابقة مكتملة) تستخدم لحساب الجهد المخمن للمشاريع المستقبلية.

2. قرار الخبراء (Experts' Decision): يعتمد تخمين الجهد في هذه الطريقة على شخص خبير، حيث تعتمد خبرته على ما واجهه من مشاريع سابقة مشابهة للمشاريع الحالية المراد ايجاد الجهد المخمن لها. وتكون هذه الطريقة دقيقة الى حد لابأس به عندما يكون الشخص المخمن لديه خبرة كافية في مجال البرمجيات ومجال التخمين على حد سواء.

3.استخدام قوانين او مايسمى بالـ (rules-of-thumb): ولها اشكال مختلفة قد تشمل معادلات رياضية بسيطة او قد تخصص نسبة مئوية من الجهد على مراحل وفعاليات معينة بناءا على بيانات تاريخية سابقة.

في مجال هندسة الانظمة ، تعتبر البيانات التاريخية السابقة قاعدة واساس لتخمين الجهد او الكلفة للمشاريع المستقبلية ، لكن في معظم الاحيان ولسوء الحظ (خاصة في مجال انتاج البرمجيات) فانه من الصعب (اذا لم يكن من المستحيل) ايجاد البيانات الموثوقة. وتكون عملية ايجاد وتخمين الجهد صعبة جدا عند بناء وتصميم مشاريع وانظمة كبيرة ومعقدة وذلك للاسباب التالية [3]:

1. نظام او مشروع بهذا الحجم / او النوع لم يسبق ان بني مثله من قبل.

2. قد استخدمت فيه تقنيات جديدة لم تستخدم قبل ذلك.

3. قد يكون معدل الانتاجية له (Productivity of Personnel) متباين ومتغاير بشكل كبير.

وتعد طريقة الكوكومو (COCOMO) والتي قدمت من قبل العالم بوهيم سنة (1981) واحدة من اوائل الطرق التي استخدمت لحساب الجهد حيث يكون فيه الجهد المخمن عبارة عن دالة مكونة من الحجم المتوقع ومجموعة متغيرات. وصيغتها كما في المعادلة (1) [6].

$$E = a\,S^{\,b} \qquad \ldots(1)$$

حيث ان:

E: تمثل الجهد المطلوب.

S : تمثل الحجم والذي يقاس عادة ب طول الكود.

a ,b : عبارة عن ثوابت.

تعتمد معظم نماذج تخمين الجهد على اشتقاقات تجريبية (Empirical Derivation) حيث يتم جمع بيانات من مشاريع سابقة ، معظم هذه النماذج يكون فيها حجم البرمجيات ادخالا لحساب الجهد المخمن وهذا الحجم يقاس اما باستخدام LOC او الـ FP [13].

4. البرمجة الجينية (Genetic Programming GP):

تعتبر البرمجة الجينية امتداد للخوارزمية الجينية وهي محاولة للاجابة على احد الاسئلة الاساسية في علم الحاسوب [17]: كيف يمكن لجهاز الحاسوب تعلم حل المشاكل بدون ان يتم برمجته بشكل صريح، او بعبارة اخرى كيف يمكن جعل اجهزة الحاسوب تقوم بعمل معين دون ان يقال لها بالضبط كيف تفعل ذلك؟

وهي عبارة عن طريقة لتوليد برامج الحاسوب ذاتيا وهي تسهم في حل المشاكل المحددة بعناية حيث انها تشكل احدى تقنيات الحوسبة التطويرية (Evolutionary Computations) [27]، وقد استخدمت هذه الطريقة لحل عدد كبير من المشاكل الصعبة كنمذجة العمليات الصناعية ، وتنبؤات تدفق النهر وغيرها.

وتعد البرمجة الجينية او ما تسمى بالبرمجة الوراثية واحدة من اهم الخوارزميات التطويرية التي تتبع نظرية التطور وفكرة البقاء للاقوى. وتكون على شكل مجتمع من الافراد (Population of Individuals) يمثل فيه الكروموسوم بطريقة تختلف عن الشكل المتسلسل المستخدم في الخوارزمية الجينية سواء كان ثنائي (Binary String) او اي نوع اخر من البيانات ولايوجد قيود على هيكلية البيانات الناتجة. يتم تمثيل الكروموسوم على شكل شجرة وقد يكون الناتج معادلة او برنامج او اي تمثيل اخر[16].

هنالك اربع خطوات مطلوبة لتهيئة الـ (GP) والتي يجب اتخاذها لتتلائم مع المشكلة المراد حلها [18]:

1. تهيئة الدوال والمتغيرات اللازمة (Terminal and Function sets) وتعريفها حسب المشكلة .
2. تهيئة القانون الملائم لدالة اللياقة لانها تعتبر مسالة مهمة وتختلف النتائج باختلاف القانون المستخدم.
3. تهيئة معاملات السيطرة (Control Parameters) وتشمل (عدد مرات التنفيذ، حجم الشجرة وعمقها ، حجم المجتمع وعدد افراده، نسبة الـ Crossover ونسبة الـ Mutation وغيرها).
4. تهيئة شرط التوقف بما يتلاءم مع المسألة .

بعد خطوات التهيئة تبدأ عملية البرمجة الجينية بمجتمع عشوائي من الكروموسومات اوالافراد التي غالبا ما تعبر عن برامج حاسوبية وتكون هذه الكروموسومات مؤلفة عشوائيا من مجموعة الدوال والمتغيرات الملائمة لتلك المسألة والتي تحدد عادة من قبل المستخدم وكما ذكر سابقا في الخطوة الاولى من خطوات التهيئة. وتكون هذه الكروموسومات ذات احجام مختلفة (عدد الدوال والمتغيرات لكل كروموسوم يختلف عن الكروموسوم الاخر).

يتم قياس اداء كل فرد او كروموسوم في المجتمع وكيف ينجز مهمته من خلال استخدام دالة اللياقة (Fitness Function) وكما هو مذكور في الخطوة الثانية من خطوات التهيئة. تقاس هذه الدالة بطرق مختلفة مثل نسبة الخطأ الناتج بين الاخراج الحقيقي والاخراج المطلوب، او تقاس بمقدار الوقت (نقود ، وقود ،....) اللازم لايصال النظام لحالة الهدف المنشود، او بمدى الدقة الناتجة في عملية تمييز الانماط او تصنيف الكائن الى الفئات المطلوبة وحسب المسألة المعطاة [18]. بعد ذلك تطبق عملية الـ (Selection) والعمليات الجينية لتكوين جيل جديد من الافراد (offspring) من الجيل الحالي بالاعتماد على مبدأ كون الفرد الذي يمتلك دالة لياقة اكبر له فرصة افضل بالانتقال الى الجيل الجديد.

5. البرمجة متعددة التعابير (Multi Expression Programming):

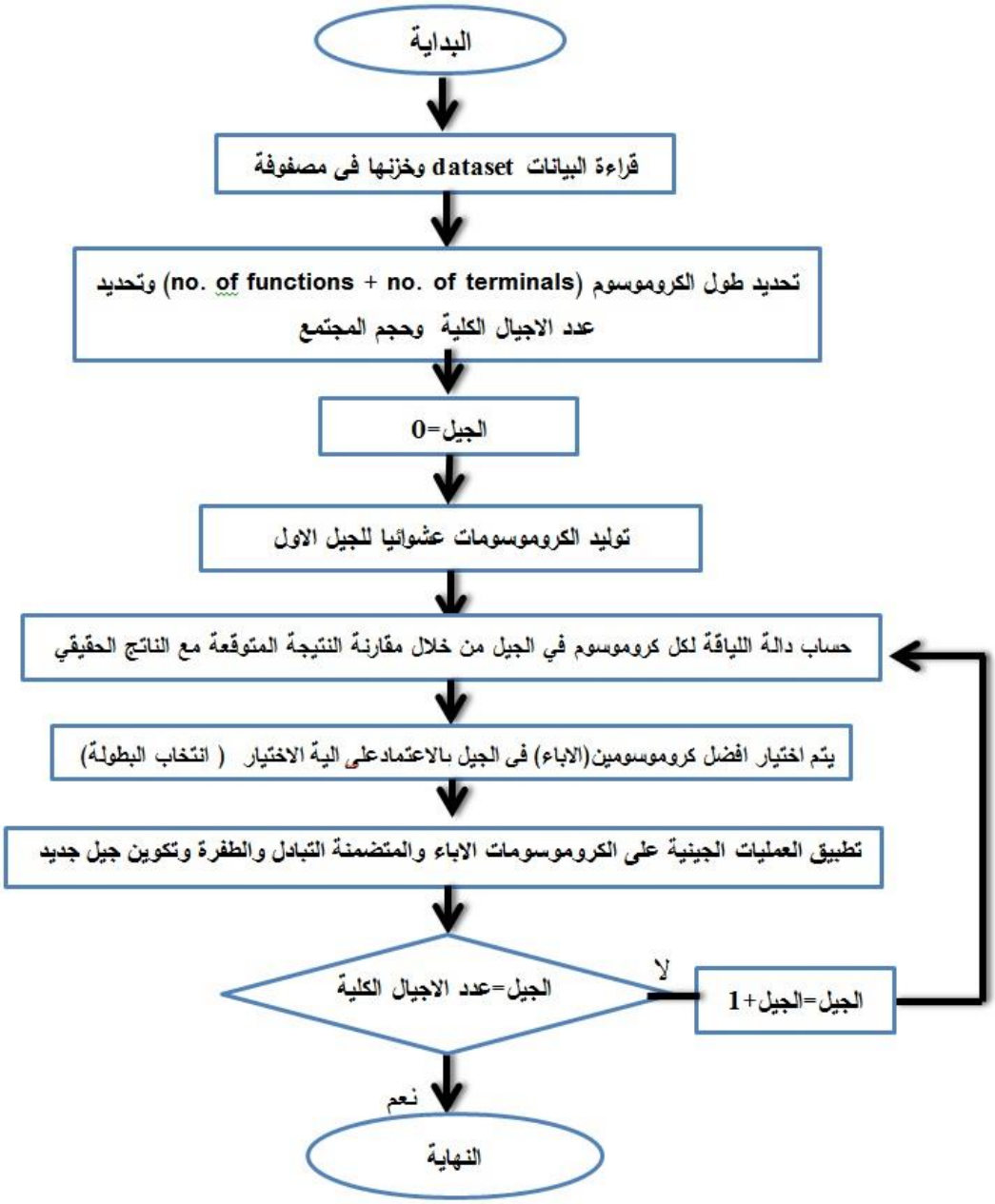
ان البرمجة متعددة التعابير هي عبارة عن برمجة جينية خطية ويكمن الفرق بين الاثنين في[24]:

1. التشفير لتعبير واحد في البرمجة الجينية ، بينما البرمجة متعددة التعابير تشفر اكثر من تعبير.
2. يتم فيها تمثيل الكروموسوم بشكل خطي حيث يتمثل الكروموسوم او الفرد بشكل سلسلة من الجينات تشفر به برامج الكمبيوتر المعقدة. والمخطط الانسيابي في الشكل (1) يوضح آلية العمل باستخدام البرمجة متعددة التعابير.

1.5. التمثيل الجيني

يكون عدد الجينات لكل كروموسوم ثابت حيث يعرف طول الكروموسوم بعدد جيناته. كل جين من هذه الجينات يشفر اما (Terminal Symbol) او (Function Symbol). والجين الذي يشفر دالة (عملية حسابية) يجب ان يحتوي على مؤشر يدل على عناصر تلك الدالة. ويجب ان يبدأ اول جين في الكروموسوم دائما بـ (Terminal Symbol). ويكون عملها مشابها للغتي باسكال وسي عندما تقومان بتحويل المعادلات الرياضية وترجمتها الى لغة الماكنة [12]. ومن الخصائص التي تميز هذه الطريقة هي القدرة على خزن عدة حلول للمشكلة في الكروموسوم او الفرد. ويتم عادة اختيار افضل حل حسب دالة اللياقة لذلك الفرد [24]. وتتلخص الخطوات الرئيسية للخوارزمية القياسية للبرمجة متعددة التعابير بالشكل(2) [23].

فيما يلي توضيح للكيفية التي يمثل بها الكروموسوم بطريقة البرمجة متعددة التعابير بالتمثيل الخطي للكروموسوم والشكل الحقيقي غير الخطي للكروموسوم (شكل شجرة) وآلية حساب دالة اللياقة للكروموسوم [24] وعلى افتراض وجود كروموسوم (C) يتألف من عدد من الجينات ، مجموعة العمليات{*,+}=F ، ومجموعة الطرفيات {a,b,c,d}=T .



الشكل (1) يوضح الية العمل باستخدام للبرمجة متعددة التعابير

فان التمثيل بطريقة البرمجة متعددة التعابير سيكون كالتالي:

1: a
2: b
3: + 1, 2
4: c
5: d
6: + 4, 5
7: * 3, 5
8: + 2, 6

حيث ان العدد الاقصى للرموز في الكروموسوم يحسب من خلال المعادلة (2).

$$Number\ of\ Symbols = (n+1) * (Number\ of\ Genes - 1) + 1 \quad \ldots(2)$$

ويمثل (n) عدد المعاملات للدالة التي تتطلب اكبر عدد من المعاملات .

```
Begin
        Generate Initial Population;
        t = 0;
        Evaluate_Individuals;
        While Not Termination_Condition Do
        Elitism;
        Selection;
        Recombination;
        Mutation;
        Evaluate_Individuals;
        End while
End
```

الشكل (2) الخطوات الاساسية للعملية التطويرية للبرمجة متعددة التعابير

ان الجينات المرقمة بـ {1,2,4,5} قد شفرت تعابير بسيطة (Simple Expressions) او ماتسمى بالـ (Terminal). اما باقي الجينات شفرت تعابير معقدة (Complex Expression) لانها تحتوي على عمليات حسابية او دوال.

E1 = a,
E2 = b,
E4 = c,
E5 =d,

ان الارقام اعلاه من 1-8 ليس لها علاقة بالكروموسوم فقط كل رقم يمثل تعبير وقد قدمت لغرض التوضيح فقط. حيث ان الـ E هنا تمثل Expression. اما التعبير الثالث المرقم بـ {3} فقد استخدم الدالة او العملية الحسابية {+} مع وجود عنصري هذه الدالة (Operand) وهما عبارة عن مؤشرين يؤشران الى المواقع المرقمة بـ{1} و{2} من الكروموسوم.

وكما ذكر سابقا فانه يجب ان يكون مع كل دالة او عملية حسابية مؤشرات تشير الى عناصر تلك الدالة ويجب ان تكون قيم تلك المؤشرات اقل من موقع العملية الحسابية الحالية. اي ان فك تشفير التعبير الثالث وتفسيره ينتج (**E3 = a + b**) وهكذا بالنسبة لبقية الكروموسوم

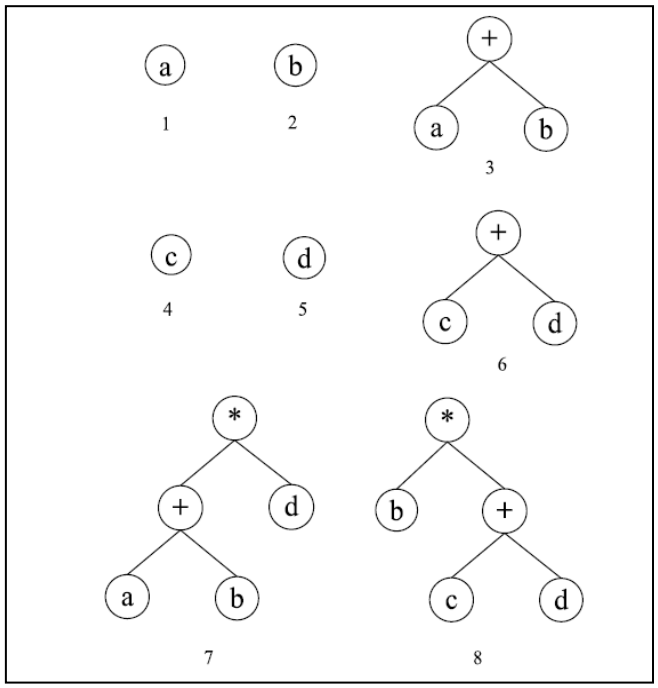
E6 = c + d
E7 = (a + b) * d
E8 = b * (c + d)

ان التعابير اعلاه المرقمة بـ {3,6,7,8} هي تعابير معقدة لانها تحتوي على دوال. سميت هذه الطريقة بالبرمجة متعددة التعابير لسماحها بتشفير عدة تعابير ويكون عددها مساوياً لطول الكروموسوم (عدد جيناته). ويلاحظ في هذا الكروموسوم ان طوله 8 جين وعدد التعابير الموجودة كذلك مساوية لهذه القيمة. فالشكل النهائي لهذا الكروموسوم:

E1 = a,
E2 = b,
E3 = a + b,
E4 = c,
E5 = d,

E6 = c + d,
E7 = (a + b) * d,
E8 = b * (c + d).

ويوضح الشكل (3) تمثيل الكروموسوم اعلاه على شكل شجرة مع ملاحظة ان الارقام الموجودة اسفل كل تفرع تمثل رقم التعبير الموجود في الكروموسوم اعلاه.



الشكل (3) تمثيل الكروموسوم كشجرة[24].

2.5. دالة اللياقة

يتم اختيار افضل فرد في الكروموسوم حسب دالة الياقة والتي تعرف بأنها دالة اللياقة لافضل تعبير موجود في ذلك الكروموسوم. ويتم حساب دالة اللياقة حسب القانون في المعادلة (3). وبعد تطبيق القانون السابق يتم اختيار دالة اللياقة الاقل قيمة للتعابير الموجودة في الكروموسوم كما في المعادلة (4) [24].

$$f(Ei) = \sum_{k=1}^{n} | O_{k,i} - W_k | \quad \ldots(3)$$

حيث ان:

O_k : تمثل القيمة الناتجة من ذلك التعبير

W_K: يمثل الناتج الحقيقي (Actual Output).

$$f(C) = \min f(E_i) \quad \ldots(4)$$

تكمن الاستفادة من استخدام هذه الطريقة في تجاوز المشاكل العديدة التي تعرقل تطبيق البرمجة الجينية والتي تترأسها مشكلة التعامل البرمجي مع الاشجار وصعوبة اجراء العمليات الجينية عليها ، بالاضافة الى ذلك فان تحديد حجم الشجرة وعمقها يمثل مسألة حرجة في انجاح العملية التطويرية والحفاظ على حجم الشجرة بعد اجراء عمليات متتابعة من التبادل بين اغصانها مع الاشجار الاخرى واحداث عملية الطفرة الوراثية لانتاج برنامج صحيح خالي من الاخطاء.

3.5. العمليات الجينية

تستخدم هذه الطريقة عدد من العمليات الجينية وكما يلي:

1. التبادل: حيث يتم اختيار اثنين من الكروموسومات بطريقة البطولة ويحدث التبادل بنفس الاسلوب المتبع في الخوارزميات الجينية وتتم باستخدام:

طريقة (One-point Recombination)

طريقة (Two-point Recombination)

طريقة (Uniform Recombination)

2. الطفرة الوراثية : تخضع جميع الرموز في الكروموسوم لاحتمال حدوث طفرة وراثية في قيمها وحين يتغير الرمز من طرفي الى دالة فان المعملات سوف تتولد تلقائيا وقد تتغير الدالة الى رمز طرفي وتهمل معاملاتها.

4.5. الاختيار

هي المرحلة التي يتم فيها اختيار افراد من المجتمع لاجراء العمليات الجينية عليهم لاحقا وحسب لياقتهم حيث تم في هذا العمل اختيار طريقة انتخاب البطولة (tournament selection) ، يتم تنفيذ الطريقة بين مجموعة من الافراد تم اختيارهم عشوائيا من ضمن المجتمع والفائز منهم (افضل فرد) يتم اختياره لاجراء العمليات الجينية عليه. ويتم اختيار حجم الانتخاب من ضمن المجتمع وبشكل عشوائي . وهذه الطريقة لها فوائدهاحيث يتم تحويلها الى شفرة برمجية بسهولة كذلك يتم تطبيقها على المعماريات المتوازية [30]. تضمن عملية الاختيار اعطاء فرصة اكبر لافضل فرد في المجتمع كي ينتقل للجيل التالي . وقد اثبتت معظم البحوث ان الحجم الانسب =2 وقد طبق هذا الحجم في الجانب العملي من البحث[23].

6.التجارب والنتائج

تم في هذا العمل دراسة امكانية ايجاد دالة لتخمين الجهد للبرمجيات وذلك من خلال تطبيق طريقة البرمجة متعددة التعابير(MEP) على البيانات الموجودة في مجاميع البيانات المذكورة في الجدول (1). حيث تم اعتماد عدد من المجاميع المختلفة والمتنوعة وذلك لتوفرها علنا ولكثرة ورودها واعتمادها من قبل الباحثين في هذا المجال الامر الذي جعلها تصبح مجاميع قياسية تستخدم في المقارنة بين الطرق المختلفة المستخدمة في تخمين جهد البرمجية. وفيما يلي التجارب التي اجريت في هذا العمل مع التحليل لنتائجه.

1.6. التجربة الاولى

تتضمن التجربة الاولى في هذا العمل تطبيق خوارزمية البرمجة متعددة التعابير على مجاميع البيانات المذكورة في الجدول (1) وبعد الحصول على النتائج تتم المقارنة بينها وبين طريقة البرمجة الجينية التي استخدمت من قبل الباحث (Dolado) [8]. وقد استخدمت نسبة تبادل (0.7) وطفرة (0.05) لكل التجارب.

جدول (1) مجاميع البيانات المستخدمة في التجارب

ت	اسم مجموعة البيانات	اسم الباحث	عدد المشاريع الكلية	عدد المشاريع
1	Albrecht & Gaffney[1]	A.J. Albrecht, J.R. Gaffney	24 points: 5 incomplete points (3,6,7,22,24)	24 points
2	Bailey & Basili[4]	J.W. Bailey, V.R. Basili	18 points	18 points

35 points	35 points	A. Heiat, N. Heiat	Heiat & Heiat[11]	**3**
15 points	15 points	C.F. Kemerer	Kemerer[15]	**4**
48 points	48 points	Y. Miyazaki, M. Terakado, K. Ozaki, H. Nozaki	Miyazaki et al.[21]	**5**
77 points	81 points: 4 incomplete points (38,44,66,75)	J.M. Desharnais	Desharnais[7]	**6**

تتضمن عملية تهيئة خوارزمية البرمجة متعددة التعابير تحديد قيم الاعدادات لمعاملاتها حيث تم تحديد الحد الاقصى للمجموعة السكانية بـ 40 فرد وعدد الاجيال الكلي لكل تنفيذ بـ 200 جيل. تتكون مجموعة العمليات من { + ، − ، / ، * ، power ، exp ، log ، sqrt} ومجموعة الطرفيات تتكون من المتغيرات للمشاريع وحسب مجاميع البيانات.

ويوضح الجدول (2) النتائج المستحصلة مع المقارنة والتي تظهر مدى تفوق طريقة البرمجة متعددة التعابير في ايجاد نتائج افضل من البرمجة الجينية بشكل ملحوظ . يظهر الجدول القيمة المثلى لدالة اللياقة بالاضافة عدد الاجيال التي تطلبتها عملية ايجاد تلك القيمة المثلى. اقتصرت المقارنة على قيمة دالة اللياقة فقط ولم يشمل حجم المجموعة السكانية وعدد الاجيال لعدم ذكرها في البحوث التي تستخدم البرمجة الجينية.

جدول (2) المقارنة بين نتائج البرمجة متعددة التعابير والبرمجة الجينية للتجربة الاولى

	مجموعة البيانات	نتائج البرمجة الجينية	نتائج البرمجة متعددة التعابير	
		دالة اللياقة	دالة اللياقة	الاجيال
.1	Albrecht & Gaffney	0.548	0. 33910	129
.2	Bailey and Basili	0.269	0.142	95
.3	Desharnais	0.623	0.38951	107
.4	Heiat & Heiat	0.087	0.0857	200
.5	Kemerer	0.584	0.36854	200
.6	Miyazaki	0.506	0.3242	200

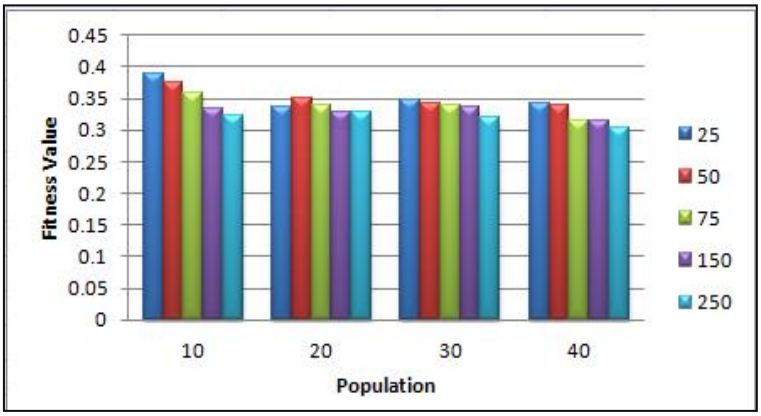
2.6. التجربة الثانية

تم في هذه التجربة اجراء دراسة لحدود المجتمع وعدد الاجيال اللازمة لاستقصاء فضاء البحث الخاص بالمسألة وباستخدام نفس مجموعات العمليات والطرفيات، وقد اجريت هذه التجربة على مرحلتين:

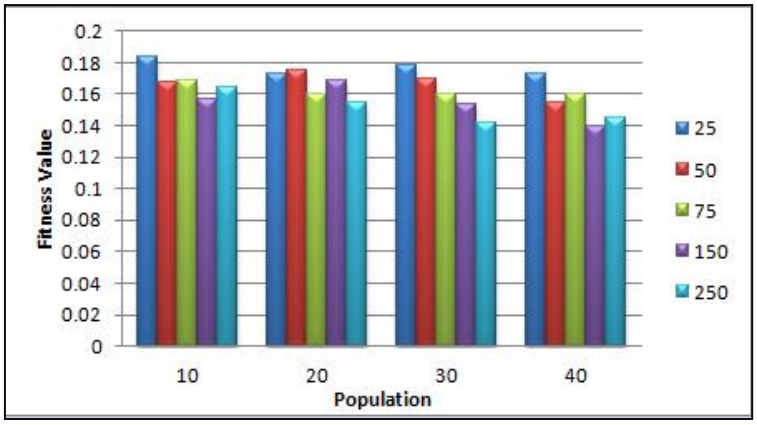
المرحلة الاولى وتضمنت اخذ مجموعة من العينات للمجتمع من الاحجام الصغيرة المتقاربة التالية (10, 20, 30, 40) واعداد اجيال محددة بالاعداد التالية (25 ، 50، 75 ، 150 ، 250).

المرحلة الثانية وتضمنت عينات اكبر واكثر تباعدا للمجتمع (50 ، 100، 150، 200 ، 250 ، 300 ، 350 ، 400 ، 450 ، 500) ومجموعة مماثلة للاجيال.

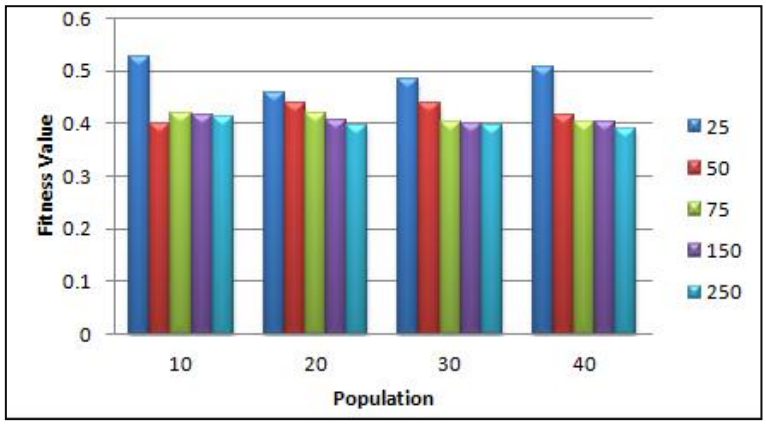
وتوضح المخططات المتتالية بدآ من الشكل (4) الى الشكل (9) نتائج تطبيق عينات المرحلة الاولى على مجاميع البيانات المذكورة في الجدول (1) ، حيث يمثل الاحداثي السيني المجتمع على شكل مجاميع ، وضمن كل مجموعة اعداد الاجيال والمتمثلة بالالوان المختلفة والموضح قيمها في الـ (Legend) على يمين المخطط. ويمثل المحور الصادي قيمة دالة اللياقة المستحصلة لكل مجتمع وعدد اجيال.



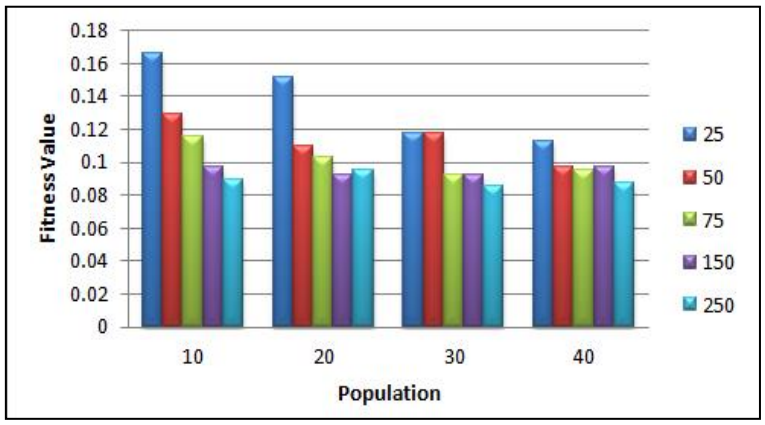
الشكل (4) مخطط النتائج لمجموعة بيانات (Albrecht & Gaffany)



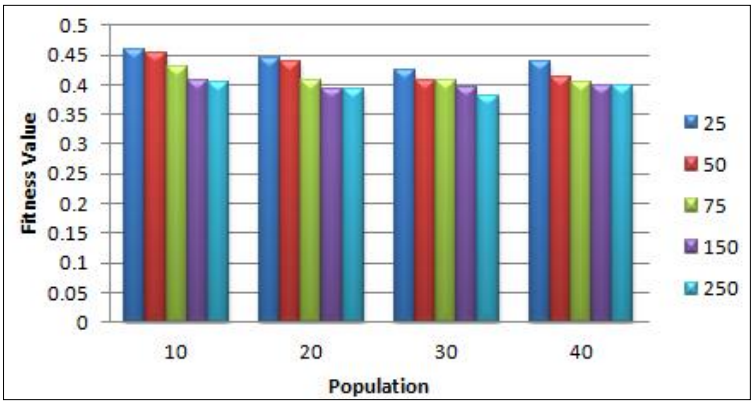
الشكل (5) مخطط النتائج لمجموعة بيانات (Bailey & Basili)



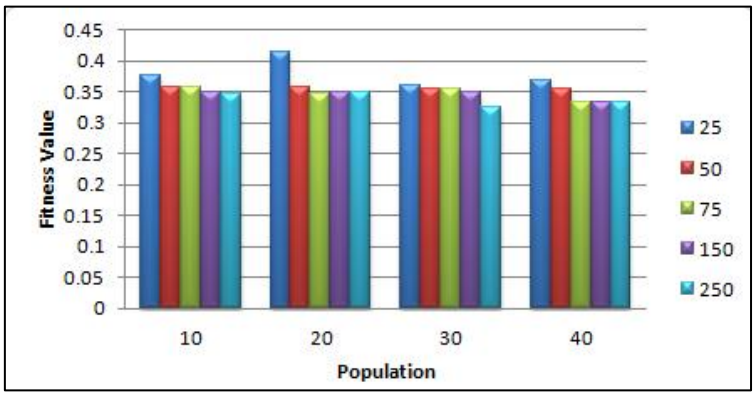
الشكل (6) مخطط النتائج لمجموعة بيانات (Desharnais)



الشكل (7) مخطط النتائج لمجموعة بيانات (Heiat & Heiat)



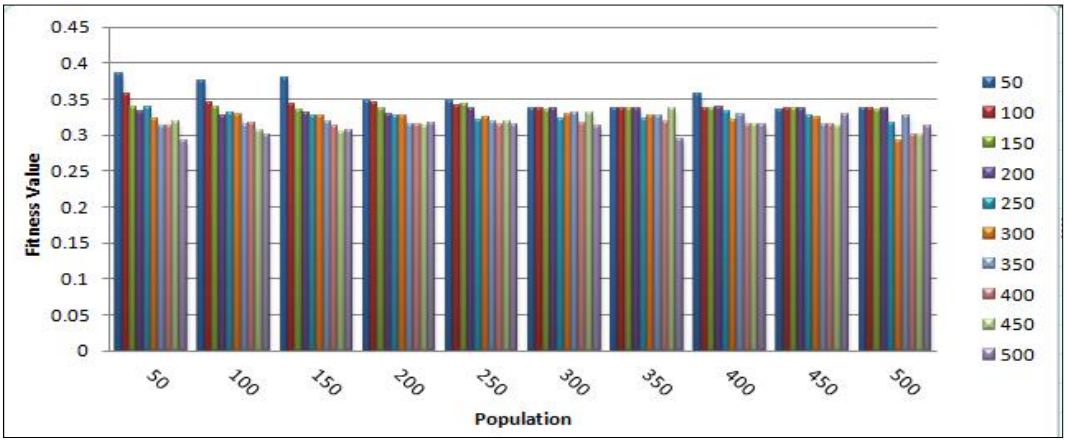
الشكل (8) مخطط النتائج لمجموعة بيانات (Kemerer)



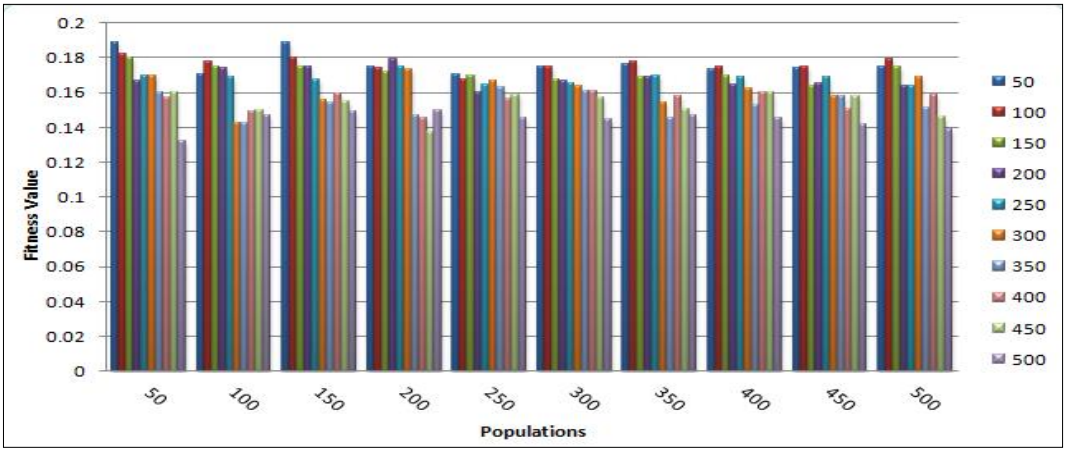
الشكل (9) مخطط النتائج لمجموعة بيانات (Miyazaki)

وكما تظهر المخططات السابقة فان بامكان خوارزمية البرمجة متعددة التعابير ان تحقق نجاحا متميزا باحجام صغيرة من المجتمعات واعداد قليلة من الاجيال ، وتوضح ايضا ان زيادة عدد الاجيال يزيد من فرص الحصول على نتائج افضل وذلك بسبب طبيعة الخوارزمية التطويرية حيث ان الاجيال تمثل الوقت اللازم لاتمام عملية التطور ويمثل المجتمع عدد الافراد التي تجسد عملية التطور على شكل حلول تتنافس فيما بينها مع فكرة البقاء للاقوى.

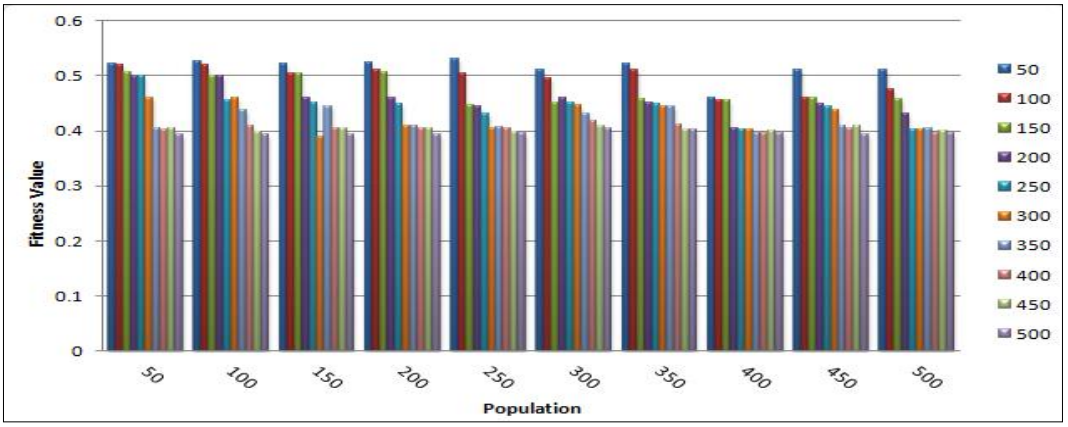
اما الاشكال من (10) الى (5) فتبين نتائج المرحلة الثانية للتجربة والتي تضمنت احجام اكبر للمجتمعات واعداد اوسع للاجيال كمحاولة لاستقصاء اوسع لفضاء البحث ولدراسة استراتيجية الخوارزمية في عملية البحث عن المعادلات الافضل والتي تلائم مجاميع البيانات بشكل ادق وحسب دالة اللياقة.



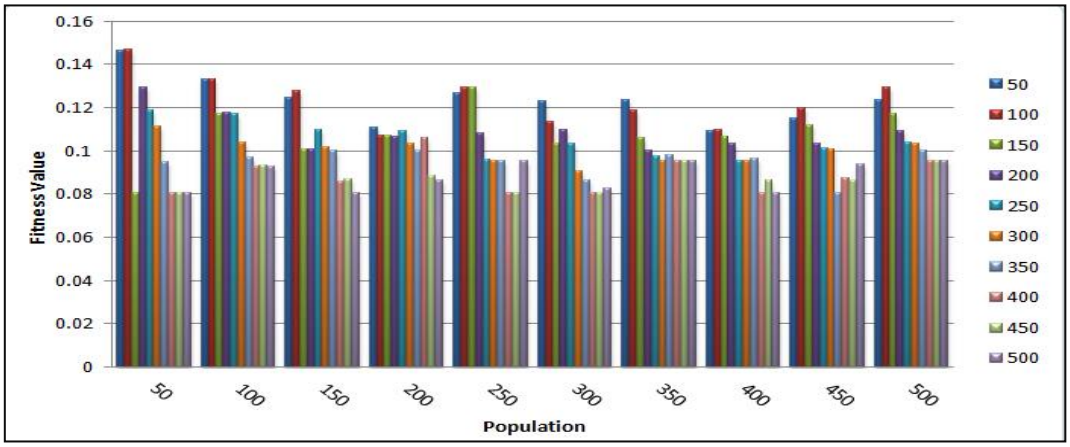
الشكل(10) مخطط النتائج لمجموعة بيانات (Albrecht & Gaffany)



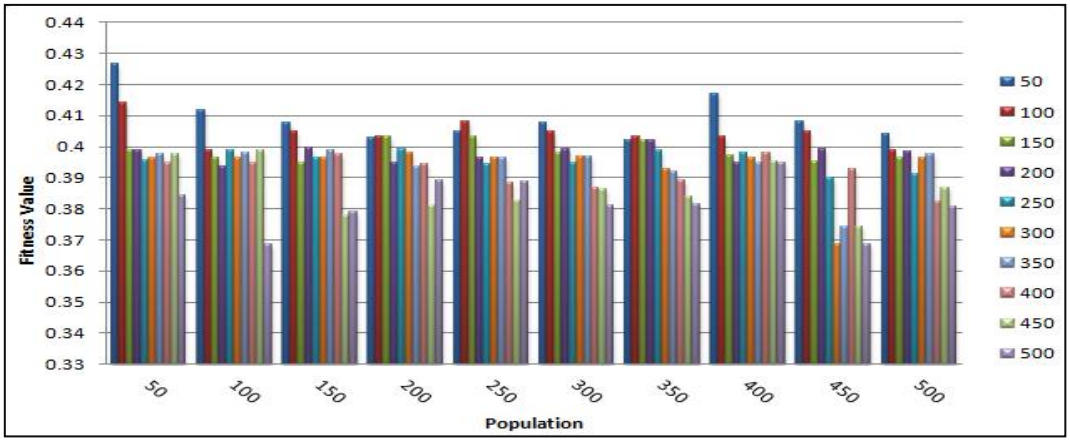
الشكل (11) مخطط النتائج لمجموعة بيانات (Bailey & Basili)



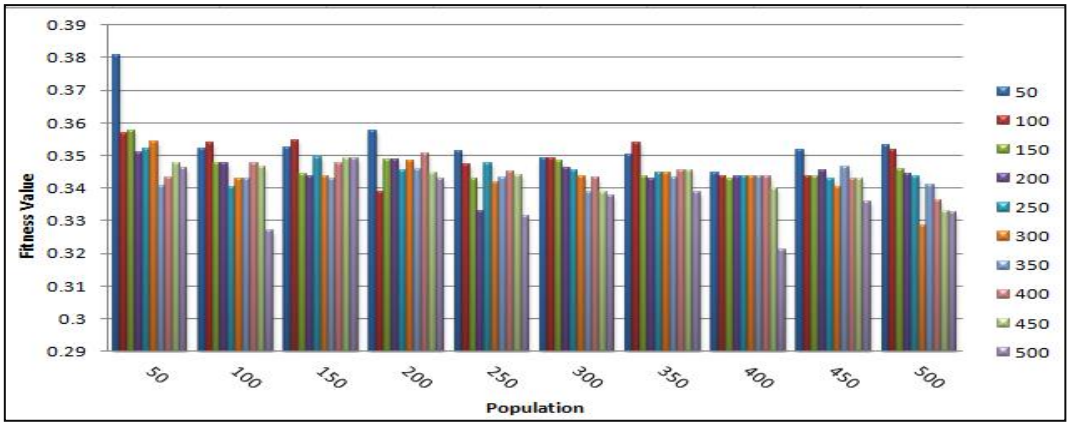
الشكل (12) مخطط النتائج لمجموعة بيانات (Desharnais)



الشكل (13) مخطط النتائج لمجموعة بيانات (Heiat & Heiat)



الشكل (14) مخطط النتائج لمجموعة بيانات (Kemerer)



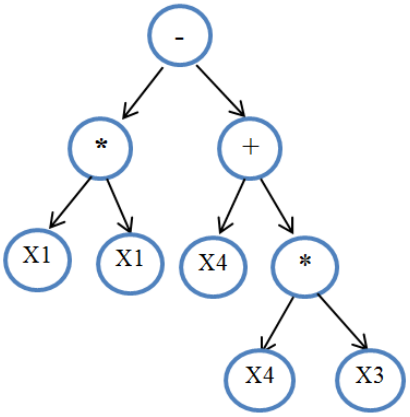
الشكل (15) مخطط النتائج لمجموعة بيانات (Miyazaki)

وكما هو مبين من خلال المخططات السابقة فان زيادة اعداد الاجيال لكل مجتمع ادى الى تحسين النتائج بشكل عام ولكل مجاميع البيانات اما بالنسبة لتكبير حجم المجتمع فقد كان التحسن طفيفاً جدا مما يعكس حقيقة كون المجتمعات الصغيرة المأخوذة في المرحلة الاولى كانت كافية لاستقصاء فضاء البحث ولم يكن هناك استفادة واضحة من زيادة الحجم. وبشكل عام فان الاستقصاء الاوسع لفضاء البحث لم يؤدي الى اقتراب ادق الى النتائج المثلى ولم يتمكن من ايجاد نتائج ادق مما توصلت اليه التجربة الاولى ولكنه اوضح الحدود المناسبة والملائمة التي

تحتاجها عملية البحث للوصول الى نتائج جيدة جدا وافضل بكثير مما توصلت اليه الطرق السابقة وكما هو مبين في التجربة الاولى والجدول (2).

وفيما يلي مثال توضيحي لاحد النتائج المستحصلة من التجارب السابقة :

بعد تطبيق البرمجة متعددة التعابير على احد البيانات المستخدمة والمذكورة في الجدول رقم (1) كان الناتج المعادلة التالية : $\boldsymbol{E = x1 * x1 - x4 + x4 * x3}$



الشكل رقم (16) يمثل المعادلة اعلاه كشجرة

7. الاستنتاجات والتوصيات المستقبلية

ان الغرض الاساسي من اجراء هذه الدراسة هو الاستفادة من التقنية الذكية للبرمجة الجينية في ايجاد معادلات لتخمين الجهد للبرمجيات مع امكانية التخلص من مشاكل البرمجة الجينية العديدة كصعوبة برمجة الاشجار والتعامل معها ومشكلة التحديد المسبق لحجم الشجرة والمشاكل التي تنتج عن عملية اجراء عمليات التبادل والطفرات الوراثية وغيرها من مشاكل البرمجة الجينية.

لذا فقد تم تجاوز المشاكل السابقة من خلال استخدام الطريقة الخطية للبرمجة الجينية والمسماة **بالبرمجة المتعددة التعابير** وتطبيقها بنجاح على مسالة تخمين الجهد للبرمجيات وذلك من خلال ايجاد معادلات تقوم باعطاء الحلول التخمينية لكلف المشاريع قبل الشروع في تنفيذها مما يعود بالنفع على الشركة في اتمام المشاريع بشكل كفوء. وقد تم مقارنة النتائج المستحصلة مع نتائج البرمجة الجينية وكانت افضل بكثير من حيث الدقة.

تم ايضا في هذا البحث اجراء استقصاء لفضاء البحث من خلال دراسة الاحجام المختلفة للمجتمع ولاعداد الاجيال المطلوبة وتم ذلك على مرحلتين لبيان تاثير تلك الاحجام والاعداد على عملية البحث.

اما فيما يختص بالاعمال المستقبلية فان المجال واسع جدا لتطبيق مختلف اساليب الذكاء الاصطناعي وتقنياته المتغايرة لمحاولة ايجاد حلول افضل لمسالة تخمين الجهد للبرمجيات في المشاريع الكبيره للشركات التي تمثل فيها عملية تحديد الكلفة مسبقا عملية حيوية لنجاح الشركة وتحقيق الارباح الجيدة. بالامكان تطبيق الطرق الخطية الاخرى المقترحة للبرمجة الجينية ومقارنة النتائج ، وبالامكان ايضا استخدام احد خوارزميات ذكاء السرب العديدة والمختلفة لاستقصاء فضاء البحث واجراء المقارنات اللازمة لبيان كفاءة كل منها.

المصادر

[1] Albrecht1, A.J., Gaffney, J.R., (1983)," Software Function, Source Lines of Code, and Development Effort Prediction: a Software Science Validation", IEEE Transactions on Software Engineering 9 (6) 639–648.

[2] Arnuphaptrairong, T., (2013),"Early Stage Software Effort Estimation Using Function Point Analysis: Empirical Evidence", Proceedings of the International MultiConference of Engineers and Computer Scientists Vol. II, (IMECS), March 13-15, Hong Kong. pp: 730-735.

[3] Asundi, J. (2005)," The Need for Effort Estimation Models for Open Source Software Projects", Software Engineering (5-WOSSE) May 17, St Louis, MO, USA. © ACM 1-59593-127-9.pp:1-3.

[4] Bailey, J.W., Basili, V.R., (1981), "A Meta-model for Software Development Resource Expenditures". Proceedings of the Fifth International Conference on Software Engineering, 1981, pp: 107–116.

[5] Bhatnagar, R., Ghose, M.K., (2012) "Early Stage Software Development Effort Estimations-Mamdani FIS VS Neural Network Models", CS & IT , pp. 377–384.

[6] Burgess, C., Lefley, M., (2001) "Can Genetic Programming Improve Software Effort Estimation: A Comparative Evaluation". Inform. and Softw. Technology 43(14), pp: 863–873.

[7] Desharnais, J.M., (1988), "Analyse statistique de la productivite´ des projects de de´velopment en informatique a` partir de la technique des points de function", Master's Thesis, Univ. du Que´bec a` Montreal, De´cembre,.

[8] Dolado J.J., (2001) On the problem of the software cost function", Information and Software Technology, pp: 2001 Elsevier Science B.V.

[9] Ferrucci, F., Gravino, C., Oliveto, R., Sarro, F. (2010) "Genetic Programming for Effort Estimation: An Analysis of the Impact of Different Fitness Functions". In Procs. of SSBSE, pp: 89–98.

[10] Huang, S., Chiu, N. (2006) Optimization of Analogy Weights by Genetic Algorithm for Software Effort Estimation. Journal of Systems and Software 48 (11), pp:1034-1045.

[11] Heiat, A., Heiat, N., (1997) "A Model for Estimating Efforts Required for Developing Small-Scale Business Applications", Journal of Systems and Software 39 (1) pp:7–14.

[12] Heshmati, A.A.R., Salehzade, H., Alavi, A.H., Gandomi, A.H., and Abadi M. M., (2010) "A Multi Expression Programming Application to High Performance Concrete", World Applied Sciences Journal 11 (11), pp: 1458-1466, ISSN 1818-4952. © IDOSI Publications.

[13] Jeng, B., Yeh, D, Wang, D., Chu, S., and Chen, C., (2011),"A Specific Effort Estimation Method Using Function Point", Journal Of Information Science And Engineering 27, pp: 1363-1376.

[14] Jørgensen, M., (2004), "Regression Models of Software Development Effort Estimation Accuracy and Bias". Empirical Software Engineering, 9, 297–314. Kluwer Academic Publishers. Manufactured in The Netherlands

[15] Kemerer, C.F., (1987), "An Empirical Validation of Software Cost Estimation Models", Communications of the Association for Computing Machinery 30 (5). pp:416–429.

[16] Koza, J.R., (1992)," Genetic Programming On the Programming of Computers by Means of Natural Selection", © Massachusetts Institute of Technology.

[17] Koza, J.R. (1994), "Genetic Programming II: Automatic Discovery of Reusable Programs". ©1994 Massachusetts Institute of Technology.

[18] Koza, J.R., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J., Lanza, G, (2003) "Genetic Programming IV Routine Human-Competitive Machine Intelligence", ISBN 1-4020-7446-8. © Springer Science+Business Media, Inc.

[19] Lefley, M., Shepperd, M. (2003), "Using genetic programming to improve software effort estimation based on general data sets". In Procs. of Genetic and Evolutionary Computation Conference, 2003, 2477–2487.

[20] Mendes, E., Mosley,(2008) ,N. Bayesian Network Models for Web Effort Prediction: A Comparative Study. IEEE Trans. Software Eng., 34(6), 723-737 .

[21] Miyazaki, Y., Terakado, M., Ozaki, K., Nozaki, H., (1994), "Robust regression for developing software estimation models", Journal of Systems and Software 27 (1), pp: 3–16.

[22] Ohsugi,N., Tsunoda, M., Monden, A. and Matsumoto, K. (2004) , "Effort Estimation Based on Collaborative Filtering", In the 5th International Conference on Product Focused Software Process Improvement (PROFES2004), pp. 274-286.

[23] Oltean, M., Dumitrescu, D., (2002) "Multi Expression Programming".Technical-Report,UBB-01-2002.

[24] Oltean,M.,(2006),"Multi Expression Programming".Technical Report,Babes-Bolyai Univ,Romania. 28p.

[25] Patil, M.V., Yogi, AM. N., (2010). "Effort Estimation and Risk Analyses for Software projects by Data Analyses of Developed Projects", ACS-International Journal on Computational Intelligence, Vol-1, Issue-2. pp: 43-52.

[26] Quba, I., Z. (2012). "Software Projects Estimation using Neural Networks". M.Sc Thesis. College of Computers Sciences & Mathematics , University of Mosul. (in Arabic).

[27] Sheta A.F., Al-Afeef A., (2010). "A GP Effort Estimation Model Utilizing Line of Code and Methodology for NASA Software Projects", In proceeding of: 10th International Conference on Intelligent Systems

Design and Applications, ISDA, pp: 290-295.

[28] Singh, B.K., Misra, A.K., (2012)," An Alternate Soft Computing Approach for Effort Estimation by Enhancing Constructive Cost Model in Evaluation Method ". In the International Journal of Innovation, Management and Technology, Vol. 3, No. 3, pp: 272-275 ISSN: 2010-0248.

[29] Tsakonas, A., Dounias, G., (2009). "Deriving Models for Software Project Effort Estimation by Means of Genetic Programming". In KDIR-2009 Workshop (INSTICC Conference), 6-8 October 2009, Madeira, pp: 34-42.

[30] Wikipedia The Free Encyclopedia http://en.wikipedia.org/wiki/ tournament ; selection

[31] Ziauddin, Tipu S. K. and S. Zia,(2012), "An Effort Estimation Model for Agile Software Development," Advances in Computer Science and Its Applications (ACSA), Vol. 2, No. 1, , pp. 314-324.

[32] Živadinović, J., Medić, Z., Maksimovi, D., Damnjanović, A., Vujčić, S., (2011) "Methods Of Effort Estimation In Software Engineering", In International Symposium Engineering Management And Competitiveness (EMC2011), June 24-25, 2011, Zrenjanin, Serbia. pp: 417- 422.