

The Use of Original and Hybrid Flower Pollination Algorithm In Estimating The Parameters of Software Reliability Growth Models

Jamal Salahaldeen Alneamy¹ Marwah Marwan Abdulazeez Dabdoob²

^{1,2} Software Engineering Department, Computer and Mathematics Science College,
University of Mosul, Iraq

Received	Accepted
21/06/2018	15/08/2018

Abstract

In order to assess software reliability, many software reliability growth models (SRGMs) have been used for estimation of reliability growth. . In this work, the parameters of (SRGMs) were estimated by using Flower Pollination Algorithm (FPA). Then, the (FPA) was hybrid with Real Coded Genetic Algorithm (RGA) to obtain Hybrid FPA (HFPA).

The results that obtained from (FPA) are compared to the results of five algorithms: Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), the Dichotomous Artificial Bee Colony (DABC), Classic Genetic Algorithm (CGA) and the Modified Genetic Algorithm (MGA).

The results showed that (FPA) outperformed the rest of the algorithms in parameters estimating accuracy and performance using identical datasets. Sometimes, the (DABC) showed better performance than (FPA).

Other comparisons were made between (FPA) and (HFPA) and the results show that the hybrid algorithm outperformed the original one.

Keywords: Swarm Intelligence, Software Reliability Models.

استعمال خوارزمية تلقيح الزهرة الاعتيادية والمهجنة في تخمين معلمات نماذج نمو وثوقية البرمجيات

أ.م. د جمال صلاح الدين النعيمي¹ م. م مروة مروان عبد العزيز دبدوب²

^{1,2} قسم هندسة البرمجيات/ كلية علوم الحاسوب والرياضيات/ جامعة الموصل

تاريخ الاستلام تاريخ القبول
2018/06/21 2018/08/15

الخلاصة

من أجل تقييم وثوقية البرمجيات ، تم استعمال العديد من نماذج نمو وثوقية البرمجيات (Software Reliability Growth Models (SRGMs) لتخمين نمو الوثوقية. في هذا العمل ، تم تخمين معلمات الـ (SRGMs) باستعمال خوارزمية تلقيح الزهرة ((Flower Pollination Algorithm(FPA). ثم تم تهجين الـ (FPA) مع الخوارزمية الجينية ((Real) Coded Genetic Algorithm(RGA) للحصول على خوارزمية الزهرة المهجنة ((Hybrid FPA (HFPA).

تمت مقارنة النتائج التي تم الحصول عليها من خوارزمية (FPA) مع نتائج خمس

خوارزميات وهي: خوارزمية سرب الطيور ((Particle Swarm Optimization (PSO) ، خوارزمية مستعمرة النحل ((Artificial Bee Colony (ABC) والخوارزمية المطورة منها ((Dichotomous ABC(DABC) ، الخوارزمية الجينية الكلاسيكية ((Classic Genetic Algorithm (CGA) والخوارزمية المحدثة منها ((Modified Genetic Algorithm (MGA).

أظهرت النتائج تفوق خوارزمية (FPA) على باقي الخوارزميات في تخمين المعلمات من إذ الدقة والأداء وباستعمال بيانات متطابقة. في بعض الأحيان ، أظهرت خوارزمية الـ (DABC) أداء أفضل من (FPA).

وبالتالي تم إجراء مقارنات أخرى بين (FPA) و (HFPA) وأظهرت النتائج تفوق الخوارزمية المهجنة على الخوارزمية الأصلية من إذ الدقة وسرعة الأداء .

الكلمات المفتاحية: ذكاء السرب, نماذج نمو وثوقية البرمجيات.

المقدمة

تعرف وثوقية البرمجيات بـ "كيفية تحقيق البرنامج للمتطلبات المرجوة" وتعرف أيضاً "إحتمالية عمل البرنامج بدون فشل لمدة معينة من الوقت وفي بيئة محددة" [1].

إن محاولة معرفة هل أن البرنامج الذي سيسلم للزبون موثوق أم لا من الأمور الصعبة، فمن الممكن معرفة وثوقية البرنامج عندما يجهز الزبون به لكن عندها يكون قد فات الأوان، فمجهزي البرامج بحاجة إلى معرفة وثوقية البرنامج قبل تجهيز الزبون به. إن نماذج وثوقية البرمجيات تزودنا بمثل هذه المعلومات قبل التسليم [2].

خلال العقود الأربعة الأخيرة، تم اقتراح وتحليل العديد من نماذج نمو وثوقية البرمجيات لقياس نمو وثوقية البرامج. تحوي المعادلات الخاصة بالنماذج على معلمات يتم تخمينها بالإعتماد على بيانات الفشل وبما أن أغلب نماذج نمو وثوقية البرمجيات غير خطية فأصبح من الصعب استعمال الطرائق التقليدية لهذا الغرض كطريقة تخمين التربيغات الأصغر ((Least Squares Estimation (LSE)) وطريقة تخمين الإمكانية القصوى ((Maximum Likelihood Estimation (MLE)) [3]، إنطلاقاً من هذا المبدأ أصبح من الضروري إيجاد طرائق أخرى لتخمين الوثوقية للمعادلات غير الخطية فكان التوجه لخوارزميات ما وراء الحدس (تتضمن: الخوارزميات التطورية، خوارزميات أساسها الظواهر الفيزيائية وخوارزميات ذكاء السرب) والتي لاقت شعبية كبيرة في هذا المجال [4].

في هذا العمل ، سنستعمل خوارزمية من خوارزميات ذكاء السرب وهي خوارزمية تلقيح الزهرة (FPA) لتقدير معلمات الـ (SRGMS)، ثم تهجين الـ (FPA) مع الخوارزمية الجينية المشفرة بقيم حقيقية (RGA) لتحسين أداء التخمين.

1. نماذج نمو وثوقية البرمجيات

في منتصف السبعينيات ،حققت الدراسات في نماذج نمو وثوقية البرمجيات تقدماً كبيراً ، وقد تم بالفعل استعمال العديد من نماذج الوثوقية في هذا المجال [3].

ان الهدف من الـ (SRGMS) هو التنبؤ بسلوك الفشل المستقبلي ، والتي تستعمل الوقت بين حالات الفشل أو عدد الفشل الملاحظ في فترة زمنية محددة كبيانات لها [5].

هنالك العديد من نماذج نمو وثوقية البرمجيات لكن تعد نماذج عملية بواسون غير المتجانسة (Non-Homogeneous Poisson Process (NHPP)) هي الأكثر دقة والاکثر استعمالاً وهي من النماذج غير الخطية [6].

هذه النماذج تساعد مهندسي البرنامج لإتخاذ القرار ما إذا وصلت وثوقية البرنامج إلى حد مقبول وتحديد متى يكون النظام جاهزاً للتسليم [7]. تقترض نماذج الـ NHPP بأن عدد العيوب المكتشفة خلال الوقت (t) يتبع (NHPP) بدالة قيمة المتوسط $\mu(t)$. ويؤدي إشتقاق دالة قيمة المتوسط إلى $\lambda(t)$ والتي هي كثافة الفشل للبرنامج التي نقل كلما تم إكتشاف وإزالة العيوب [8] .

هناك العديد من نماذج الـ (NHPP) استعملنا اربعة منها وهي نفسها المستعملة في الدراسات السابقة التي سيتم المقارنة معها وكما موضح في الجدول ادناه:

الجدول (1) : نماذج الـ SRGM المستعملة في البحث.

المعلومات المخزنة	دالة قيمة المتوسط	وصف النموذج	النموذج
a , b	$\mu(t) = a(1 - e^{-bt})$	تم تقديم هذا النموذج من قبل العالمين Goel و Okumoto عام 1978 ويسمى أيضاً نموذج NHPP الأسّي (Exponential NHPP Model) [8].	G-Om
a , b	$\mu(t) = at^b$	نموذج القوة (power) يعد من أقدم النماذج واقترح من قبل العالم Duane عام 1964 ويسمى النموذج بإسمه أحياناً ويعد طريقة رسومية ليؤدي تحليل بيانات نمو الوثوقية وهو بسيط ومباشر للفهم [9].	POWm
a , b	$\mu(t) = a(1 - (1 + bt)e^{-bt})$	نموذج (Delayed S-Shaped model) اقترح من قبل العالم Yamada عام 1983 [8].	DSSm
a , b and c	$\mu(t) = \frac{a(1 - e^{-bt})}{1 + ce^{-bt}}$	(Inflection S-Shaped model) اقترح من قبل العالم Ohba سنة 1984 اذ يفترض إن معدل إكتشاف العيب يزداد بزيادة الإختبار [10].	INFSm

إذ إن :

$a > 0$: تمثل التخمين الأولي للعدد الكلي لحالات الفشل الذي سيكتشف في نهاية الإختبار.

$b > 0$: تمثل كثافة الفشل للعيب.

t : تمثل وقت حصول الفشل.

$c > 0$: تمثل معلمة الإلتواء وتحسب من المعادلة :

$$c = \frac{1-r}{r} \dots \dots \dots (1)$$

اذ ان $0 < r < 1$: تمثل معدل الإلتواء .

الجانب العملي

سنتطرق في هذا الجانب الى الخوارزميات التي تمت المقارنة معها اضافة الى الخوارزميات المستعملة في هذا العمل وتفاصيل كل منهم.

1- الخوارزميات التي تمت المقارنة معها:

1-1 الخوارزميات ABC, PSO و DABC

تم استعمال خوارزمية (PSO) من قبل الباحث [11] Sheta اما خوارزميتي ABC و DABC فقد استعملت من قبل Ajith Abraham وآخرون [12] لتخمين معلمات نماذج وثوقية البرمجيات.

1-1-1 البيانات المستعملة:

تم استعمال مجموعة البيانات التي تم الحصول عليها من قياس فشل النظام أثناء الاختبار في تطبيق الزمن الحقيقي والتي تحوي على 111 قياساً والجدول الخاص بهذه البيانات يحوي على عدد حالات الفشل المقاسة التي تم قراءتها (X_i) ، وعدد الفشل التراكمي (m_i) وعدد عمال الإختبار (tw_i) المشاركين في عملية الإختبار، ولقد تم قياس مجموعة البيانات هذه بالأيام والتي عددها (111) يوم. تم استعمال 70% من البيانات للتدريب و 30% من البيانات للإختبار وسيتم تسمية هذه البيانات بالبيانات الأولى.

1-1-2 دالة اللياقة:

تم استعمال دالة الجذر متوسط مربع الخطأ (RMSE) لإعطاء قيمة اللياقة لكل حل وتم استعمالها لغرض المقارنة النهائية أيضاً، فكلما كانت قيمتها أقل كان الحل المرشح أفضل.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (m_i - \mu_i)^2} \dots\dots\dots(2)$$

إذ إن:

N : تمثل عدد القياسات المستعملة لتخمين معلمات النموذج.

m_i : تمثل عدد الفشل الملاحظ التراكمي.

μ_i : تمثل عدد الفشل المتوقع التراكمي.

1-1-3 النماذج المستعملة:

تم استعمال ثلاثة نماذج وهي: (G-O), (POW), و(DSS).

1-2 خوارزميات CGA و MGA :

تم استعمال الخوارزمية الجينية المطورة (MGA) من قبل الباحثين Chao-Jung Hsu

و Chin-Yu Huang [13] ومقارنتها بالخوارزمية الجينية التقليدية (CGA) التي إستعملها الباحثان Minohara و Tohma.

1-2-1 البيانات المستعملة :

تم استعمال ثلاثة أنواع من البيانات (Data set) وكما مبين في الجدول أدناه.

الجدول (2): البيانات المستعملة في الخوارزميات CGA و MGA .

وصف البيانات	عدد حالات الفشل	عدد أسابيع الإختبار	مجموعة البيانات
البيانات المجمعّة من إختبار تطبيقات اللغات البرمجية (PL/I (database application	328	19 إسبوع	DS1
بيانات إختبار الإصدار الأول لنظام برمجي من أربع إصدارات منتجة من شركة الحاسوب	100	20 إسبوع	DS2
بيانات إختبار الشبكة اللاسلكية	181	34 إسبوع	DS3

إذ يحوي كل جدول على إسبوع الإختبار وعدد الفشل التراكمي (m_i). وسيتم تسمية هذه البيانات بمجموعة البيانات الثانية.

1-2-2 دالة اللياقة:

تم استعمال دالة لياقة مستندة على طريقة تخمين التربيعات الأصغر (Least Squares Estimation (LSE) لحساب لياقة كل كروموسوم وكلما كانت قيمتها أكبر كان الكروموسوم المرشح أفضل.

$$\text{fitness}_{\text{LSE}} = \frac{1}{\sum_{i=1}^N (m_i - \mu_i)^2} \dots\dots\dots(3)$$

وتم استعمال متوسط مربع الخطأ ((Mean Square Error (MSE) لغرض المقارنة النهائية والتي كلما كانت أقل كانت أفضل.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (m_i - \mu_i)^2 \dots\dots\dots(4)$$

إذ إن:

N : تمثل عدد القياسات المستعملة لتخمين معاملات النموذج.

m_i : تمثل عدد الفشل الملاحظ التراكمي.

μ_i : تمثل عدد الفشل المتوقع التراكمي.

1-2-3 النماذج المستعملة:

تم استعمال نموذجين (EXP) و (INFS).

2- الخوارزميات المستعملة في البحث:

Flower Pollination Algorithm(FPA)

1-2 خوارزمية تلقيح الزهرة

تعد خوارزمية تلقيح الزهرة إحدى الخوارزميات المستوحاة من الطبيعة والتي إستلهمت من عملية تلقيح الزهور لإعادة إنتاجها من جديد. وأوجدت من قبل العالم Yang سنة 2012 [14].

يمكن أن يأخذ التلقيح شكلين رئيسيين: حيوي (Biotic) وغير حيوي (Abiotic)، حوالي 90% من النباتات المزهرة تنتمي إلى التلقيح الحيوي إذ ينتشر غبار الطلع بواسطة ملقحات كالحشرات والحيوانات، و حوالي 10% من التلقيح يكون غير حيوي أي لا يحتاج ملقحات بل تقوم الرياح والمياه بالمساعدة في التلقيح.

يتم التلقيح بطريقتين: التلقيح الذاتي (Self-Pollination): يحصل عندما حبوب لقاح من زهرة تلقح الزهرة نفسها أو زهرة أخرى على النبات نفسه. التلقيح الخلطي (Cross-Pollination): يحصل عندما تنتقل حبوب لقاح من زهرة على نبات إلى زهرة أخرى من نبات آخر [15].

Rules Of FPA

1-1-2 قواعد خوارزمية FPA

يمكن أن تُوصف ثبوتية الزهرة وسلوك الملقحات في عملية التلقيح بالقواعد الأربع الآتية [14]:

- 1- التلقيح الخلطي الحيوي يعد عملية تلقيح شاملة (Global Pollination) إذ تؤدي الملقحات فيها رحلة ليفي.
- 2- التلقيح الذاتي غير الحيوي يعد تلقيحاً محلياً (Local Pollination).
- 3- يمكن اعتبار ثبوتية الزهرة بأنها احتمالية إعادة الإنتاج والتي تتناسب حسب تشابه إثنين من الزهور المعنية.
- 4- التبديل أو التفاعل بين التلقيح الشامل والتلقيح المحلي يمكن أن يُسيطر عليه من خلال احتمالية التبديل $p \in [0,1]$.

2-1-2 التمثيل الرياضي للتلقيح الشامل والتلقيح المحلي

Mathematical Representation Of Global And Local Pollination

يعد التلقيح الشامل والتلقيح المحلي الخطوات الرئيسية لخوارزمية تلقيح الزهرة . ففي التلقيح الشامل تحمل حبوب اللقاح بواسطة الملقحات كالحشرات، وحبوب اللقاح يمكن أن تسافر لمسافات بعيدة لأن الحشرات يمكنها على الأغلب الطيران والتحرك لمدى طويل، لهذا تمثل القاعدة رقم 1 والقاعدة رقم 3 رياضياً كالاتي [14]:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(x_i^t - x_{best}) \dots\dots\dots (5)$$

إذ إن:

x_i^t : الحل في الدورة t.

x_{best} : أفضل حل تم الحصول عليه في الدورة t.

γ : عامل التحجيم للسيطرة على حجم الخطوة.

$L(\lambda)$: معلمة تتبع توزيع ليفي.

x_i^{t+1} : الحل الجديد.

أما في التلقيح المحلي فيكون التلقيح ذاتياً أي بين زهور على النبات نفسه, إذ تمثل القاعدة رقم 2 والقاعدة رقم 3 رياضياً كآلاتي:

$$x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t) \dots\dots\dots (6)$$

إذ إن:

x_i^t : الحل في الدورة t.

x_i^{t+1} : الحل الجديد.

x_j^t, x_k^t : حبوب لقاح من زهور مختلفة على النبات نفسه إذ k, j مؤشرات مختارة عشوائياً.

ϵ : متغير عشوائي يتبع التوزيع المنتظم $U(0,1)$.

3-1-2 السير العشوائي و رحلة ليفي Lévy Flight And Random Walks

يُعرّف السير العشوائي بأنه عملية إتخاذ سلسلة من الخطوات العشوائية الواحدة تلو الأخرى. فلنفرض إن (X_i) تمثل خطوة عشوائية مستمدة من التوزيع العشوائي, وإن (S_N) تمثل مجموع هذه الخطوات المتتالية, فإن (S_N) تشكل سيراً عشوائياً كما في المعادلة التالية:

$$S_N = \sum_{i=1}^N X_i = X_1 + \dots + X_N, \dots\dots\dots (7)$$

ويمكن لحجم الخطوة في السير العشوائي أن يكون ثابتاً أو متفاوتاً [3]. أما رحلة ليفي فنُعرف بأنها سير عشوائي والذي تتبع فيه حجم الخطوة توزيع ليفي [16]. تعد L المعلمة التي تمثل قوة حبوب اللقاح, وبما أن الحشرات تنتقل لمسافات طويلة فيتم استعمال رحلة ليفي لتقليد هذه الظاهرة إذ تؤخذ $L > 0$ وتتبع توزيع ليفي [15]:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}} \dots\dots\dots (8)$$

إذ إن:

~: تشير إلى أن L تتبع توزيع ليفي.

$\Gamma(\lambda)$: دالة كاما القياسية, $\lambda = 1.5$

S: حجم الخطوة وتحسب من المعادلة الآتية:

$$s = \frac{U}{|V|^{1/\lambda}} \dots\dots\dots (9)$$

$U \sim N(0, \sigma^2)$: إذ إن U تتبع التوزيع الطبيعي بمتوسط حسابي مساوي للصفر وإنحراف معياري مساوي ل σ^2 .

$V \sim N(0,1)$: إذ إن V تتبع التوزيع الطبيعي بمتوسط حسابي مساوي للصفر وإنحراف معياري مساوٍ للواحد.

ويتم حساب σ^2 من المعادلة الآتية:

$$\sigma^2 = \left[\frac{\Gamma(1+\lambda)}{\lambda \Gamma((1+\lambda)/2)} \cdot \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}} \right]^{1/\lambda} \dots\dots\dots (10)$$

Algorithm Steps

4-1-2 خطوات الخوارزمية

يتم اتباع الخطوات التالية [14]:

الخطوة الأولى: توليد المجتمع الأولي عشوائياً لـ (n) من الزهور باستعمال المعادلة الاتية [12]:

$$\bar{X} = \min + \text{rand} * (\max - \min) \dots \dots \dots (11)$$

إذ إن:

min : الحد الأدنى لفضاء البحث.

max : الحد الأعلى لفضاء البحث.

rand : عدد عشوائي ضمن الفترة [0,1].

وتحديد دالة الهدف $f(x), x = (x_1, x_2, \dots, x_n)$ المراد إيجاد أفضل حل لها، تحديد عدد الدورات الكلية

(MCN)، تحديد عداد الدورات $\text{cycle}=0$ ، وتحديد احتمالية التبديل $p \in [0,1]$.

الخطوة الثانية: حساب دالة الهدف لجميع الزهور ويُؤخذ أفضل حل وليكن (x_{best}) .

الخطوة الثالثة: فحص معيار التوقف، طالما عداد الدورات cycle أقل من عدد الدورات الكلية (MCN) إنتقل

إلى الخطوة الرابعة وإلا توقف.

الخطوة الرابعة: لكل زهرة يتم توليد رقم عشوائي وليكن r يتبع التوزيع المنتظم $U(0,1)$.

الخطوة الخامسة: إذا كانت $r < p$ يتم التلقيح الشامل والذي يتبع توزيع ليفي باستعمال المعادلة (5) وإلا فيتم

التلقيح المحلي والذي يتبع التوزيع المنتظم باستعمال المعادلة (6).

الخطوة السادسة: حساب دالة الهدف للحل الجديد ولتكن f_j وتتم مقارنتها بدالة الهدف للحل السابق f_i إذا كانت

f_j أفضل من f_i يتم استبدال الحل السابق بالحل الجديد $x_i = x_j$ وإلا يبقى الحل السابق x_i

كما هو.

الخطوة السابعة: إيجاد أفضل حل جديد وليكن $(x_{\text{new-best}})$ ومقارنته مع (x_{best}) إذا كانت قيمته أفضل فيتم

تحديث أفضل حل $(x_{\text{best}} = x_{\text{new-best}})$ وإذا كان أسوأ يبقى (x_{best}) كما هو.

الخطوة الثامنة: تحديث عداد الدورات $\text{cycle}=\text{cycle}+1$ والانتقال إلى الخطوة الثالثة.

الخطوة التاسعة: x_{best} يعتبر الحل الأمثل.

Genetic Algorithm(GA)

2-2 الخوارزمية الجينية

تعد الخوارزمية الجينية تقنية بحث دُرست من قبل العالم John Holland سنة 1970. لها القدرة

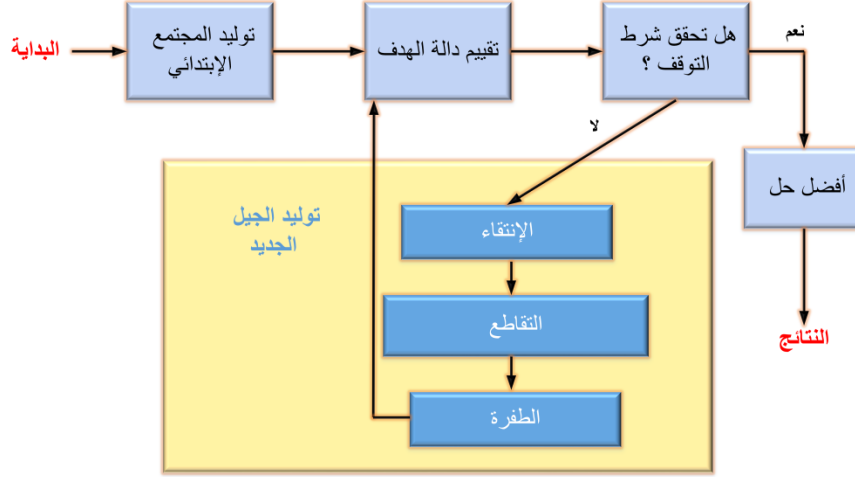
على حل مسائل الأمثلية المعقدة خلال العقود الثلاثة المنصرمة إستعملت الخوارزمية الجينية كخوارزمية تكيفية

لحل مسائل عملية، وقد إستعملت بشكل واسع في العديد من المجالات العلمية [17].

GA Cycle

1-2-2 دورة أو مراحل الـ GA

يوضح الشكل (1) المراحل الرئيسية للخوارزمية الجينية [6].



الشكل (1) مراحل الخوارزمية الجينية [18]

للحصول على أفضل الحلول للتطبيق باستعمال الـ (GA) تتبع الخطوات التالية:

Generate Initial Population

1- توليد المجتمع الابتدائي

يتم إنشاء المجتمع الأولي \vec{X} باستعمال المعادلة رقم (11).

Evaluation

التقييم

2-

يتم التقييم بتعريف دالة اللياقة لكل كروموسوم. إن دالة اللياقة هذه هي مؤشر للكروموسوم والتي توضح مدى قرب هذا الكروموسوم من الحل المطلوب [13].

Termination Condition

3- شرط التوقف

يفحص شرط التوقف بعد كل جيل لتحديد ما إذا كانت ستستمر الخوارزمية أم تتوقف [18] بالنسبة للبيانات الأولى إذا كان الجيل الحالي G أكبر من G_{max} فيتم التوقف، وبالنسبة لمجموعة البيانات الثانية يتم فحص دالة اللياقة كل 10 دورات إذا لم تتغير يتم التوقف.

Generate New Population

4- توليد الجيل الجديد

تتمثل عملية توليد الجيل الجديد بالخطوات الثلاث الآتية والمستعملة في هذا البحث [6]:

Selection

1- الإختيار أو الإنتقاء

هي عملية إختيار الأفراد من أجل التزاوج وإنتاج جيل جديد ويتم إختيار الأفراد الأكثر صلاحية (أفضل دالة لياقة) لإعادة الإنتاج.

• -: Top- Mate Selection

يتم إختيار الأب الأول الذي له أفضل دالة لياقة أما الأب الثاني فيتم إختياره عشوائياً.

2- التقاطع أو التزاوج Crossover

بعد إختيار الآباء بإحدى طرائق الإنتقاء تتم عملية التقاطع وتسمى أيضاً بإعادة الإنتاج. يتم تقاطع الآباء لإنتاج الأبناء (الجيل الجديد), الفكرة من التقاطع إن الكروموسوم الإبن الناتج يكون أفضل من الأبوين إذا أخذ الصفات الجيدة من كليهما . تردد التقاطع يتم التحكم به عن طريق احتمالية التقاطع (Crossover Probability (pc) وهناك عدة أنواع من التقاطع ولكل نوع تشفير هنالك نوع مناسب من التقاطع . من أنواع التقاطع:

• التقاطع الحدسي (Heuristic Crossover):

يستعمل مع التشفير الحقيقي إذ يعتمد على قيمة اللياقة للأبوين لإنتاج الأبناء, يتم توليد إبن واحد فقط أما الإبن الآخر فهو عبارة عن تمرير الأب ذو اللياقة الأفضل إلى الجيل الجديد دون أي معالجة, والمعادلات الآتية توضح هذا النوع من التقاطع :

fitness_{parent1} is better than fitness_{parent2}

$$\text{offspring}_1 = \text{parent}_1 \dots\dots\dots (12)$$

$$\text{offspring}_2 = \text{parent}_1 \pm r * (\text{parent}_1 - \text{parent}_2) \dots\dots\dots (13)$$

إذ ان:

r رقم عشوائي يتبع التوزيع المنتظم (0,1) .U

3- الطفرة

Mutation

تتم عملية الطفرة بعد مرحلة التقاطع, وهي عبارة عن تغيير في قيمة الجين الواحد أو أكثر للكروموسوم بالإعتماد على إحتتمالية الطفرة (Mutation Probability (pm) والتي تكون صغيرة عادةً. عملية الطفرة تحافظ على التنوع في الجيل إذ تؤدي إلى إنتاج حلول جديدة متنوعة وهذا يزيد من سرعة الإقتراب إلى الحل الأمثل. من أنواع الطفرات:

• الطفرة غير المنتظمة (Non-Uniform Mutation): يستعمل في التشفير الحقيقي إذ قيمة

الكروموسوم الأب تتغير في نطاق محدود بالنظر إلى رقم الجيل الحالي, فإذا كان رقم الجيل الحالي صغيراً يكون مدى التغيير للكروموسوم كبيراً وبينما يزداد رقم الجيل يقل مدى التغيير.

$$\text{offspring}_i = \left\{ \begin{array}{l} \text{parent}_i + (\text{upperbound}_i - \text{parent}_i) * f(G) \\ \text{or} \\ \text{parent}_i - (\text{parent}_i - \text{lowerbound}_i) * f(G) \end{array} \right\} \dots(14)$$

إذ إن:

upperbound_i : الحد الأعلى للكروموسوم

lowerbound_i : الحد الأدنى للكروموسوم

f(G) : دالة مدى التغيير والتي تحسب من المعادلة:

$$f(G) = \left(r * \left(1 - \frac{G}{G_{\max}} \right) \right)^b \dots\dots\dots (15)$$

إذ إن :

- r : عدد عشوائي يتبع التوزيع المنتظم U (0,1).
- G_{max} : الحد الأقصى لعدد الاجيال.
- G : الجيل الحالي.
- b : معلمة الحالة والتي تحدد درجة الإعتمادية على الجيل الحالي.

Algorithm Steps

2-2-2 خطوات الخوارزمية

يتم اتباع الخطوات التالية [19]:

- الخطوة الأولى: توليد جيل أولي من الكروموسومات عشوائياً بالحجم (N) باستعمال المعادلة (11) , وتحديد دالة الهدف $f(x), x = (x_1, x_2, \dots, x_n)$ المراد إيجاد أفضل حل لها, تحديد الحد الأقصى لعدد الاجيال G_{\max} , تحديد الجيل الحالي $G = 0$, وتحديد إحتمالية التقاطع (pc) وإحتمالية الطفرة (pm).
- الخطوة الثانية: حساب دالة الهدف لجميع الكروموسومات.
- الخطوة الثالثة: فحص معيار التوقف, طالما لم يتحقق شرط التوقف إنتقل إلى الخطوة الرابعة وإلا توقف.
- الخطوة الرابعة: يتم إختيار أبوين لإنتاج الأبناء باستعمال طريقة (Top-Mate Selection).
- الخطوة الخامسة: توليد عدد عشوائي $pcross$, إذا كانت $pcross < pc$ يتم التقاطع باستعمال طريقة التقاطع الحدسي.
- الخطوة السادسة: توليد عدد عشوائي $pmut$, إذا كانت $pmut < pm$ تتم الطفرة باستعمال الطفرة غير المنتظمة.
- الخطوة السابعة : وضع الكروموسومات الناتجة (الأبناء) في الجيل الجديد.
- الخطوة الثامنة: طالما حجم الجيل الجديد أقل من N, إنتقل إلى الخطوة الرابعة وإلا إنتقل إلى الخطوة التاسعة.
- الخطوة التاسعة: إبدال الجيل السابق بالجيل الحالي وتحديث الجيل الحالي $G = G + 1$ وإلإنتقال إلى الخطوة الثانية.

Proposed Hybrid Algorithm

3-2 الخوارزمية المهجنة المقترحة

لغرض تحسين اداء خوارزمية تلقيح الزهرة تم تهجينها في هذا العمل مع الخوارزمية الجينية المشفرة بقيم حقيقية. ان الهدف من الخوارزمية المقترحة تسريع عملية الوصول للحل الامثل إذ يكون المجتمع الأولي لخوارزمية الزهرة هو المجتمع الذي نتج من الخوارزمية الجينية والأقرب للأمثلية, بمعنى آخر يتم تنفيذ الخوارزمية الجينية أولاً والمجتمع الذي يتم الحصول عليه عند تحقق شرط التوقف يُعَيّن كمجتمع إبتدائي لخوارزمية الزهرة ثم تنفذ الخوارزمية للحصول على أفضل حل.

النتائج والمناقشة

النتائج عند استعمال البيانات الأولى:

1- نتائج تنفيذ خوارزمية FPA:

تمت مقارنة نتائج تنفيذ خوارزمية الـ (FPA) للبيانات الأولى و بنسبة 70% للتدريب و30% للاختبار مع النتائج المذكورة في كل من [11] و [12]. الجدول (3) يوضح نتائج تنفيذ الخوارزمية مع أفضل قيم للمعلمات المخمنة.

الجدول (3): مقارنة نتائج FPA مع نتائج PSO, ABC, DABC للبيانات الأولى

Model	RMSE-testing (30%of data)				Best values by FPA	
	PSO	ABC	DABC	FPA	A	b
G_O	80.8963	119.6425	72.0187	77.8622	684.3160	0.0174
POW	149.9684	158.6753	81.9231	146.3812	22.3832	0.7281
DSS	17.0638	17.091	29.8051	16.6277	501.8481	0.0636

إذ تفوقت خوارزمية (FPA) على خوارزمية (PSO) و (ABC) لجميع النماذج وتفوقت على (DABC) في نموذج الـ (DSS) فقط. وقد تم تحديد أفضل القيم بالخط الغامق.

2- نتائج تنفيذ خوارزمية تلقيح الزهرة المهجنة HFPA:

تم تنفيذ الخوارزمية المهجنة للبيانات الأولى ولجميع النماذج ومقارنة النتائج مع خوارزمية (FPA) الاعتيادية وكما هو ملاحظ من الجدول (4) إن (HFPA) توصلت تقريباً إلى قيمة (RMSE) نفسها بعدد دورات أقل بكثير مما إحتاجته خوارزمية (FPA). إذ تم الحصول على عدد الدورات الكلية لخوارزمية (HFPA) من خلال جمع عدد دورات الـ (RGA) مع عدد دورات الـ (FPA).

الجدول (4): مقارنة نتائج HFPA مع نتائج FPA للبيانات الأولى

Model	FPA		Hybrid FPA				Improvement
	RMSE-testing	No. Of cycles	RMSE-testing	No. Of cycles by RGA	No. Of cycles by FPA	Total no. of cycles by HFPA	
G_O	77.862	508	77.859	50	316	366	27.95%
POW	146.381	591	146.785	80	491	571	3.38%
DSS	16.627	480	16.627	40	197	237	50.62%

النتائج عند استعمال مجموعة البيانات الثانية:

1- نتائج تنفيذ خوارزمية FPA:

لكل نموذج من البيانات الثلاثة تم تنفيذ كل خوارزمية 100 مرة ثم إختيار أقل (MSE) تم الحصول عليها ثم يتم حساب معدل الدورات اللازمة للوصول إلى أقل (MSE).

الجدول (5) مقارنة نتائج FPA مع نتائج CGA, MGA لمجموعة البيانات الثانية

Data-set	Model	CGA		MGA		FPA		Best values found by FPA		
		MSE	Av. No. Of gen.	MSE	Av. No. Of gen.	MSE	Av. No. Of gen.	A	b	C
DS1	G_O	139.815	13415	139.815	5341	139.815	759	760.534	0.032	-
	INFS	82.704	78664	82.704	15664	82.704	809	382.373	0.178	2.886
DS2	G_O	11.617	48104	11.617	7331	11.617	299	130.201	0.083	-
	INFS	8.98	77889	8.98	17490	8.980	616	110.631	0.173	1.221
DS3	G_O	22.863	39209	22.863	3350	20.391	325	1000	0.006	-
	INFS	5.82	83578	5.82	20749	5.820	889	229.198	0.087	3.703

2- نتائج تنفيذ خوارزمية تلقيح الزهرة المهجنة HFPA:

تم تنفيذ الخوارزمية المهجنة لمجموعة البيانات الثانية ولجميع النماذج إذ تم التنفيذ لـ 100 مرة وفي كل تنفيذ تم الحصول على قيمة (MSE) وعلى عدد الدورات اللازمة للوصول الى هذه القيمة (عدد الدورات لـ RGA و عدد الدورات لـ FPA) ثم يتم جمع عدد الدورات للحصول على عدد الدورات الكلية اللازمة من قبل الخوارزمية المهجنة للوصول الى قيمة (MSE). بعد انتهاء جميع التنفيذات يتم إختيار أقل (MSE) تم الحصول عليها ثم يتم حساب معدل الدورات الكلية اللازمة للوصول إلى أقل (MSE). تمت مقارنة النتائج مع خوارزمية (FPA) الإعتيادية وكما هو ملاحظ من الجدول (6) إن (HFPA) توصلت تقريباً إلى نفس قيمة (MSE) بمعدل دورات أقل بكثير مما إحتاجته خوارزمية (FPA).

الجدول (6) مقارنة نتائج HFPA مع نتائج FPA لمجموعة البيانات الثانية

Data set	Model	FPA		HFPA				Improve-ment
		MSE-testing	Av. No. Of gen.	MSE-testing	No. Of cycles by RGA	No. Of cycles by FPA	Av. No. Of gen. by HFPA	
DS1	G-O	139.815	759	139.815	60	382	527	30.56%
	INFS	82.704	809	82.704	40	240	419	48.20%
DS2	G-O	11.617	299	11.617	20	46	86	71.23%
	INFS	8.980	616	8.979	20	138	152	75.32%
DS3	G-O	20.391	325	20.429	20	6	23	92.92%
	INFS	5.820	889	5.820	20	105	222	75.02%

وفيما يأتي المقطع البرمجي الخاص بخوارزمية تلقيح الزهرة FPA:

```

clc;
mina=-1000;          maxa=1000;
minb=-1;            maxb=1;
m1=xlsread(FileName1); m2=xlsread(FileName2);
l1=length(m1);      l2=length(m2);

%__model__
f=inline('a*(1-exp(-b*t))','a','b','t');
%f=inline('a*(t^b)','a','b','t');
%f=inline('a*(1-(1+(b*t))*exp(-b*t))','a','b','t');
n=20; p=0.8; gen=1000; d=2;
for k=1:n
    flo(k,1)=rand*(maxa-mina)+mina;
    flo(k,2)=rand*(maxb-minb)+minb;
end
%__RMSE training__
for i=1:n
    su=0;
    for k=1:l1
        u=f(flo(i,1),flo(i,2),k);
        su=su+(m1(k)-u)^2;
    end
    fitness(i)=sqrt(su/l1);
end
[fmin,I]=min(fitness);
best=flo(I,:);
nflo=flo;
itr=1;
while itr<=gen
    for i=1:n
        if rand<p
            beta=3/2;

sigma=(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*beta*2^((beta-1)/2)))^(1/beta);
            u=randn(1,d)*sigma;
            v=randn(1,d);
            step=u./abs(v).^(1/beta);
            L=0.01*step;
            dS=L.*(flo(i,:)-best);
            nflo(i,:)=flo(i,:)+dS;
        else
            epsilon=rand;
            JK=randperm(n);
            nflo(i,:)=nflo(i,:)+epsilon*(flo(JK(1),:)-
flo(JK(2),:));
        end %if
        su=0;
        for k=1:l1
            u=f(nflo(i,1),nflo(i,2),k);
            su=su+(m1(k)-u)^2;
        end
        Fnew=sqrt(su/l1);
        if (Fnew<=fitness(i)),
            flo(i,:)=nflo(i,:);
            fitness(i)=Fnew;
        end
        if Fnew<=fmin,

```

```

        best=nflo(i,:);
        fmin=Fnew ;
    end
end %for i=1:n
itr=itr+1;
end %while
su=0;
for k=1:l2
    u=f(best(1),best(2),k+78);
    su=su+(m2(k)-u)^2;
end
RMSEtesting=sqrt(su/l2);

```

المقطع البرمجي الخاص بخوارزمية الزهرة المهجنة HFPA كالآتي:

```

%##### HFPA #####
[ch,loRGA,hoh]=hh(Filename1,Filename2)
mina=-1000; maxa=1000;
minb=-1;    maxb=1;
m1=xlsread(Filename1); m2=xlsread(Filename2);
l1=length(m1);    l2=length(m2);
%__model__
f=inline('a*(1-exp(-b*t))','a','b','t');
% f=inline('a*(t^b)','a','b','t');
% f=inline('a*(1-(1+(b*t))*exp(-b*t))','a','b','t');
n=20; p=0.8; gen=1000; d=2;
flo=ch;
arrf=zeros(1,gen);
%__RMSE training__
for i=1:n
    su=0;
    for k=1:l1
        u=f(flo(i,1),flo(i,2),k);
        su=su+(m1(k)-u)^2;
    end
    fitness(i)=sqrt(su/l1);
end
[fmin,I]=min(fitness);
best=flo(I,:);
nflo=flo;
itr=1;
while itr<=gen
    for i=1:n
        if rand<p
            beta=3/2;
            sigma=(gamma(1+beta)*sin(pi*beta/2)/(gamma((1+beta)/2)*beta*2^((beta-1)/2)))^(1/beta);
            u=randn(1,d)*sigma;
            v=randn(1,d);
            step=u./abs(v).^(1/beta);
            L=0.01*step;
            dS=L.*(flo(i,:)-best);
            nflo(i,:)=flo(i,:)+dS;
        else
            epsilon=rand;
            JK=randperm(n);
            nflo(i,:)=nflo(i,:)+epsilon*(flo(JK(1),:)-flo(JK(2),:));
        end
    end
end

```

```

end %if
su=0;
for k=1:l1
    u=f(nflo(i,1),nflo(i,2),k);
    su=su+(m1(k)-u)^2;
end
Fnew=sqrt(su/l1);
if (Fnew<=fitness(i)),
    flo(i,:)=nflo(i,:);
    fitness(i)=Fnew;
end
if Fnew<=fmin,
    best=nflo(i,:);
    fmin=Fnew ;
end
end %for i=1:n
itr=itr+1;
end %while
su=0;
for k=1:l2
    u=f(best(1),best(2),k+78);
    su=su+(m2(k)-u)^2;
end
RMSEtesting=sqrt(su/l2);

%##### GA #####
function [ch,loRGA,hoh]=hh(Filename1,Filename2)
mina=-1000; maxa=1000;
minb=-1;   maxb=1;
m1=xlsread(Filename1); m2=xlsread(Filename2);
l1=length(m1);      l2=length(m2);
% model
f=inline('a*(1-exp(-b*t))','a','b','t');
% f=inline('a*(t^b)','a','b','t');
% f=inline('a*(1-(1+(b*t))*exp(-b*t))','a','b','t');

%initial
size=20; gen=1000; po=0;
ch=zeros(size,2);
for k=1:size
    ch(k,1)=rand*(maxa-mina)+mina;
    ch(k,2)=rand*(maxb-minb)+minb;
end
I=1;
%generation
while (I<=gen)
    num=size;
    t=0;
    for j=1:size
        fitt(j)=fitness(ch(j,1),ch(j,2),l1,m1,f);
    end
%selection
    while(num>=2)
        [val seq]=min(fitt);
        p1=ch(seq,:);
        h=randperm(num);
        for e=1:num
            if(h(e)~=seq)
                a=h(e);
                break
            end
        end
    end
end

```

```

        end
        p2=ch(a,:);
        off1=p1;
        off2=p2;
        if(rand<0.5)
%cross
[off1,off2]=crossover(p1,p2);
        end
%mutation
pmut=rand;
        if(pmut<0.1)
            off=mutation(off2,I,gen,maxa,mina,maxb,minb);
        end
%store
        t=t+1;
        newch(t,:)=off1;
        newfitt(t)=fitness(off1(1,1),off1(1,2),l1,m1,f);
        t=t+1;
        newch(t,:)=off2;
        newfitt(t)=fitness(off2(1,1),off2(1,2),l1,m1,f);
%delete from ch and fitt
dch=inf(size,2);
dfitt=inf(size,1);
        z=0;
        for fd=1:num
            if((fd~=a)&&(fd~=seq))
                z=z+1;
                dch(z,:)=ch(fd,:);
                dfitt(z)=fitt(fd);
            end
        end
        ch=zeros(size,2);
        fitt=zeros(size,1);
        ch=dch;
        fitt=dfitt;
        num=num-2;
    end %end while(num>=2)
    ch=newch;
    fitt=newfitt;
    newch=zeros(size,2);
    newfitt=zeros(size,1);
    if round(I/10)==I/10,
        [va ya]=min(fitt);
        po=po+1;
        gg(po)=va;
        if po>1
            z1=abs(va-gg(po-1));
            if (z1<0.0001)
                break
            end
        end
    end
    end
    I=I+1;
end %end while(I<=gen)
[val y]=min(fitt);

%_____RMSE testing_____
rmse=RMSEF(ch(y,1),ch(y,2),l2,m2,f);

%_____crossover_____
function [off1,off2]=crossover(p1,p2)

```

```

off1=p1;
off2=p1-rand*(p1-p2);

%_____mutation_____
function off=mutation(off,I,gen,maxa,mina,maxb,minb)
    fg=(rand*(1-(I/gen)))^5;
    off(1,1)=off(1,1)+(maxa-off(1,1))*fg;
    off(1,2)=off(1,2)-(off(1,2)-minb)*fg;

%_____fitness_____
function fitt=fitness(a,b,l1,m1,f)
    su=0;
    for k=1:l1
        u=f(a,b,k);
        su=su+(m1(k)-u)^2;
    end
    fitt=sqrt(su/l1);

%_____RMSEF_____
function rmse=RMSEF(a,b,l2,m2,f)
    su=0;
    for k=1:l2
        u=f(a,b,k+78);
        su=su+(m2(k)-u)^2;
    end
    rmse=sqrt(su/l2);

```

الاستنتاجات والتوصيات

هناك عدة طرائق تقليدية لتخمين معاملات نماذج الـ (SRGMs) لكنها تعاني من عدة مشاكل، لذلك استخدمنا طرائق الذكاء الاصطناعي للتخمين، إذ تم استعمال خوارزمية من خوارزميات السرب هي خوارزمية تلقح الزهرة، وكذلك تم استعمال خوارزمية تطويرية وهي الخوارزمية الجينية الحقيقية، ثم تم تهجين الخوارزمية الجينية مع خوارزمية ذكاء السرب. تمت مقارنة النتائج مع نتائج خوارزميات الـ (PSO,ABC, DABC,CGA,MGA). ويلاحظ من خلال نتائج تنفيذ الخوارزميات احتياج خوارزمية (HFPA) إلى عدد دورات أقل مما إحتاجته خوارزمية (FPA) للوصول إلى أفضل حل، إذ تناقصت عدد الدورات في الخوارزمية المهجنة بالنسب المذكورة في الجدولين (4) و (6).

المصادر

- [1] Sheakh, T. H. and Singh, V. P., "Taxonomical Study Of Software Reliability Growth Models", International Journal of Scientific and Research Publications, Volume 2, Issue 5, pp.1-3 (2012).
- [2] Wood, A., "Software Reliability Growth Models", Tandem Tech., Technical Report, Vol. 96.1, Tandem Computers Inc., Corporate Information Center, Cupertino Calif., Part Number 130056 (1996).
- [3] Al-Saati, N. A. and Abd-Alkareem, M., " The Use of Cuckoo Search in Estimating the Parameters of Software Reliability Growth Models", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 11, No. 6, (2013).
- [4] Mirjalili, S. A. and Mirjalili , S. M., Lewis, A., "Grey Wolf Optimizer", Advances in Engineering Software 69, pp. 46–61(2014).
- [5] Kelanibandara, K.W.K.B.P.L.M., "Software Reliability Estimation Using Cubic Splines Network Model", M.Sc. thesis, School of Computing, University of Colombo, (2012).
- [6] Alneamy, J. S. and Dabdoob, M. M., "The Use of Original and Hybrid Grey Wolf Optimizer in Estimating the Parameters of Software Reliability Growth Models", International Journal of Computer Applications, Volume 167-No.3, pp.12-21(2017).
- [7] Lai, R. and Garg, M., "A Detailed Study of NHPP Software Reliability Models", Journal of Software, Vol.7, No.6, pp.1296-1306 (2012).
- [8] Meyfroyt, P. H. A., "Parameter Estimation for Software Reliability Models", M.Sc. thesis, Eindhoven: Technische Universiteit Eindhoven, (2012).
- [9] Shanmugam, L. and Florence, L., "A Comparison of Parameter Best Estimation Method for Software Reliability Models", International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.5, pp.91-102 (2012).
- [10] Prasad, R. S., Rao, K. P. and Mohan, G. K., "Software Reliability using SPRT: Inflection S-shaped Model", International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 6, pp. 349-355 (2013).
- [11] Sheta, A. F., " Parameter Estimation of Software Reliability Growth Models by Particle Swarm Optimization", AIML Journal, Volume (7), Issue (1), pp.55-61(2007).
- [12] Sharma, T.K., Pant, M. and Abraham, A., "Dichotomous Search in ABC and its Application in Parameter Estimation of Software Reliability Growth Models ", Third World Congress on Nature and Biologically Inspired Computing, pp.214-219 (2011).
- [13] Hsu, C.J. and Huang, C.Y., "A Study on the Applicability of Modified Genetic Algorithms for the Parameter Estimation of Software Reliability Modeling " , IEEE 34th Annual Computer Software and Applications Conference, pp.531-540 (2010).

- [14] Yang, X. S., "Flower Pollination Algorithm for Global Optimization", in: Unconventional Computation and Natural Computation, Lecture Notes in Computer Science, Vol. 7445, pp. 240-249 (2013).
- [15] Yang, X. S., Karamanoglu, M. and He, X. S., "Flower Pollination Algorithm: A Novel Approach for Multiobjective Optimization", Engineering Optimization, vol. 46, Issue 9, pp. 1222-1237 (2014).
- [16] Yang, X. S., "Nature Inspired Metaheuristic Algorithms", Second Edition, Luniver Press, pp:1-160 (2010).
- [17] HERRERA, F., LOZANO, M. and VERDEGAY, J.L., "Tackling Real Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis", Artificial Intelligence Review 12: 265–319 (1998).
- [18] AL Neamy, J. S., "Brain Tumors Images Diagnosis Using Hybrid Intelligency Techniques", Ph.D. Thesis, college of computers and mathematics science, University of Mosul (2006).
- [19] MENDES, J. M., "A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop Scheduling Problem", WSEAS TRANSACTIONS on COMPUTERS, Issue 4, Volume 12, pp. 164-173 (2013).