https://doi.org/10.33261/jaaru.2021.28.4.008 *Association of Arab Universities Journal of Engineering Sciences (2021)28(4):85–95*

Association of Arab Universities Journal of Engineering Sciences

مجلة اتحاد الجامعات العربية للدراسات والبحوث الهندسية

كلية الهندسة - جامعة بغداد

جمعية كليات الهندسة اعضاء اتحاد الجامعات العربية

دراسة طرق توليد الأجيال الأولية في البرمجة الجينية القائمة على الأشجار واستخدامها في مجال توليد برامج لحركة روبوت مُتتبع مسارات متعددة ضمن مضمار محدود

عبد الرحمن سلوم [1]، محمود رحال[2]، محمد خطيب [3]

[1] جامعة حلب، حلب، سوريا، abdulrahmansalloum92@gmail.com

[2] كلية الهندسة الكهربائية والالكترونية، جامعة حلب، حلب، سوريا، Mrahhal2000@hotmail.com

[3] كلية الهندسة المعلوماتية، جامعة حلب، حلب، سوريا، mhmd.khatib1970@gmail.com

* الباحث الممثل: عبد الرحمن سلوم، abdulrahmansalloum92@gmail.com

نشر في: 31 كانون الاول 2021

الخلاصة – نهدف في هذا البحث إلى دراسة طرق توليد الأجيال الأولية في البرمجة الجينية القائمة على الأشجار واستخدامها في توليد برنامج تلقائي لحركة روبوت، حيث تشمل هذه الدراسة عدة طرق وهي الطريقة النامية (Grow method) والكاملة (Full method) والمشتركة (Ramped Half and Half) والطريقة الموحدة (Uniform method) وطريقتي الإنشاء الاحتمالي للأشجار (Probabilistic Tree-Creation 1 and 2)، قمنا بتطبيق هذه الطرق في مجال توليد برامج لروبوت مُتتبع مسارات متعددة تحتوي على أهداف، وقمنا بمحاكاة عمل هذا الروبوت على مسائل شهيرة مشابهة له في العمل وهي مسائل النمل الاصطناعي، وهي مسألة مسار (Santa Fe) ومسألة مسار (John Muir) ومسألة مسار (Los Altos)، وإضافة مسارات معدلة لهذا البحث بهدف دراسة تأثير تغير عوامل متعددة، حيث كانت أحجام البرامج تتضخم بشكل كبير عند زيادة عدد الانعطافات أو عدد المسارات أو طول المسار، حيث كان لذلك تأثير سلبي على لياقة البرامج، وعلى عدد مرات التقييم، وعلى نسبة البرامج الصحيحة الناتجة. ووجدنا في هذه الدراسة أن أفضل الطرق من ناحية أحجام البرامج المولدة كانت الطريقة الموحدة (uniform method)، كون أحجام البرامج الصحيحة كانت أقل بنسبة 66% من أحجام البرامج الصحيحة المولدة في الطرق الأخرى، وتليها طرق PTC2 بنسبة 51% وGrow بنسبة 47% وPTC1 بنسبة 45%.

الكلمات الرئيسية – برمجة جينية، توليد أجيال أولية، برامج حركة روبوت.

1. المقدمة

تعتبر البرمجة الجينية إحدى أهم مجالات الحوسبة التطورية وأكثرها انتشارا، كون هذا المجال يقدم العديد من التطبيقات والخدمات، حيث تُستخدم البرمجة الجينية بهدف الحصول على جهاز حاسوب يستطيع القيام بما يجب القيام به، دون أن يخبره أحد بكيفية القيام بذلك. حيث تم استخدامها في توليد برامج في شتّى المجالات منها المجالات الطبية والاعمال والرياضة والألعاب، اما نحن في هذا البحث سنقوم باستخدامها في توليد البرامج الخاصة بتحريك الروبوت.

العديد من الدراسات قامت في مجال استخدام البرمجة الجينية في مجال توليد وتطوير برامج حركة الروبوت، حيث قام الباحثون بإجراء دراسة [6] لتطوير برامج حركة روبوت مقاتل في لعبة، حيث أن البرامج التي تم إنشاؤها أثناء التجارب أظهرت عدة استراتيجيات قتالية مختلفة وفعالة مثل التجنب والقنص والتطويق وإطلاق النار.

أيضاً قام الباحثون بإجراء دراسة [4] للتخطيط لحركة مركبة مستقلة تحت الماء باستخدام البرمجة الجينية، حيث يهدف هذا الروبوت إلى التجول في بيئة، وتحديد كائن بلاستيكي، مثل الزجاجات، والتقاطه واستعادته إلى مكان محدد، حيث تضمن هذا الروبوت حساسات وكاميرة تصوير عميق.

2. أهمية البحث وأهدافه:

الهدف الأول لهذا البحث هو دراسة استخدام طرق متعددة لتوليد الأجيال الأولية في البرمجة الجينية القائمة على الأشجار، كون موضوع توليد الأجيال الأولية يؤثر بشكل كبير -سلباً أو إيجاباً- على الأفراد الناتجة من البرمجة الجينية ومدى قرب الفرد الناتج (لياقته) من الهدف المطلوب، وبالتالي على ناتج تنفيذ البرمجة الجينية بشكل عام.

والهدف الثاني هو تنفيذ الطرق السابقة بهدف توليد برنامج تلقائي لروبوت مُتتبع مسارات متعددة، ضمن مضمار محدد، حيث استخدمنا في هذه الدراسة عدة أنواع من المسارات منها الشهيرة كمسارات النمل الاصطناعي ومسارات معدلة قمنا بتصميمها بهدف دراسة تأثير تغير معاملات مختلفة وهي تغير عدد الانعطافات وتغير عدد المسارات وتغير طول المسار، والنتائج المترتبة على هذه التغيرات على الأداء من ناحية عدد الأفراد الصحيحة والأفراد الممتازة والأفراد الجيدة، وأيضاً النتائج المترتبة على الأداء من ناحية أحجام البرامج الناتجة، وتأثير تغير أحجام البرامج على الأداء الكلي للبرمجة الجينية.

تأتي أهمية هذا البحث كوننا سنعتمد عليه في عملية بناء إطار عمل برمجي غرضي التوجه للبرمجة الجينية لتطوير البرامج التلقائية في مجال التطبيقات الروبوتية، حيث يساعد هذا البحث والأبحاث اللاحقة على توسيع أفق البحث العملي في إحدى المجالات المطلوبة بكثرة حالياً وهو مجال توليد برامج حركة الروبوت بشكل تلقائي، حيث اعتمدنا في دراستنا المرجعية لهذا البحث على عدة مراجع اختصاصية في صلب هذا المجال.

3. الدراسات السابقة لطرق توليد الأجيال الأولية

تعتبر مرحلة توليد الأجيال الأولية من أولى المراحل الواجب تنفيذها عند تطبيق البرمجة الجينية، حيث يمكن أن تكون البداية الجيدة التي يقدمها التجمع الأولي أمرًا مهمًا لنجاح تشغيل البرمجة الجينية. ولذلك تحتوي البرمجة

1726-4081© 2021 The author(s). Published by Association of Arab Universities Journal of Engineering Sciences. This is an open access article under the CC BY-NC license (https://creativecommons.org/licenses/by-nc/4.0/).

الجينية على العديد من أساليب توليد الأجيال الأولية التي تضمن تشكيل مجموعات متنوعة من الأفراد والتي يتم عليها إجراء العمليات الجينية المختلفة، وسنقوم فيما يلي باستعراض طرق توليد التجمع الأولي والتي يتم استخدامها بكثرة في تطبيقات البرمجة الجينية المختلفة.

3.1 الطريقة الكاملة (Full Method) [7] [9]

تعتبر هذه الطريقة من أوائل الطرق المستخدمة في توليد التجمعات الأولية، وفيها يتم إنشاء الأفراد الأولية بحيث لا يتم تجاوز الحد الأقصى للعمق المحدد. حيث أن عمق عقدة ما هي عدد الحواف التي يجب اجتيازها للوصول إلى العقدة المطلوبة بدءاً من عقدة الجذر الخاصة بالشجرة (والتي يُفترض أنها في العمق 0). أما العمق الكلي للشجرة هو عمق أعمق أوراقها. تكون كل الأوراق في هذه الطريقة في نفس العمق، حيث يتم أخذ العقد بشكل عشوائي من مجموعة التوابع (function set) حتى يتم الوصول إلى أقصى عمق للشجرة. ووراء هذا العمق، يتم اختيار العقد الطرفية (terminal) فقط. على الرغم من أن الطريقة الكاملة تولد الأشجار حيث توجد جميع الأوراق في نفس العمق، هذا لا يعني بالضرورة أن جميع الأشجار الأولية سوف يكون لديها عدد مماثل من العقد، وهذا يؤدي إلى أن تكون مجموعة أحجام البرنامج والأشكال التي تنتجها الطريقة الكاملة محدودة إلى حد ما.

3.2 الطريقة النامية (Grow Method): [7][9]

تسمح بإنشاء أشجار بأحجام وأشكال أكثر تنوعًا. حيث يتم اختيار العقد من المجموعة البدائية (primitives) بأكملها (أي التوابع والعقد الطرفية) إلى أن يتم الوصول إلى الحد الأقصى للعمق. وبمجرد الوصول إلى الحد الأقصى للعمق يتم اختيار العقد الطرفية.

3.3 الطريقة المشتركة (Ramped half-and-half) : [5]

تم اقتراح طريقة ثالثة تجمع كلا الطريقتين -الكاملة والنامية- تدعى (ramped half-and-half)، حيث أن نصف التجمع الأولي سيتم إنشاؤه باستخدام الطريقة الكاملة (Full method) والنصف الآخر سيتم إنشاؤه باستخدام الطريقة النامية (grow method)، حيث يتم إنجازه باستخدام حدود للعمق وذلك لضمان توليد أشجار متنوعة الأحجام والأشكال.

3.4 الطريقة الموحدة (Uniform Method):[5] [3]

تم استخدام خوارزمية (Alonso's bijective) لتوليد أشجار موحدة دقيقة، وهي سهلة الاستخدام في التطبيقات العملية، ولكن تم انتقادها لكونها مكلفة حسابيا [8]، قدم لانغدون [10] طريقة التهيئة الموحدة المنحدرة (ramped uniform initialization) والتي تعتبر امتدادا لخوارزمية (Alonso's bijective) عن طريق السماح للمستخدم بتحديد مجموعة من أحجام الأشجار الأولية، ثم ينشئ أعداداً متساوية من الأشجار العشوائية لكل طول في مجال العمل المختار. تتميز هذه الطريقة بكون معظم الأشجار غير متماثلة (متناظرة)، وهذا يختلف تمامًا عن الأشجار الناتجة بطريقة (ramped half-and-half) والتي تقع في المتوسط على مسافة محددة من الجذر. قد يكون أخذ العينات الموحدة (Uniform sampling) أفضل في المشكلات التي تكون فيها الحلول المطلوبة غير متماثلة مع كون بعض الأوراق أكثر أهمية من غيرها.

3.5 طريقة الانشاء الاحتمالي للأشجار 1 (Probabilistic Tree- 1 (Creation):[8] [9]

تعتبر هذه الطريقة تطويرا للطريقة النامية، حيث تقوم على مبدأ إنتاج أشجار حول حجم شجرة متوقع ومحدود، حيث يتم تحديد حجم الشجرة المتوقع والعمق الأقصى المسموح به. ومن ثم تبدأ هذه الطريقة بحساب احتمال اختيار عقدة غير طرفية (من مجموعة التوابع) على عقدة طرفية ويرمز له بالرمز p وذلك بهدف الحفاظ على حجم الشجرة المتوقع E كما يلي:

$$p = \frac{1 - \frac{1}{Etree}}{\sum_{n \in N} q_n \, b_n} \quad (1)$$

معادلة 1: حساب احتمال اختيار عقدة غير طرفية على عقدة طرفية.

حيث أن b_n هي مرتبة العناصر غير الطرفية n، هذه العملية تتم لمرة واحدة ومن ثم تستمر الخوارزمية في إنشاء الشجرة حيث تبدأ طريقة PTC1 بتمرير قيمة p، وقيمة 0 من أجل d، وقيمة العمق الأقصى من أجل D، حيث يكون التعقيد الحسابي لهذه الطريقة هو خطي أو شبه خطي في حجم الشجرة المتوقع.

3.6 طريقة الانشاء الاحتمالي للأشجار 2 (Probabilistic Tree- 2 (Creation):[8] [9]

تقوم هذه الطريقة على تلقي حجم شجرة محدد، وتضمن أنه سيتم إنشاء شجرة لا يزيد حجمها عن الحجم المحدد، ولا يقل حجمها عن الحجم مطروحًا منه الحد الأقصى لتوزيع أي تابع (Function) في مجموعة التوابع. تعمل هذه الخوارزمية عن طريق زيادة أفق (horizon) الشجرة في نقاط يتم اختيارها بشكل عشوائي حتى تكون كبيرة بما يكفي.

تأخذ PTC2 حجم الشجرة المطلوب S. إذا كانت S = 1، فتُرجع طرفية عشوائية. وإلا فإنه يختار بشكل عشوائي عقدة غير طرفية كجذر للشجرة ويقلل من S بمقدار 1. ثم تقوم الطريقة بوضع كل فتحة غير ممثلة للعقد السابقة غير الطرفية في مجموعة تسمى H، والتي تمثل "أفق" الفتحات غير المعبئة. ثم يدخل الحلقة التالية:

إذا كانت $H \leq S$، يتم الخروج من الحلقة.

وإلا يتم حذف فتحة عشوائية من H، ويتم ملئها بعقدة غير طرفية مختارة بشكل عشوائي، وإنقاص S بمقدار 1، وإضافة كل فتحات العقدة الجديدة إلى المجموعة H، ومن ثم إعادة الخطوة السابقة. في هذه النقطة، يكون المجموع الكلي لعدد العقد غير الطرفية في الشجرة، بالإضافة لعدد الفتحات في H، يساوي أو بالكاد يتجاوز حجم الشجرة الذي يطلبه المستخدم. تنتهي PTC2 عن طريق إزالة فتحات من H واحد تلو الآخر وملؤها بالعقد الطرفية المختارة عشوائيا، حتى يتم استنفاد H، ومن ثم يتم إرجاع الشجرة. أيضا التعقيد الحسابي لـهذه الطريقة هو خطي أو شبه خطي في حجم الشجرة المطلوب.

4. منهجية البحث في توليد برامج لحركة روبوت:

سنقدم في هذا القسم شرحا موضحاً لاستخدام البرمجة الجينية في مجال توليد البرامج الروبوتية، حيث ستتوجه هذه الدراسة لمحاكاة عمل الوظائف الأساسية لروبوت في مضمار وهي التحرك لليمين أو لليسار أو للأمام، حيث يحتوي هذا المضمار على مسار فيه أهداف، ويجب على هذا الروبوت أن يسير على المسار للوصول للأهداف المطلوبة.

توليد هذا النوع من البرامج -بشكل تلقائي- غالبا ما يكون من الصعوبة بمكان، خاصة إذا كان الشخص القائم على تحديد حركة الروبوت لا يعلم -بشكل مسبق- شكل مسار الأهداف المرسوم ضمن المضمار والواجب على الروبوت تقفّيه. وهنا تكمن قوة البرمجة الجينية التي قمنا باستخدامها في هذه الدراسة وهي جعل الحاسوب يقوم بتوليد برنامج لحركة الروبوت السابق الذكر بشكل تلقائي.

اعتمدنا بشكل مرجعي على أبحاث ومسائل شهيرة مشابهة لدراسة حركة الروبوت وهي مسائل النمل الاصطناعي. حيث تعتبر مسائل (مشاكل) النمل الاصطناعي [8,2] من أهم المسائل المستخدمة في تقييم أداء الخوارزميات التطورية، حيث تتكون هذه المسائل من حقل مربعي الشكل تحتوي بعض خلاياه على كريات طعام "أهداف" يتم توزيعها في مسار معين. يكون الموضع الأولي للنملة في الخلية العلوية اليسرى من الحقل والبحث الأولي يكون باتجاه الشرق، ومن ثم يتم تحديد فيما إذا كانت الخلية التالية تحتوي على طعام أو لا. وفي كل خطوة يمكن أن تتحرك إلى اليسار أو إلى اليمين أو الإكمال إلى الأمام، وتناول قطعة من الطعام إذا كانت الخلية تحتوي على واحدة. حيث تهدف هذه المسائل إلى بناء برنامج يقوم بتحريك النملة بشكل يسمح لها بتناول كل الطعام "الأهداف" في عدد محدود من الخطوات.

في هذا النوع من المسائل يوجد ثلاثة أنواع من الخلايا وهي الخلايا الفارغة، والخلايا التي تحتوي على طعام، والخلايا التي تحتوي على تتمة المسار، ويوجد ثلاثة أنواع من الحركات وهي: Left وتعني الالتفاف لليسار، Right وتعني الالتفاف لليمين، Move وتعني الحركة للأمام. تم تقديم العديد من الدراسات والأبحاث العلمية لحل مسائل النمل منها ما تم تقديمه في [2]، حيث تم التطرق إلى استخدام FSM في بناء عميل (agent) قادر على محاكاة عمل النمل الاصطناعي وحل العديد من المسائل، أيضا قام بعض الباحثين في [3] بدراسة معايير وطرق تنوع البرامج المولدة باستخدام البرمجة الجينية حيث كانت إحدى أنواع المسائل المدروسة هي مسائل النمل.

نحن في هذه الدراسة سنعمل على ثلاثة أنواع لمسارات شهيرة من مسائل مسارات النمل وهي (Los - John Muir Trail – Santa Fe Trail Altos Trail) وستة أنواع أخرى من المسارات قمنا نحن بتصميمها لدراسة أداء البرمجة الجينية على مسارات مختلفة في درجة التعقيد.

4.1 *Santa Fe Trail [2] [1] [5]*

هي إحدى أشهر مسائل النمل الاصطناعي التي تستخدم لتقييم أداء الخوارزميات التطورية، حيث تتكون هذه المسألة من مضمار مربع الشكل أبعاده32x32، ويحتوي بداخله على مسار طوله الكلي 144، فيه 89 حبة طعام و21 انعطاف.

4.2 *John Muir Trail [2]*

حيث تتكون هذه المسألة من مضمار مربع الشكل أبعاده 32x32، ويحتوي بداخله على مسارين الأول طوله 35 ويحتوي على 35 حبة طعام و4 انعطافات، أما الثاني فطوله 92 ويحتوي على 54 حبة طعام و17 انعطافا، حيث يكون المسار الإجمالي لهذه المسألة هو 127، فيها 89 حبة طعام و21 انعطاف.

4.3 *Los Altos Hills Trail [1] [2]*

تعتبر هذه المسألة امتدادا للمسائل السابقة، حيث تتكون من مضمار مربع الشكل أبعاده 100x100، وتحتوي على مسار طوله الكلي 221، فيه 157 حبة طعام و29 انعطافا. تعتبر المسارات الثلاث السابقة من أشهر مسائل النمل الاصطناعي وأكثرها استخداما كونها تراعي تغيير العدد من المعاملات منها تغيير مساحة المضمار المستخدم، والطول الكلي للمسار بالإضافة إلى عدد حبات الطعام وعدد الانعطافات. ولذلك لجأنا إلى تصميم مسارات معدلة هدفنا من خلالها لدراسة أداء البرمجة الجينية عند تغيير معاملات في طبيعة وتصميم المسار، لنحاكي بذلك مسارات مختلفة ودراسة ومقارنة أداء البرمجة الجينية وقدرتها على إنتاج برامج تحل هذه المسارات.

4.4 *Tiny Santa Fe Trail (TSFT):*

قمنا بتصميم هذا المسار منطلقين من مسألة Santa Fe، حيث حافظنا على الشكل المربع للمضمار ذو الأبعاد 32x32، وقمنا بتقليل عدد الانعطافات حيث أصبح 14 انعطافا، مع المحافظة على الطول الكلي للمسار والبالغ 144، وعدد حبات الطعام 89.

4.5 *Big Santa Fe Trail (BSFT):*

هذا المسار هو أيضاً تطوير لمسألة Santa Fe، ولكننا قمنا بمضاعفة عدد الانعطافات حيث أصبح 42 انعطافا، مع المحافظة على الطول الكلي للمسار والبالغ 144، وعدد حبات الطعام 89. نهدف في المسارين BSFT وTSFT إلى دراسة تأثير إحدى عوامل البيئة المدروسة وهو **عدد الانعطافات** على أداء البرمجة الجينية، وقدرة البرمجة الجينية على إيجاد برامج قادرة على حل هذه المسارات، ومقارنة الأداء مع أداء البرمجة الجينية في مسار Santa Fe.

4.6 *Tiny John Muir Trail (TJMT):*

قمنا بتصميم هذا المسار منطلقين من مسار john Muir، حيث حافظنا على الشكل المربع للمضمار ذو الأبعاد 32x32، وقمنا بتقليل عدد المسارات ليصبح مسارا واحدا طوله الكلي 62، ويحتوي على 53 حبة طعام.

4.7 *Big John Muir Trail (BJMT):*

هذا المسار هو أيضاً تطوير لمسار john Muir، ولكننا قمنا بمضاعفة عدد المسارات ليصبح 4 مسارات حيث يكون الطول الكلي لهذه المسارات 127، وتحتوي جميعها على 89 حبة طعام. نهدف في المسارين TJMT وBJMT إلى دراسة تأثير عامل **تعدد المسارات** على أداء البرمجة التطورية، وقدرة البرمجة التطورية على تقديم برامج تحل هذا النوع من المسارات ومقارنة الأداء مع الأداء المقدم في مسار john Muir Trail .

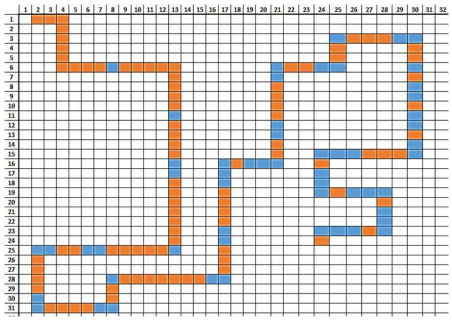
4.8 *Tiny Los Altos Trail (TLAT):*

قمنا بتصميم هذا المسار منطلقين من مسار Los Altos، حيث حافظنا على الشكل المربع للمضمار ذو الأبعاد 100x100، وقمنا بتقليل الطول الكلي للمسار ليصبح 173، حيث يحتوي على 136 حبة طعام و23 انعطافا.

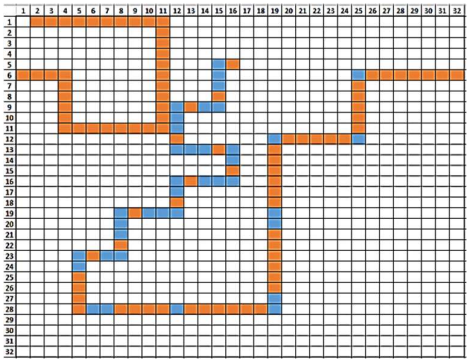
4.9 *Big los Altos Trail (BLAT):*

هذا المسار هو أيضاً تطوير لمسار Los Altos ، ولكننا قمنا بمضاعفة الطول الكلي للمسار ليصبح 442، حيث يحتوي على 314 حبة طعام و29 انعطافا. نهدف في المسارين TLAT وBLAT إلى دراسة تأثير عامل **طول المسار** على أداء البرمجة الجينية، وقدرتها على تقديم برامج تحل هذا النوع من المسارات، ومقارنة الأداء مع الأداء المقدم في مسار Los Altos.

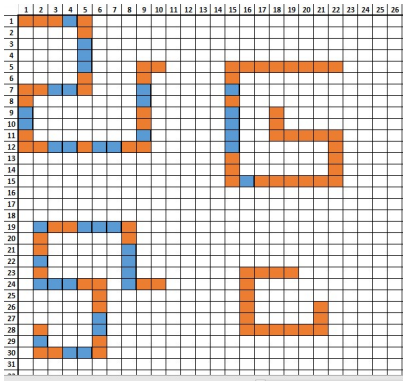
وفي الأشكال التالية (من 1 وحتى 9) نبين المسارات (المضامير) المستخدمة في الدراسة والتي تحتوي على الأهداف، حيث يعتبر كل مضمار هو عبارة عن قطعة أرض مسطحة أبعادها X و Y حسب أبعاد المسار كما تم ذكرها في الفقرات السابقة.



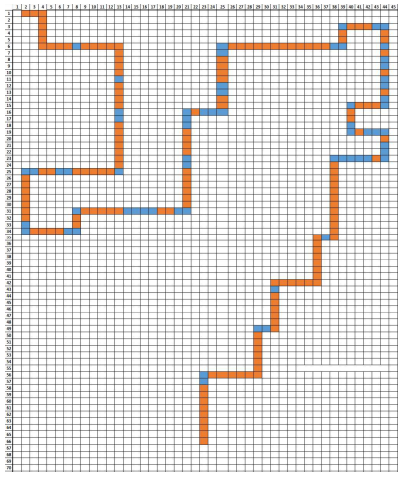
الشكل1: مسار مسألة Santa Fe Trail



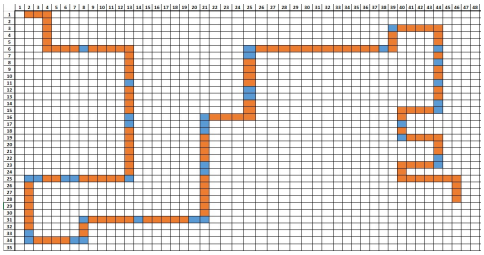
الشكل 2: مسار مسألة John Muir Trail



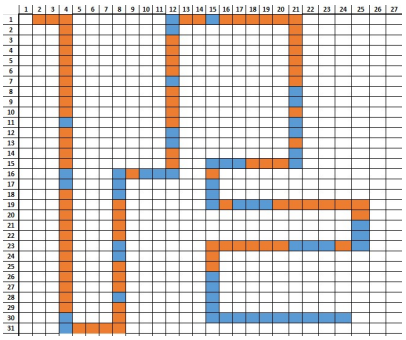
الشكل 7: مسار BJMT



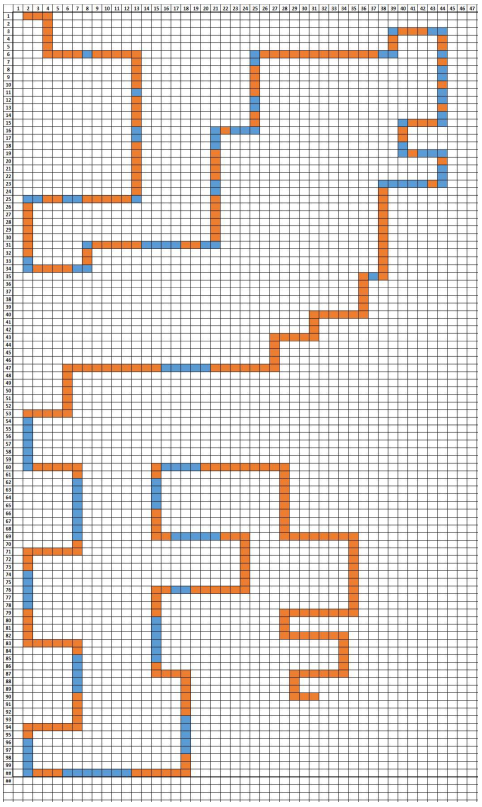
الشكل 3: مسار مسألةLos Altos Hills Trail



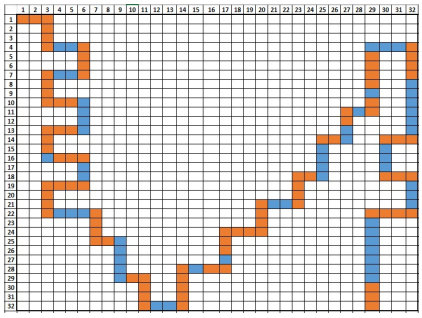
الشكل 8: مسار TALT



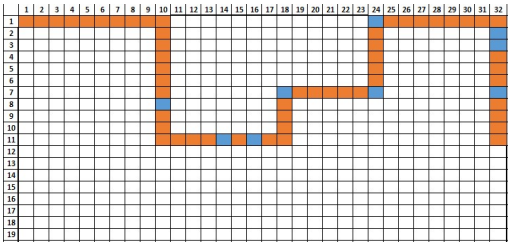
الشكل 4: مسار TSF



الشكل 9: مسار BLAT



الشكل 5: مسار BSFT



الشكل 6: مسار TJMT

5. تنفيذ الدراسة على بيئة المحاكاة ومناقشة النتائج:

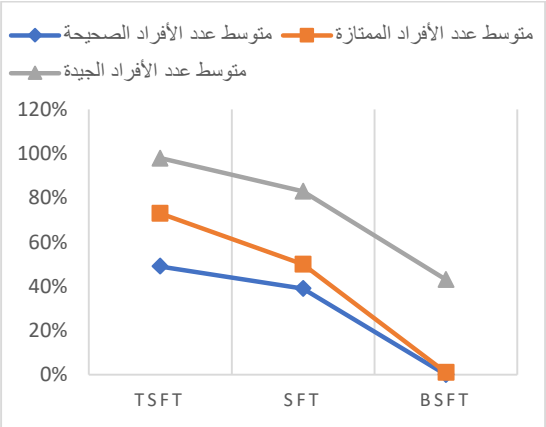
سنستخدم لمحاكاة الطرق السابقة منظومة ECJ، وهي عبارة عن نظام بحثي للحوسبة التطورية مكتوب بلغة جافا. تم تصميمه بحيث يتم فيه تحديد أغلب الأصناف (وجميع إعداداتها) ديناميكيًا في وقت التشغيل من خلال ملف معاملات يوفره المستخدم. وفيه يتم ترتيب جميع مكونات النظام لتكون قابلة للتعديل بسهولة. ولتطبيق خطوات البرمجة الجينية على مسائل النمل الاصطناعي والمسائل المعدلة السابقة، يجب علينا تحديد المعاملات الخاصة بالبرمجة الجينية، حيث يحتوي الجدول (1) على المعاملات اللازمة وقيمها. والتي يجب ضبطها قبل العمل على أي طريقة من الطرق الموضحة سابقا.

جدول1: معاملات البرمجة الجينية المحددة خلال الدراسة.

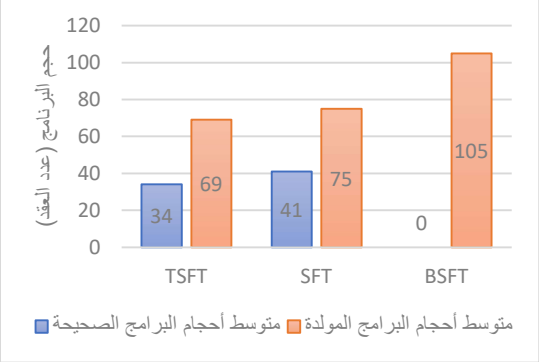
الهدف	إيجاد برنامج حاسوبي للتحكم بحركة نملة اصطناعية بحيث يمكنها ايجاد كل قطع الطعام الموجودة في مسار محدد
مجموعة الطرفيات Terminal Set	Left, Right, Move
مجموعة التوابع Function Set	ifFoodAhead, prog2, prog3
اللياقةFitness	إيجاد كل حبات الطعام "الأهداف" من قبل النملة
الاختيار Selection	TournamentSelection; size =7
عدد الخطوات steps	600 steps
حجم التجمع Population size	1024
معاملات التصالب والطفرات	90% crossover, 10% reproduction, no mutation
العمق الأقصى للتصالب Crossover max depth	17
الانتهاء Termination	50 generations, or find optimal program

5.1 *تأثير معامل عدد الانعطافات على أداء البرمجة الجينية:*

بعد تشغيل البرمجة الجينية باستخدام الاعدادات السابقة على مسارات TSFT وSFT وBSFT وجدنا أن طرق توليد الأجيال الأولية المستخدمة هذه الدراسة قدمت أداء ممتازاً نسبياً في حال كان عدد الانعطافات قليلاً، وكلما ارتفع عدد الانعطافات في المسار انخفض الأداء، حيث كما هو موضح في الشكل (10) نجد أن نسبة الأفراد الصحيحة التي تحل مسار TSFT هي 49% من إجمالي الأفراد المولدة، أما بالنسبة لعدد الأفراد الممتازة والتي حققت أعلى من 90% من المسار فكانت نسبة الأفراد هي 73% من إجمالي الأفراد المولدة، أما بالنسبة لعدد الأفراد الجيدة التي حققت أعلى من 80% فكانت نسبتها 98%. أما بالنسبة لمسارات SFT وBSFT نجد أن أداء البرمجة الجينية **ينخفض** بشكل ملحوظ في حال زيادة عدد الانعطافات في المسارات، وبالتالي نستنتج قاعدة مهمة وهي أن زيادة عدد الانعطافات في المسارات تؤدي إلى قلة عدد البرامج الصحيحة المولدة باستخدام البرمجة الجينية.



الشكل 01: أداء البرمجة الجينية تحت تأثير معامل عدد الانعطافات

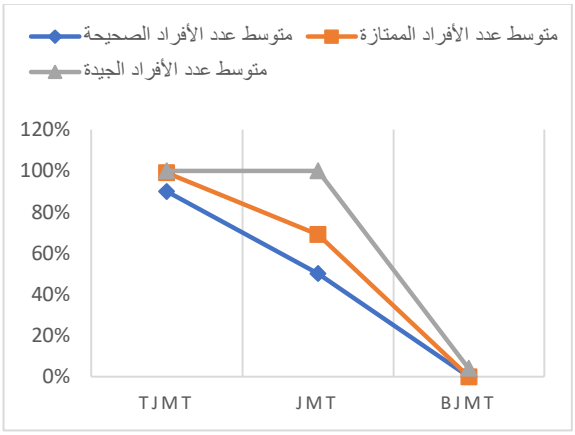


الشكل 11: تأثير عدد الانعطافات على أحجام البرامج.

أما بالنسبة لأحجام البرامج كما هو موضح في الشكل (11) فنجد أن زيادة عدد الانعطافات في المسارات المدروسة تؤثر بشكل سلبي على أحجام البرامج المولدة، دون العثور على حل صحيح وهذا ما يضر بالأداء كون زيادة الحجم تستهلك طاقة ووقت أكبر والأهم من ذلك أن زيادة الحجم تؤدي للابتعاد عن الحل الصحيح كما سنناقشه في الفقرة (5-4).

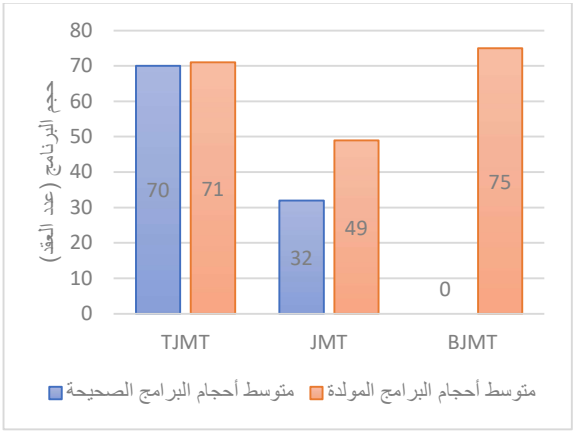
5.2 *تأثير معامل تعدد المسارات على أداء البرمجة الجينية:*

بعد تشغيل البرمجة الجينية باستخدام الاعدادات السابقة على مسارات TJMT و JMT وBJMT وجدنا أن طرق توليد الأجيال الأولية المستخدمة في هذه الدراسة قدمت أداء ممتازاً نسبياً في حال كان عدد المسارات الموجودة ضمن المضمار قليلاً، وكلما ارتفع عدد المسارات في المضمار الواحد انخفض الأداء، حتى مع المحافظة على عدد الانعطافات والطول الكلي للمسار.



الشكل 12: أداء البرمجة الجينية على المسارات المختلفة تحت تأثير تغير معامل عدد المسارات

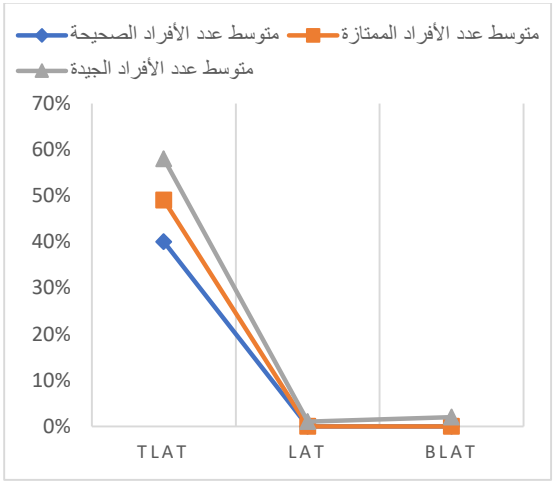
نجد من الشكل (12) أن نسبة الأفراد الصحيحة التي تحل مسار TJMT هي 90% من إجمالي الأفراد المولدة، أما بالنسبة لعدد الأفراد الممتازة والتي حققت أعلى من 90% من المسار هي 99% من إجمالي الأفراد المولدة، أما بالنسبة لعدد الأفراد الجيدة التي حققت أعلى من 80% فكانت نسبتها 100%. أما بالنسبة لمسارات JMT وBJMT نجد أن أداء البرمجة الجينية **ينخفض** بشكل ملحوظ في حال زيادة **عدد المسارات** الفرعية في المضمار، وهذا ما ينعكس سلباً على أداء البرمجة الجينية بشكل كلي. ومن هنا نستنتج قاعدة مهمة وهي أن أداء الطرق السابقة في البرمجة الجينية **ينخفض** بشكل ملحوظ في حال تعدد المسارات الفرعية الموجودة ضمن المضمار حتى في حال المحافظة على عدد الانعطافات والطول الكلي للمسار، وبالتالي فإن تعدد المسارات يؤدي إلى قلة عدد البرامج الصحيحة المولدة. أما بالنسبة لأحجام البرامج المولدة وأحجام البرامج الصحيحة كما هي موضحة في الشكل (13)، حيث نجد أن تعدد المسارات كان له تأثير سلبي في حالتين الأولى عندما كان المسار الكلي يتألف من مسار واحد فقط، والثانية في حال مضاعفة عدد المسارات لتصبح أربع مسارات.



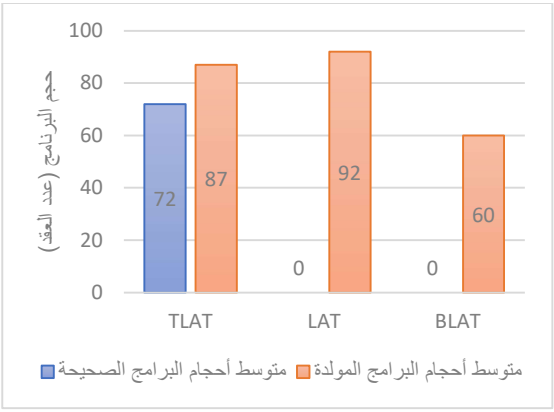
الشكل 13: تأثير عدد المسارات على أحجام البرامج.

5.3 تأثير معامل طول المسار على أداء البرمجة الجينية:

بعد تشغيل البرمجة الجينية باستخدام الاعدادات السابقة على مسارات TLAT وLAT وBLAT وجدنا أن طرق توليد الأجيال الأولية المستخدمة في هذه الدراسة قدمت أداء جيداً في حال كان طول المسار مقبولاً، وهذا ما وجدناه في كل المسارات السابقة TSFT-SFT-TJMT-JMT-TLAT، حيث كان طول المسار الكلي ما بين 53 و136، ولكن عندما ارتفع طول المسار الكلي أدى ذلك إلى انعدام أداء البرمجة الجينية وعدم انتاجها أفرادا صحيحة، وحتى الأفراد الممتازة أو الجيدة كانت شبه معدومة، وبذلك نستنتج قاعدة مهمة وهي أن معامل طول المسار له تأثير سلبي كبير على الأداء، لأنه كلما ارتفع طول المسار كلما انخفض الأداء. والشكل (14) يوضح نسب أداء البرمجة الجينية تحت تأثير تغير معامل طول المسار. أما بالنسبة لأحجام البرامج كما هي موضحة الشكل (15) نجد أن متوسط أحجام البرامج ارتفع في مساري LAT-TLAT أما بالنسبة لمسألة BLAT فكانت أقل ولكننا لا نستطيع الاعتماد على هذه القيم كونه لا يوجد برامج صحيحة أو ممتازة أو حتى جيدة كما سبق ذكره.



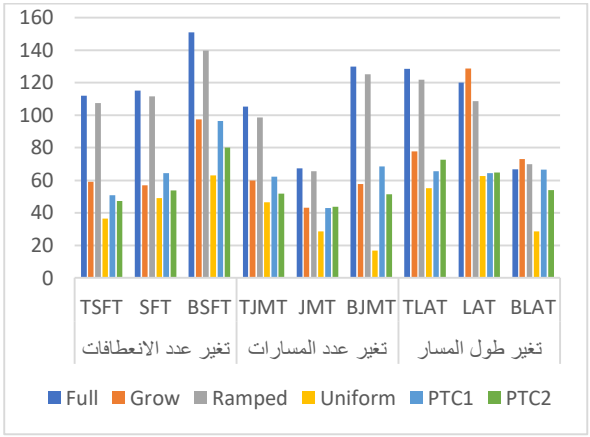
الشكل 14: أداء البرمجة الجينية على المسارات المختلفة تحت تأثير معامل طول المسار



الشكل 15: تأثير طول المسار على أحجام البرامج.

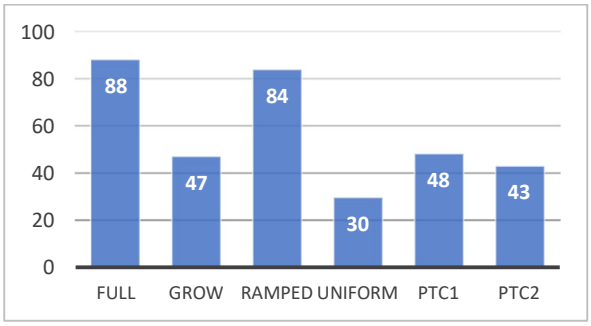
5.4 تأثير حجم البرنامج على عدد مرات التقييم المطلوبة:

وجب علينا دراسة تأثير علاقة تغير أحجام البرامج التي تم توليدها على تغير الأداء، كون المسارات السابقة كانت تواجه زيادة في أحجام البرامج المولدة عند زيادة عدد الانعطافات أو عدد المسارات أو زيادة الطول الكلي للمسار، وفي الشكل (16) التالي، قمنا بإنجاز عملية المتوسط الحسابي لأحجام البرامج المولدة في جميع الطرق، ومن الشكل نجد أن البرامج المولدة باستخدام الطريقة الموحدة (Uniform) هي أقل حجماً من الطرق الأخرى بنسبة تصل إلى أكثر من 50% في بعض المسائل كما هو موضح في الشكل.



الشكل 16: متوسط أحجام البرامج المولدة (متوسط عدد العقد في البرنامج).

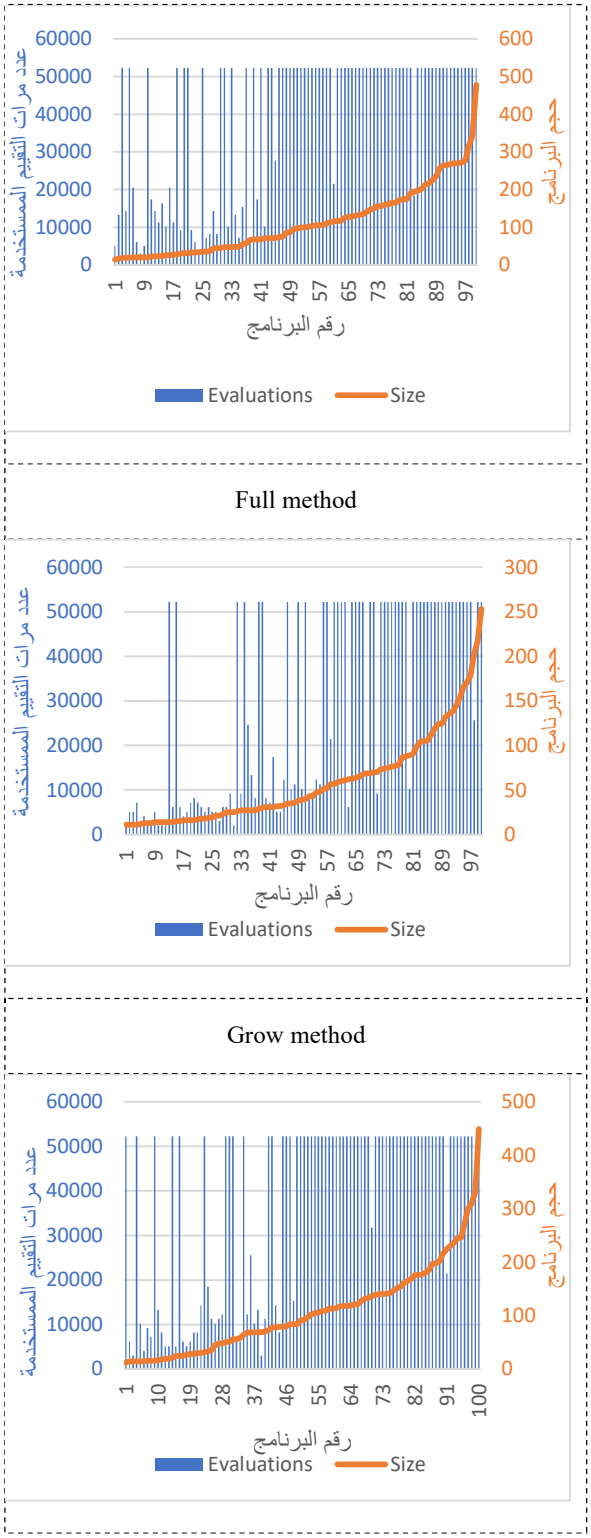
أما الشكل (17) فهو يوضح متوسط أحجام البرامج الصحيحة المولدة في الطرق السابقة بشكل عام، حيث أن البرامج الصحيحة المولدة باستخدام الطريقة الموحدة هي الأقل حجماً حيث بلغ متوسط الأحجام فيها 30 عقدة، وهي الأقل عن باقي الطرق بنسبة تصل إلى 60% من متوسط الأحجام المولدة. وتليها طريقتي الإنشاء الإحتمالي للأشجار (PTC1 , PTC2) والطريقة النامية (Grow) بمتوسط مابين 43 عقدة و48 عقدة، أما الطريقة الكاملة (Full) والطريقة المشتركة (Ramped) كان متوسط أحجام البرامج الصحيحة المولدة في كل منها تتروح مابين 84 و88 عقدة.



الشكل 17: متوسط أحجام البرامج الصحيحة المولدة لكل طريقة (متوسط عدد العقد في البرامج الصحيحة).

وهذا ما يدفعنا للتركيز على عامل زيادة الحجم وتأثيره على الأداء الكلي للبرمجة الجينية، كون هذا العامل هو أحد العوامل الخفية التي لا يمكن اكتشافها بسهولة، حيث كما نشاهد في الجدول (2) التالي، نجد أنه كلما ازداد حجم البرنامج (المرموز له في الجدول بالخط البرتقالي) كلما ازدادت عدد مرات التقييم التي يتطلبها انتاج البرنامج (المرموز لها في الجدول بالأشرطة الزرقاء)، وغالبا ما يكون هذا العدد الكبير من مرات التقييم دون جدوى ويكون الفرد غير مناسب (غير صحيح). وبذلك كلما ازدادت أحجام البرامج كلما ازدادت عدد مرات التقييم وبالتالي كلما ابتعدنا عن انتاج برامج الصحيحة، حيث أننا وجدنا خلال هذه الدراسة أن اغلب الافراد الصحيحة كانت أحجامها قليلة وبالتالي كانت تتطلب عدد مرات تقييم أقل، وهذا مابرهنته التجربة بتوليد 100 برنامج لكل مسار في كل طريقة من الطرق السابقة المذكورة في الدراسة ومقارنة أحجامها مع عدد مرات التقييم التي احتاجها كل برنامج.

جدول 2: العلاقة ما بين حجم البرنامج (حجم كل فرد) وعدد مرات التقييم التي احتاجها.

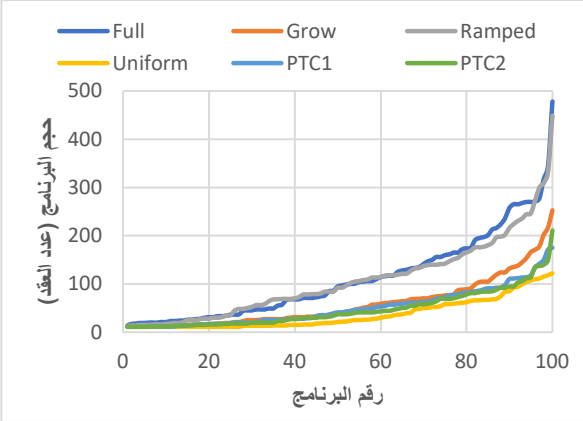


حجم البرنامج عن هذه القيمة أصبحت البرامج - في الغالب -عديمة الفائدة كونها أصبحت تحتاج إلى عدد مرات تقييم كثيرة دون الوصول إلى حل صحيح للمسألة.

وهذا يقودنا إلى المقارنة مابين الطرق السابقة من حيث أحجام البرامج الناتجة، حيث نلاحظ من الشكل (18) أن أفضل الطرق في توليد برامج مناسبة (لمسألة مسار TSF على سبيل المثال)، كانت الطريقة الموحدة (Uniform method) كون أن 90% من البرامج المولدة في هذه الطريقة هي من المناسبة لحل هذه المسألة وذلك لأن أحجامها أقل من 50 عقدة في البرنامج الواحد.

وتليها طريقتي الإنشاء الإحتمالي للأشجار (PTC1 وPTC2) والطريقة النامية (Grow) كون أن أكثر من 80% من البرامج المولدة هي برامج مناسبة كانت أحجامها أقل من 50 عقدة.

أما بالنسبة للطريقة الكاملة (Full) والطريقة المشتركة (Ramped) كانت نسبة البرامج المناسبة هي أقل من 60% وهي أقل نسبة مقارنة بالطرق السابقة.

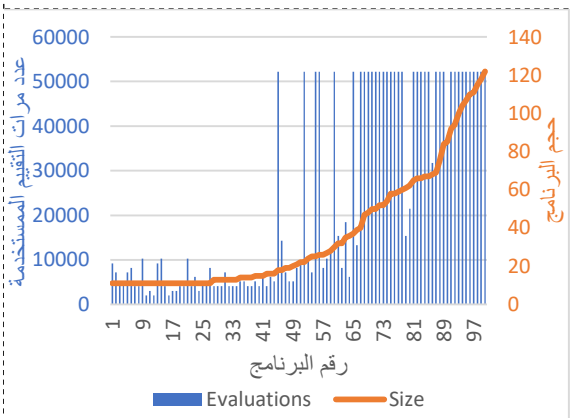


الشكل 18: مقارنة مابين أحجام البرامج المولدة في الطرق المستخدمة في الدراسة.

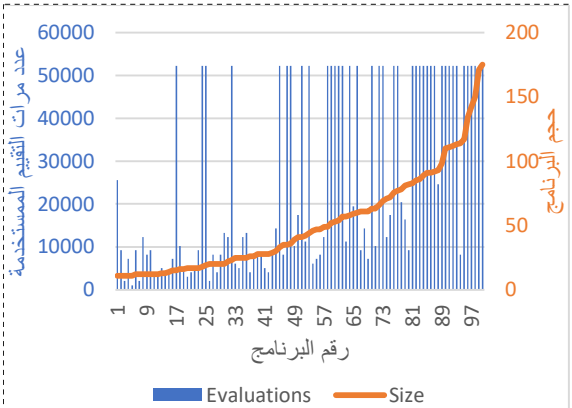
5.5 *تأثير حجم البرنامج المولد على لياقة الفرد:*

وجدنا في هذه الدراسة أن زيادة حجم البرنامج كان لها تأثير سلبي أيضا على لياقة الأفراد الناتجة ومدى صحتها أو قربها من أن تكون برامج صحيحة، وكما هو موضح في الجدول (3) التالي وجدنا أن صحة الأفراد (المرموز لها في الجدول بالخط الأزرق) تنخفض كلما ازدادت أحجام البرامج (المرموز لها في الجدول بالأشرطة البرتقالية)، وبذلك نستنتج أن لياقة البرامج المولدة تتناسب بشكل عكسي مع حجم البرنامج، ولذلك تأثير على عدد البرامج الصحيحة التي يمكن إنتاجها، ولذلك يجب أخذ هذه العلاقة بعين الاعتبار أثناء ضبط معاملات تشغيل البرمجة الجينية.

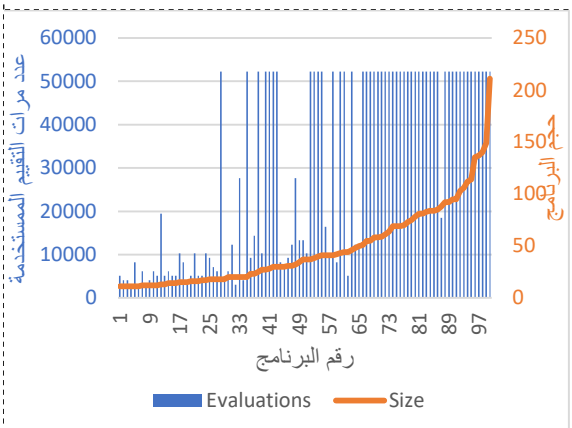
Ramped half-and-half method



Uniform method



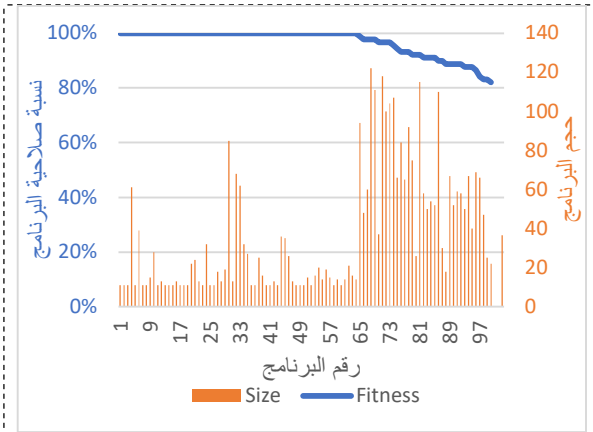
PTC1 method



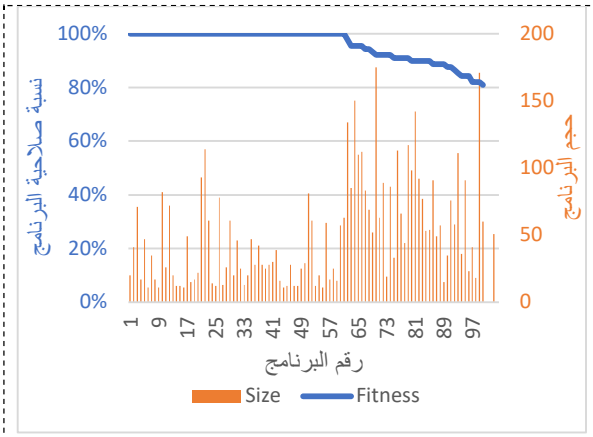
PTC2 method

نستنتج أيضاً من الجدول (5) السابق أن البرامج (الأفراد) الصحيحة كانت -في الغالب- ذات أحجام أقل من 50 عقدة في الشجرة الواحدة، وعندما يزداد

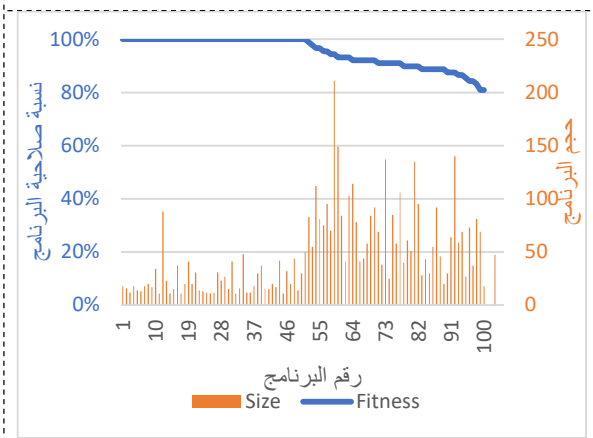
جدول 3: تأثير حجم البرنامج على لياقة الفرد في كل طريقة من الطرق المستخدمة.



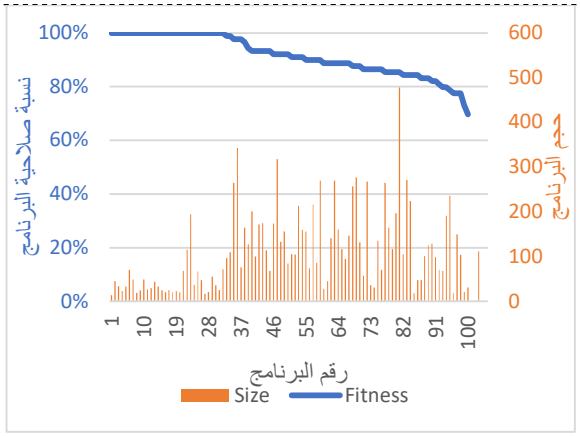
Uniform method



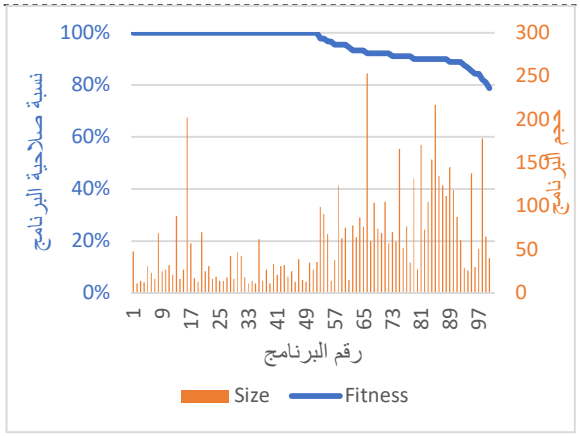
PTC1 method



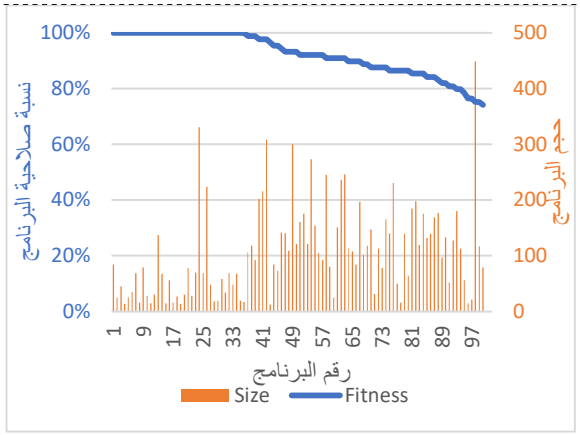
PTC2 method



Full method



Grow mthod

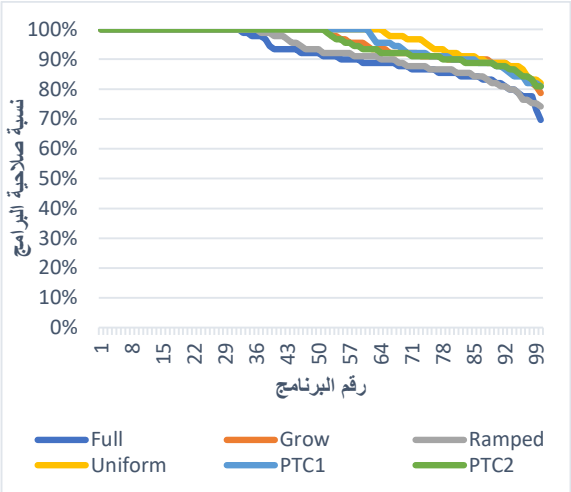


Ramped half-and-half method

الجدول (3) السابق يقودنا للمقارنة مابين الطرق السابقة من حيث نسبة صلاحية البرامج المولدة، ولذلك قمنا بمقارنة البرامج المولدة - لمسألة TSFT على سبيل المثال- والناتجة باستخدام الطرق السابقة، نجد كما هو موضح في الشكل (19) التالي أن أفضل الطرق في توليد برامج صحيحة هي الطريقة

الموحدة (uniform) كون أن أكثر من 62% من البرامج المولدة باستخدامها كانت برامج صحيحة بنسبة 100%، وتليها طريقتي الانشاء الإحتمالي (PTC1 وPTC2) والطريقة النامية (Grow) بنسب تتراوح مابين 50% و60% من البرامج المولدة باستخدمها كانت صحيحة بنسبة 100%.

أما بالنسبة للطريقة الكاملة (Full) والطريقة المشتركة (Ramped) كانت من أسوأ الطرق في توليد برامج صحيحة بنسبة 100%، حيث كانت نسبة البرامج الصحيحة فيهما تتراوح مابين 30% و40% من إجمالي الأفراد المولدة.



الشكل 19: نسبة صلاحية البرامج المولدة لمسألة TSFT.

6. مناقشة النتائج:

قمنا في هذه الدراسة بتطبيق الطرق المستخدمة في توليد الأجيال الأولية في البرمجة الجينية القائمة على الأشجار بهدف الحصول على برامج مولدة بشكل تلقائي لحركة روبوت متتبع مسارات متعددة ضمن مضمار، وناقشنا أداء هذه الطرق على أنواع مختلفة من المسارات وذلك بهدف الحصول على رؤية أوضح لأداء البرمجة الجينية في مجال توليد البرامج الروبوتية.

استخدمنا أنواع مختلفة من المسارات لدراسة تأثير عدة أنواع من المعاملات، واستنتجنا خلال هذه الدراسة عدة قواعد ونقاط هامة كان لها تأثير كبير على أداء الطرق السابقة وهذه القواعد سيتم الارتكاز عليها عندما سنقوم بتطوير إطار عمل برمجي خاص بالتطبيقات الروبوتية، وهي أنه بزيادة عدد الانعطافات في المسار المدروس أو زيادة عدد المسارات الفرعية أو بزيادة طول المسار فإن قدرة الطرق السابقة في إنتاج برامج صحيحة لحل المسائل كانت معدومة 0%، حيث أن أحجام البرامج ازدادت بشكل كبير مما أضر بالأداء الكلي للطرق السابقة.

استنتجنا في هذه الدراسة أن أفضل الطرق في توليد برامج بأحجام مناسبة كانت الطريقة الموحدة (Uniform method) كون أحجام البرامج الصحيحة المولدة في هذه الطريقة هي أقل حجما بنسبة تصل إلى 60% من البرامج المولدة في الطرق الأخرى، وبمتوسط أحجام 30 عقدة، وتليها طرق الإنشاء الاحتمالي للأشجار (PTC2,PTC1) والطريقة النامية (Grow) بمتوسط أحجام برامج مابين 43 عقدة 48 عقدة، أما بالنسبة للطريقة المشتركة (Ramped) و الطريقة الكاملة (Full) كانت من أسوأ الطرق من ناحية حجم البرامج الصحيحة المولدة بمتوسط أحجام كبير نسبياً مابين 84 عقدة و 88 عقدة.

قمنا أيضا بدراسة تأثير زيادة حجم البرامج على عدد مرات التقييم التي يتطلبها انتاج البرامج، واستنتجنا أثناء هذه الدارسة أنه بزيادة أحجام البرامج المولدة تزداد عدد مرات التقييم إلى الحد الأعلى المسموح فيه (52000 عملية تقييم)، وبالتالي تقل أعداد البرامج الصحيحة أو القريبة من الصحة، أيضا قمنا بدراسة عامل زيادة الحجم على لياقة الأفراد واستنتجنا في هذه الدراسة أن الأفراد ذات اللياقة العالية تكون ذات أحجام منخفضة، ولذلك يجب الانتباه لهذه النقاط كونها ستحدد مصير تشغيل البرمجة الجينية بشكل كلي. تعتبر النتائج التي توصلنا لها في هذه الدراسة مهمة جداً كونها أوضحت العديد من العوامل المؤثرة التي لم تأخذ في الأبحاث المرجعية السابقة، وكونها حددت قواعد مهمة لتطوير أي خوارزمية أو طريقة لتحسين أداء البرمجة الجينية في توليد التطبيقات الروبوتية، حيث أننا سنعتمد على نتائج هذه الدراسة في إنشاء وتطوير إطار عمل غرضي التوجه لتوليد برامج حركة روبوت تلقائياً وهو ما سيتم تقديمه في أبحاث أخرى.

References

[1] Christopher H., William T., 2014, "Benchmarking Grammar-Based Genetic Programming Algorithms," International Conference on Innovative Techniques and Applications of Artificial Intelligence, SGAI 2014: Research and Development in Intelligent Systems XXXI pp. 135-148.

[2] Daniil C., Anatoly S., Vladimir U. 2013, "Solving Five Instances of the Artificial Ant," 7th IFAC Conference on Manufacturing Modelling, Management, and Control, pp. 1043-1048.

[3] Edmund B., Steven G., Graham K., 2002, "A Survey and Analysis of Diversity Measures in Genetic Programming," Proceedings of the 4th Annual Conference on Genetic and Evolutionary ComputationJuly 2002 pp. 716–723.

[4] Gabrielė K., Stig Anton N., Dhruv B., Peter N., Alexandru D., 2019, "Plastic Grabber: Underwater Autonomous Vehicle Simulation for Plastic Objects Retrieval Using Genetic Programming," Business Information Systems Workshops Lecture Notes in Business Information Processing, Vol. 339, pp. 527-533.

[5] John K., "Genetic Programming," http://geneticprogramming.com/tutorial/. [Accessed 19 2 2020].

[6] Juan R., Antonino S., Adrian C., Nereida R., Iria S., Alvaro T. , Juan T., Penousal M., 2020. "A Genetic Programming-Based Low-Level Instructions Robot for Realtimebattle," Artificial Intelligence and Complexity in Art, Music, Games and Design Entropy 2020, 22(12), 1362.

[7] Riccardo P., William L., Nicholas M., John K., 2008,"A Field Guide to Genetic Programming, " ISBN 1409200736.

[8] Sean L, 2000, "Two Fast Tree-Creation Algorithms for Genetic Programming," IEEE Transactions on Evolutionary

[10] William L., 2000, "Size fair and homologous tree genetic programming crossovers," Genetic Programming and Evolvable Machines volume 1, pp. 95–119.

ComputationVolume 4, Issue 3, September 2000 pp. 274–283.

[9] Sean L., Liviu P., 2000, "A Survey and Comparison of Tree Generation Algorithms," Proceedings of the 3rd Annual Conference on Genetic and Evolutionary ComputationJuly 2001 pp. 81–88.

The methods of generating initial populations in tree-based genetic programming, and applying them to create robot's movement program within bounded area.

***Abdulrahman Salloum*[1], *Mahmoud Rahhal*[2], and *Mohammed Khatib*[3,*]**

[1] *University of Aleppo, Aleppo, Syria, Abdulrahamansalloum92@gmail.com*

[2]*Faculty of Electrical & Electronic Engineering, University of Aleppo, Aleppo, Syria, Mrahhal2000@htmail.com*

[3] *Faculty of Informatics Engineering, University of Aleppo, Aleppo, Syria, mhmd.khatib1970@gmail.com*

**Corresponding author: Abdulrahman Salloum, Abdulrahmansalloum92@gmail.com*

Published online: 31 December 2021

Abstract— In this research, we aim to study the methods of generating initial populations in tree-based genetic programming and using them in generating a program for robot movement. This study includes several methods, which are, Grow method, Full method, Ramped Half and Half, Uniform method, Probabilistic Tree-Creation 1 and 2, We applied these methods in generating programs for a multiple path tracker robot and we simulated these methods on artificial ants' problems, which are Santa Fe trail problem, John Muir trail problem, and Los Altos trail problem, and adding modified paths to this research in order to study the effect of changing multiple factors such as increasing the number of turns in paths, increasing the number of subpaths, and increasing path length. We found in this study that the best method was the uniform method, because of the size of the correct programs was 66% less than the size of the correct programs generated other methods, then PTC2 at 51%, Grow method at 47% and PTC1 at 45%, and we found that the increment in program size had a negative impact on the program's fitness, the number of Evaluations, and on the percentage of correct programs.

Keywords— genetic programming, generate initial population, Robot programs.