استخدام الخوارزمية الجينية في جدولة المهام في الأنظمة المتعددة المعالجات

أسماء ياسين حمو **غصون سالم بشير** **منى محمد طاهر**

كلية علوم الحاسبات والرياضيات

جامعة الموصل

تاريخ استلام البحث: ٢٠٠٦/٦/٧ **تاريخ قبول البحث**: ٢٠٠٦/١٠/٤

ABSTRACT

In this work we use Genetic Algorithm for best section to implement many independent tasks on multiprocessor systems. The chromosome represented by numbers of integer value, every value represents one of the processors in the system, we use the simple crossover to generate the next population, and we us the mutation of type partial gen for mutation which has a good role to improve results of scheduling, the program written by matlab (6.5).The results, after a small number of iterations, were very good .

الملخص

تم في هذا البحث اعتماد الخوارزمية الجينية في الاختيار الأفضل لتنفيذ عدد من المهام المستقلة(Independent Tasks) في الأنظمة متعددة المعالجات. تم تمثيل الكروموسوم بشكل إعداد صحيحة يمثل كل عدد منها معالجاً من المعالجات الموجودة في النظام الذي ستتنفذ عليه المهام، تم استخدام التزاوج من النوع البسيط(simple crossover) لتوليد الجيل التالي، وتم تنفيذ البرنامج باستخدام الطفرة من نوع الجين الجزئي(Partial Gen) وتمت مقارنة النتائج مع النتائج بتنفيذ البرنامج دون استخدام الطفرة مما وضح الدور الجيد للطفرة في تحسين النتائج وتسريع الحصول على الأفضل.تمت كتابة البرنامج بلغة(matlab(6.5. تم الحصول على نتائج جيدة جدا بعدد تكرار بسيط، يمكن الاعتماد عليه للحصول على جدولة سريعة وفعالة وبمساحة خزن معقولة.

1.مقدمة

إن المسألة المهمة والأصعب في النظم المتعددة المعالجات (Multiprocessor system) هي الجدولة. والجدولة تعني تخصيص مجموعة من المهام المراد تنفيذها من قبل النظام على مجموعة من المصادر في ذلك النظام، وتعرف على أنها صعبة الحل(NP-Complete) لذلك فان هذه الجدولة غالبا ما تعالج بطرائق ألـ heuristic التي تجهز حلول معقولة للمسألة.(12)

2.الأعمال السابقة:

إن الأسبقية للمهام التي تنفذ على الأنظمة ذات المعالج الواحد أو على الأنظمة المتعددة المعالجات تكون على نوعين: ثابتة(Fixed Priority) منذ دخول المهمة إلى النظام أو تكون متغيرة(Dynamic Priority) هي التي تدخل النظام بأسبقية معينة وتتغير هذه الأسبقية حسب ظروف معينة داخل النظام(11)، اعتبر كل من Lee,Y._H. و.Ghen,C(10) في عمليهما وجود معالج واحد في النظام والأسبقية من نوع المتغير، بينما Cook, D. J&Kidwell, M.D.(7) كانت الأسبقية في بحثيهما للمهام من النوع الثابت، وكان فيه تقليل لكلفة الحسابات لكن كانت الكلفة بسبب عمليات التحميل. وفي البحث الذي قدمه.Hou, E.S.,Ansari, N. Ren, H(6) كانت الأسبقية أيضا من النوع الثابت لكن المهام كانت معتمدة (Dependent) ووقت التنفيذ ثابتاً ومعلوماً لكل مهمة، بينما كانت المعالجات مستقلة في عملها، كان الهدف تقليل الوقت المستغرق في جدولة المهام التي تصل إلى النظام، وقد حققت نتائجهم ١٠% من الوصول إلى الحل الأمثل، .Nafpliotis,N. Horn,J.و (5) الأسبقية كانت من النوع المتغير(dynamic)والمهام كانت معتمدة والمعالجات مستقلة في عملها ،كانت المساويء هي الحاجة إلى عدد كبير من المعالجات لكي تتنفذ المهام على اعتبار إن المهام معتمدة واحدة على الأخرى، في بحث.El-Gendy ,S.M (4) الأسبقية للمهام كانت ثابتة لكن فرض الباحث وجود عدد من المهام(m) قسمت إلى مجاميع تجدول على(n) من المعالجات، Pai_Chou Wang ,W. Kofhage (15) استخدم خوارزمية(Myopic Scheduling) وهي خوارزمية بحث تأخذ بنظر الاعتبار تقييدات المصادر(Recourse Constrain)، في(11) كانت الأسبقية من النوع المتغير و المصادر معتمدة.

من النقاط المهمة والأساسية في المسائل التي تستخدم الخوارزمية الجينية في الحصول على حلول أمثلية هي مسألة تمثيل الكروموسوم، وبما أن المسألة تتعلق بمساحة الخزن فلا بد من

تمثيل الكروموسوم بحيث يشغل اقل مساحة ممكنة في الذاكرة خاصة في الخوارزمية الجينية التي تعتمد في الحصول على نتائجها على عدد تكرارات معينة والتي بدورها تحتاج إلى مساحة خزن كبيرة، فضلاً عن حاجة هذه المسائل إلى تمثيل يسهل معه التعامل المتكرر مع الكروموسوم.Mahmood ,A(11) مثل الكروموسوم على انه زوج من الأعداد الصحيحة إذ يمثل كل زوج من الأزواج معالجاً من المعالجات الموجودة في النظام و واحدة من المهام التي في النظام والتي تمت جدولتها لتتنفذ على ذلك المعالج،و في.Kianzad,V. &Rinehart,M, &Bhattacharyya,S.,S.(17) تم تمثيل الكروموسوم على انه سلسلة (string) وتم التعامل مع المهام على إنها رسم(graph)،في بحث Ren,H&Hou, E. S., Ansari , N. (6) تمثيل الكروموسوم كان بسلسلة ذات طول مختلف، استفاد من ذلك لتوسيع مساحة الخزن لترتيب المهام التي ستجدول على المعالجات،لكنR.C.Correa,A. Ferreira,and P. Rebreyend (16) اثبت إن هذا التمثيل مستحيل أن يصل إلى الحل الأمثل بسبب اعتبارات الوقت المهمة في الامثلية، واثبت من خلال عمل Ren,H&Hou, E. S. Ansari , N.(5) انه ممكن باستخدام البحث الكامل للخوارزمية الجينية(Full Search Genetic Algorithm)(FSG)مع تمثيل الكروموسوم بشكل سلسلة تم الوصول إلى فضاء بحثي يتم الوصول من خلاله إلى جدولة مقبولة، إن الـ(FSG) هي واحدة من قائمة الطرق الحدسية والقائمة الكاملة للطرق الخوارزمية الحدسية تسمى قائمة الخوارزمية الجينية المعقدة(Combined Genetic List Algorithm)(CGL) وتم استخدامها من قبل.Rinehart,M. &,Kianzad,V.&Bhattacharyya ,S.,S(18) حيث مثل الكر وموسوم بقائمة من السلاسل وكل سلسلة تمثل معالجاً وبضمنها ترتيب التنفيذ لكل معالج.

في هذا البحث تم اعتبار أن أسبقية المهام من نوع المتغير(Dynamic) وان المهام الموجودة في النظام والمصادر مستقلة عن بعضها البعض(Independent) وان أسبقية المهام تحدد من قبل النتائج التي نحصل عليها من الخوارزمية الجينية، وتم تمثيل الكروموسوم على انه مجموعة أعداد صحيحة تمثل الترتيب للمعالجات في النظام(طول الكروموسوم بعدد المعالجات الموجودة في النظام(تم فرضها خمس معالجات))، والتي ستجدول عليها المهام مما حقق مساحة خزن اقل وسرعة تنفيذ أعلى للخوارزمية.

طرائق الجدولة في الأنظمة المتعددة المعالجات يمكن تقسيمها إلى نوعين:

A: List heuristic: تحدد أسبقية كل مهمة وتترتب المهام تنازليا حسب تلك الأسبقيات.

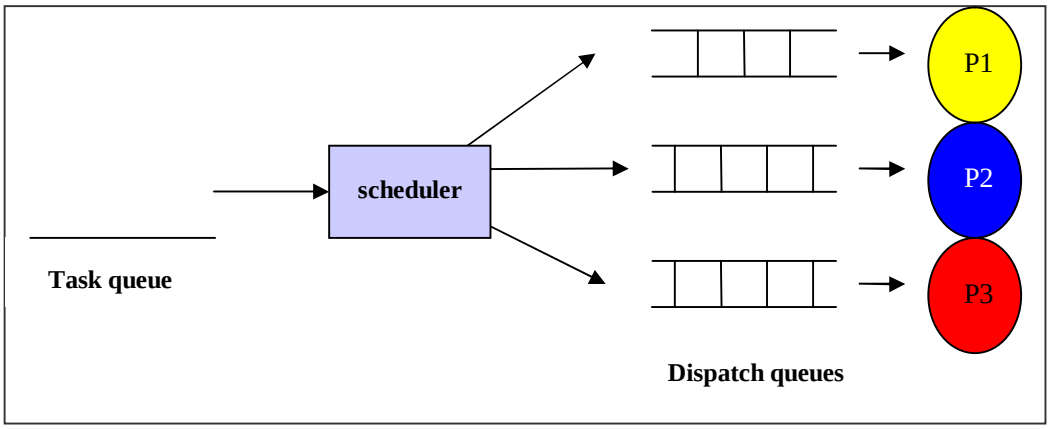
B: Meta heuristic: والتي تعرف بالخوارزمية الجينية(Genetic Algorithm)وهي طريقة بحث عشوائية ويكون عملها مستمدا من خواص الجينات الطبيعية. (9)

والمسألة المهمة هنا هي تحديد متى وعلى أي من المعالجات سيتم تنفيذ كل مهمة من المهام الموجودة في النظام والتي تنتظر الخدمة.(12)

2.جدولة المهام في الأنظمة المتعددة المعالجات:

توصف بأنها النظم المكونة من عدة معالجات ترتبط ببعضها، كل معالج من هذه المعالجات يمكنه تنفيذ مهمة في الوقت الواحد ولا يمكن سحب ذلك المعالج إلى مهمة أخرى إلا بعد انتهائه من تنفيذ المهمة الحالية (11).

على فرض أن كل المهام(تكون غير معتمدة على بعضها البعض)(Independent Tasks) تصل إلى معالج مركزي وهذا يسمى المجدول(Scheduler) ويقترن به طابور (Queue)يسمى طابور المهام(Task Queue) وتترتب فيه المهام التي تصل باستمرار إلى النظام، يقوم هذا المعالج بتوزيع المهام إلى بقية المعالجات في النظام التي تقوم بتنفيذها .يقترن بكل معالج فرعي طابور خاص به يسمى(Dispatch Queue)عن طريقه يتصل المعالج الفرعي بالمعالج الرئيسي(المجدول)، يعمل المجدول مع بقية المعالجات بشكل متوازٍ إذ يقوم بجدولة المهام التي تصل حديثا.خوارزمية الجدولة تهتم بالمهام التي تكون فعالة آنيا لكن تهمل المهام الجديدة والتي تصل خلال تنفيذ مهام مجدولة على المعالجات المتوافرة في النظام.(11) , كما موضح بالشكل (1).

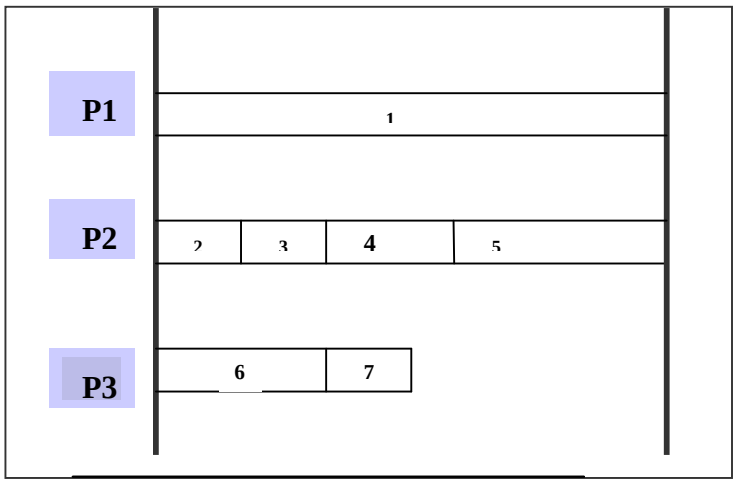


الشكل (1) يوضح علاقة المجدول مع بقية المعالجات في النظام. (11)

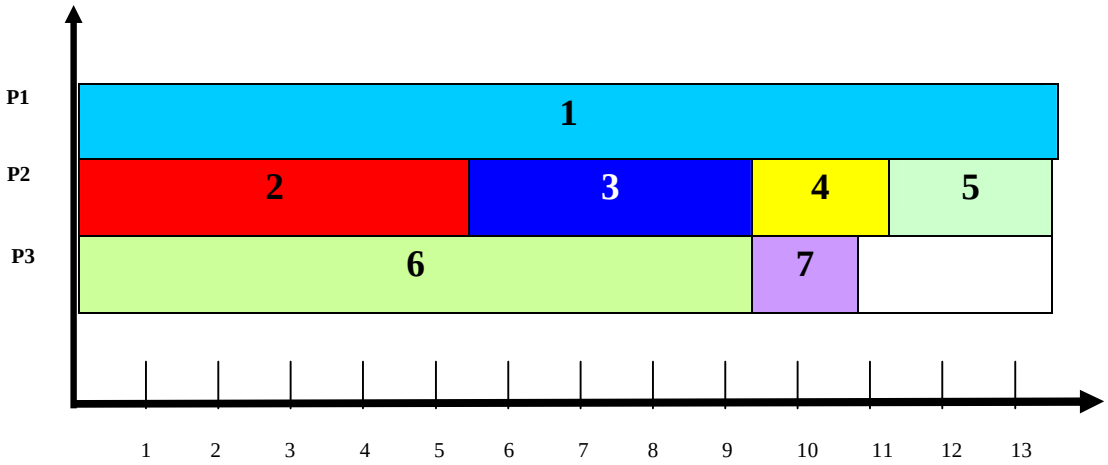
على فرض أن النظام يتكون من m من المعالجات المستقلة التي ترتبط مع بعضها البعض، المهام الموجودة ممكن أن تنفذ على أي من المعالجات, أي من هذه المهام قد يحتاج إلى بعض مصادر النظام(Resource)مثل هياكل بيانات،ذاكرة اتصال أو متغيرات، بعض هذه المصادر لا تكون لمهمة واحدة في أن واحد ولا يحق لمهمتين أو أكثر الاشتراك فيها، بينما مصادر أخرى تسمح باشتراك أكثر من مهمة في استخدام المصدر في أن واحد. كما أن هناك نوعين من المجدولات: قابل للتدخل(Preemptive) والذي فيه يمكن أن تقطع المهمة من قبل مهام أخرى، وغير قابل للتدخل(Non-Preemptive) وفيه يجب أن يكمل تنفيذ المهمة التي بدأت.(12)، في بحثنا تم استخدام (Non-Preemptive) .الشكل (2) يوضح كيفية جدولة المهام على المعالجات المتعددة في النظام الواحد,والشكل (3) يوضح المخطط الزمني لهذه الجدولة.

3.الخوارزمية الجينية(Genetic Algorithm) :

تعرف على إنها تقنية بحث ذكائية للوصول إلى الحل الأمثل(Optimal Solution)، طورت سنة1970واستمدت أفكارها من الجينات الطبيعية الوراثية، تعتمد على فكرة البقاء للأصلح



الشكل (2) يوضح عملية الجدولة في الأنظمة المتعددة المعالجات. (13)



الشكل (3) يوضح المخطط الزمني لجدولة المهام في الأنظمة المتعددة المعالجات. (13)

حيث تنتقي الأفراد الأفضل وتهمل الأفراد ذات الصفات السيئة نسبيا(14). الخطوة الأولى هي تكوين جيل عشوائي من الأفراد للمسألة ثم تقييم هذه الحلول بالاعتماد على دالة تقييم(Fitness Function) وتختلف الأخيرة حسب اختلاف المسألة المراد معالجتها، ومن ثم ترتب النتائج لتجري عليهاعملية الاختيار وعادة يوجد عدة طرائق للاختيار و من الطرائق

المشهورة والمستخدم بكثرة في تطبيق الخوارزمية الجينية هي الاختيار بعجلة الروليت (Rollet Well) التي تمتاز باختيار أفراد ذوي قيمة لدالة التقييم عالية بنسبة كبيرة ولا تهمل الأفـراد ذوي القيمة الواطئة لدالة التقييم حيث تختار نسبة بسيطة من هذه الأفراد حتى لا يكون هنالك أي إهمال لنتائج البحث، بعد أن تتم عملية الاختيار يتم توليد أفراد الجيل التالي بالتزاوج الحاصل مـا بـين الأفراد التي اختيرت، والتزاوج عبارة عن تقاطع اثنين من الكروموسومات بنقطة تحدد عـشوائيا وتكوين اثنين من الكروموسومات الجديدة ،على سبيل المثـال إذا كـان عنـدنا الكروموسـامان الأول(3 ,1 ,4 ,5 ,2) والثاني(2 ,1 ,5 ,4 ,3) وعلى فرض أن نقطة التقاطع كانت في موقع الجين الثالث في كلا الكروموسومين فان الناتج من التزاوج سيكون الكروموسومين الآتيان:(2, 2 ,1 ,5 ,5) (3 ,1 ,4 ,4 ,3).(4)

ثم يتم إجراء الطفرة، وتلعب الطفرة دوراً مهماً وفعالاً في الحصول علـى نتـائج قـد توصل إلى الحل الأمثل فيما بعد، إذ أن الطفرة تعطي فرصة لأي جين في أي كروموسوم لكـي يتغير إلى قيمة عشوائية غير موجودة سابقا، وهذه يمكن اعتبارها فرصاً إضافية للحصول علـى حل جديد قد يكون الأفضل. **هنالك ثلاثة أنواع من الطفرة**:

1: **الطفرة المعتمدة على الترتيب(Order Based Mutation)**: يتم اختيار اثنين من الجينات عشوائيا من الكروموسوم الذي وقع عليه الاختيار لإجراء الطفرة ويتم تبديل الجينين مع بعضيهما البعض .

2: **الطفرة المعتمدة على القائمة الفرعية(Sub List Based Mutation)**: يتم اختيار زوجين من الجينات عشوائيا من كروموسوم واحد ويجري تبديل مواقعهما .

3: **الجين الجزئـي(Partial Gene)**: يتم تغيير قيمة جين من جينات الكروموسوم إلـى قيمـة مختلفة ، كأن يكون رقم المعالج ١ يقلب إلى رقم أخر من أرقام المعالجـات الموجـودة (مـثلا إلى5).(11)

يلي عملية الطفرة عملية الإحلال وهي عملية اختيار عدد من الأفراد الحاليين في الجيل وإهمال أفراد أخر، الغرض منها هو التقليل من حجم الجيل الناتج لأجل أن تكون مساحة الخـزن مقبولة وتكون الحسابات دقيقة في وقت مقبول نسبيا، بعد عملية الإحلال واختيار أفراد جدد يـتم

إعادة الخطوات الأولى في الخوارزمية الجينية إلى أن تتحقق معايير التوقف والتي يحـــددها مـــن قبل المصمم للخوارزمية الجينية التي تحقق الحل الأمثل للمسألة.(14)

4:الخوارزمية المقترحة:

تم تقديم فكرة تطبيق الخوارزمية الجينية لمسالة الجدولة من اجل الحصول على جدولـــة تضمن خدمة اكبر عدد ممكن من المهام الكلية الموجودة في النظام خلال فترة زمنية محددة. تـــم إنشاء جيل ابتدائي متكون من 50 كروموسوماً كل كروموسوم يمثل أرقام المعالجات الخمسة التي تم فرض وجودها في النظام بترتيبات عشوائية وهذه هي التي ستتم الجدولة على أساســـها. وتـــم فرض وجود 20 مهمة ستتم جدولتها على المعالجات الخمسة بالترتيب المحـــدد مـــن قبـــل كـــل كروموسوم من الـــ50 كروموسوماً في الجيل الابتدائي.وبما أن عدد المهـــام اكبـــر مـــن عـــدد المعالجات فانه تم تكرار استخدام الترتيب المحدد إلى أن تنتهي جدولة كل المهام.

1.4. تمثيل الكروموسوم(Chromosome Representation):

تمثيل الكروموسومات كان على شكل أعداد صحيحة(من1 إلى 5)يمثل كل عدد منهـــا معالجاً من المعالجات الموجودة في النظام فيكون الكروموسوم ممثلا الترتيب الذي ستجدول فيـــه المهام على المعالجات، كما موضح بالشكل (4) .

2	4	1	3	5

الشكل (4) يوضح مثالاً لتمثيل الكروموسوم

2.4 العمليات في الخوارزمية الجينية:

1.2.4:التزاوج(Crossover):تم اختيار رقم عشوائي على أساســـه يـــتم تبـــديل جينـــات فـــي كروموسوم معين مع جينات من كروموسوم آخر والشكل (5) يوضح اثنين من الكروموسومات.

Parent 1	1	5	3	4	2
Parent 2	4	2	3	1	5

الشكل (5) يوضح تمثيل اثنين من الكروموسومات(اثنين من الآباء)

في حالة اختيار الجين الثالث ليكون نقطة التقاطع في عملية التزاوج سيكون الناتج الشكل (6).

Child 1	1	5	3	1	5
Child 2	4	2	3	4	2

الشكل (6) يوضح اثنين من الكروموسومات بعد عملية التزاوج(توليد اثنين من الأبناء)

2.2.4 الطفرة(Mutation): تم استخدام الطفرة من نوع **الجين الجزئي(Partial Gene)** وهي قلب ترتيب المعالج إلى تسلسل آخر .

3.2.4 دالة التقييم (Fitness Function): تم التقييم في هذا البحث على أساس خدمة اكبر عدد ممكن من المهام الموجودة في النظام في زمن محدد(25 وحدة زمن).
في الشكل (7) التسلسل العام لخطوات العمل .

5.التمثيل العملي :

1.5. توليد الجيل الابتدائي(Initialization): تمت هذه العملية بشكل عشوائي(Randomly) حيث تم استخدام دالة توليد القيم العشوائية(rand)، وتم تحديد القيم العشوائية بالنسبة للمصفوفة الأولى بمدى (1 _ 5) على اعتبار أن النظام يحتوي على خمس معالجات، وتحديد القيم للمصفوفة الثانية بمدى(1 _ 20) حيث تم فرض عدد المهام في النظام مساوٍ 20، بعد تكوين الجيل الابتدائي كان الناتج مصفوفتين الأولى ذات قيم عشوائية محصورة مابين الـ1 والـ5 (عدد الصفوف فيها يساوٍ عدد أفراد الجيل الابتدائي(تم اختياره 50) وعدد الأعمدة فيها مساوٍ 5 (عدد المعالجات في النظام)، إن القيم المولدة في هذه المصفوفة تمثل الترتيب الذي ستجدول فيه المهام على المعالجات،والثانية أيضا ذات قيم عشوائية محصورة ما بين الـ1 و الـ20 (عدد الصفوف مساوٍ لـ50 وعدد الأعمدة مساوي لـ٢يمثل العمود الأول رقم كل مهمة من المهام الـ20 بينما العمود الثاني يمثل وقت التنفيذ الخاص بكل مهمة في العمود الأول) .

2.5. تقييم أفراد المجتمع(Evaluation): تم تقييم كل فرد(كروموسوم) من أفراد الجيل الابتدائي العشوائي عن طريق دالة التقييم(Fitness Function) والتي اعتمدت في تقييمها على أعلى عدد للمهام المتنفذة خلال زمن معين يتم تحديده مسبقا, الناتج من التقييم كان عبارة عن مصفوفة عدد الصفوف فيها مساوٍ لعدد أفراد الجيل(50) واثنين من الأعمدة، الأول يمثل تسلسل الكروموسوم في الجيل الابتدائي و العمود الثاني يمثل قيمة الـFitness Function والتي تولدت من تطبيق جدولة المهام بالترتيب الذي حدده ذلك الكروموسوم.تم ترتيب القيم الناتجة من هذه المصفوفة تنازليا.

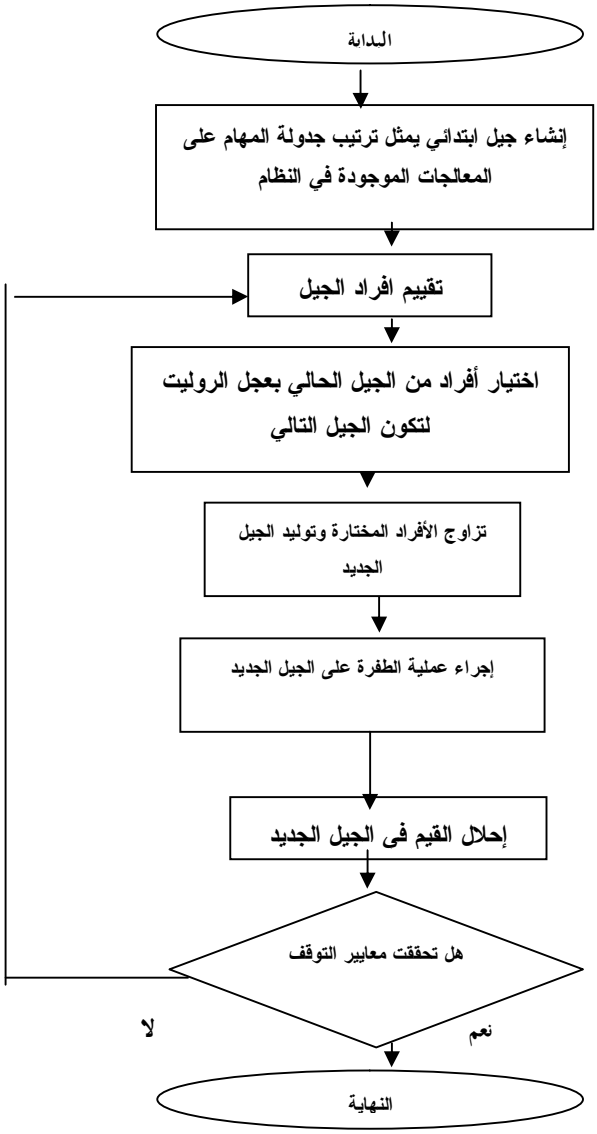
3.5. الاختيار(Selection): تم استخدام عجلة الروليت(Rollet Well) لاختيار الأفراد ذوي

الصلاحية الأعلى(ذوي القيمة الأعلى لدالة التقييم) لإنشاء الجيل التالي، مع عدم إهمال الأفراد ذوي الصلاحية المنخفضة وهذه من مميزات الاختيار بعجلة الروليت، تم بناء دالة sel التي يكون الإدخال لها المصفوفة المرتبةsdif من تسلسلات الكروموسومات و قيم الـFitness Function لتلك الكروموسومات المختارة والإخراج لها يكون مصفوفة من القيم الأصلية للكروموسومات نفسها من الجيل الأصلي التي تم الوصول إليها عن طريق التسلسلات في المصفوفة sdif .

4.5. الوصول إلى القيم الأصلية في الجيل: إن عملية الاختيار تم إجراؤها على التسلسلات للكروموسومات فقط في المصفوفة المرتبة sdif، ولتكوين الجيل فعليا نحتاج للوصول إلى تلك الكر وموسومات لإجراء عمليات التزاوج و الطفرة عليها، وباستخدام الـ index والمتمثل بالتسلسل في المصفوفة sel تم الوصول إلى الكروموسومات في الجيل الأول وتم خزنها في المصفوفة الجديدة new والتي تكون صفوفها مساوية لصفوف الجيل الأول(50) والأعمدة فيه مساوية لعدد المعالجات في النظام(5) وقيمها تحمل الأفراد المختارين لتكوين الجيل التالي .

5.5. التزاوج (Crossover): تمت مزاوجة كل فردين في الجيل الحالي والحصول على فردين جدد مع الاحتفاظ بقيم الأفراد القديمة، تنتج عملية التزاوج جيلاً عدد أفراده ضعف عدد أفراد الجيل الحالي.

6.5. الطفرة(Mutation):اختيار نسبة الطفرة في هذا البحث مساوية(0.01)وهذا يعني انه من بين كل مئة فرد تمت الطفرة على جين في كروموسوم من كروموسومات الجيل المئة.



الشكل (7) يوضح خوارزمية العمل

5 .7. الإحلال(Assignment): من غير المعقول أن تتم مضاعفة أفراد الجيل عند كل تكرار في تنفيذ الخوارزمية الجينية، لذا لا بد من تقليص عدد أفراد الجيل إلى النصف ليمكن الاستمرار في الخوارزمية دون الوصول إلى حجم كبير من البيانات يصعب التعامل معه أو يصعب توفير ذاكرة كافية لاستيعاب حجم الجيل، توجد عدة خوارزميات وطرق للإحلال، اعتمدنا في بحثنا هذا على اخذ نسبة 0.30 من الأفراد ذوي الصلاحية العالية و0.20 من الأفراد ذوي الصلاحية الواطئة، وتختلف هذه النسبة حسب اختيار المبرمج وطبيعة النتائج .

8.5. معايير التوقف(Stop Criteria): خلال ثابت زمني (تم فرضه 25 وحدة زمنية) يتم حساب عدد كل المهام المنجزة من قبل النظام ويكون التوقف في الخوارزمية الجينية عند الوصول إلى أعلى عدد ممكن من المهام المنجزة(لا يكون هنالك فرق واضح بين عدد المهام المنجزة خلال وحدة الزمن مابين تكرار و التكرار الذي يليه).

6. النموذج التطبيقي:

عدد المهام الكلي الحالي في النظام 20 مهمة ويقابل كل مهمة زمن التنفيذ الخاص بها كما موضح بالشكل (8):

tasks	18	11	6	20	1	17	5	2	19	3	9	15	4	12	16	14	7	13	8	10
Execution time	7	3	1	10	4	2	6	9	2	1	1	7	11	5	2	6	5	1	4	5

الشكل(8):يوضح المهام الحالية في النظام و زمن تنفيذ كل مهمة منها.

لتوضيح كيفية حساب دالة التقييم(Fitness Function) استخدمنا احد الكروموسومات المولدة في الجيل الابتدائي لجدولة المهام التي في النظام كما موضح بالشكل (9) :

2	4	1	5	3

الشكل(9) أحد الكروموسومات المولدة في الجيل الابتدائي.

تم اعتماد عدد المهام المنفذة خلال ثابت زمني(تم اعتباره ١٥ وحدة زمنية) كدالة تقييم للخوارزمية الجينية المقدمة في هذا البحث. تقوم الخوارزمية الجينية بتطبيق الجدولة باستخدام كل الكروموسومات المولدة ومن ثم تقييمها.

بالاعتماد على التسلسل السابق للمعالجات ستتم جدولة المهام وكما مبين بالشكل رقم(10):

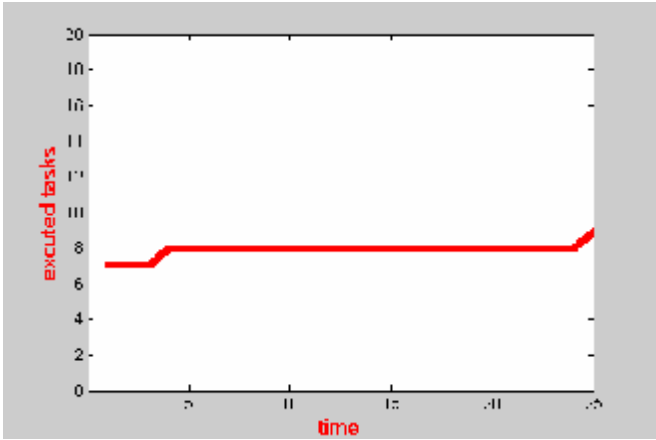
processors	tasks	Execution time	No. of tasks after 15 time unit
P1	6 ,2,4,13	1,9,11,1	2
P2	18,17,9,12	7,2,1,5	4
P3	1,3,16,10	4,1,2,5	4
P4	11,5,15,7	6,3,7,5	2
P5	20,19,12,8	10,2,5,4	2
TOTAL EXCUTION TASKS = 2 + 4 + 4 + 2 + 2 =14			

الشكل(10) :يوضح المعالجات الموجودة في النظام وأمام كل معالج المهام التي سينفذها مع وقت تنفيذ كل مهمة, في حالة تحديد الوقت الزمني الذي سنقيس فيه كفاءة الخوارزمية الجينية ليكون مساوياً لـ 15 فان اكبر عدد للمهام التي سينجزها النظام هي 14 مهمة كما موضح في العمود الرابع من الشكل(مجموع المهام التي ينفذها كل معالج).

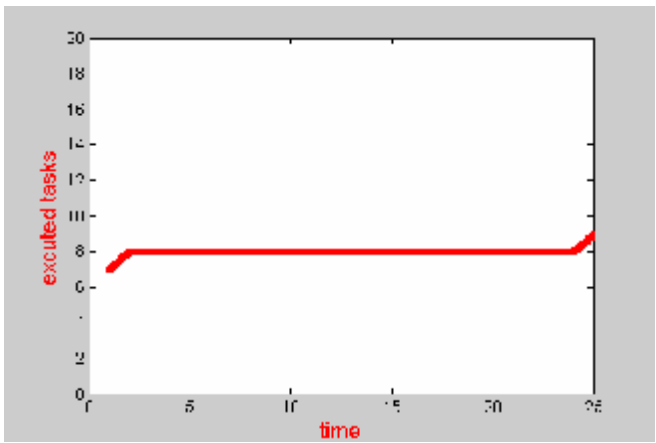
بعد جدولة المهام بخمسين تسلسل مختلف(كروموسوم مختلف) من الجيل الابتدائي تم تقييمها حسب أعلى قيمة لعدد المهام المجدولة في الزمن المحدد وتمت المفاضلة بين الكروموسومات على هذا الأساس بحيث يكون للكروموسوم الذي يحقق اكبر عدد ممكن من المهام المنجزة فرصة اكبر لتكوين الجيل التالي.

7.النتائج: لغرض السيطرة على تنفيذ المهام في النظام وتقييم أداء الخوارزمية الجينية تم توليد 20 مهمة وهذه تم تنفيذها في نظام يحتوي على5 معالجات تعمل بالتوازي,احتمالية التزاوج والطفرة كانت ثابتة(نسبة التزاوج100% ونسبة الطفرة1%) ,تم التوقف في تنفيذ الخوارزمية الجينية بالحصول على نتائج متكررة في أجيال متعاقبة(عندما لا يتغير الحل إلى أفضل).والجدول(1) يمثل النتائج التي تم الحصول عليها,الشكلان 11 و12 يوضحان النتائج دون طفرة, فبعد تكرارين للخوارزمية المقترحة كان عدد المهام المنفذة خلال ثلاث وحدات زمنية هو7 بينما بالتكرار الرابع كان العدد يساوي 8 (تم الحصول على تحسين في إنجاز الخوارزمية للجدولة).أما باستخدام الطفرة فان الشكلين13 و14 يوضحان أن عدد المهام المنفذة بتكرارين للخوارزمية كان 8 خلال 3 وحدات زمن بينما التكرار الرابع أنجز تنفيذ 8 مهام منذ أول وحدة زمن .

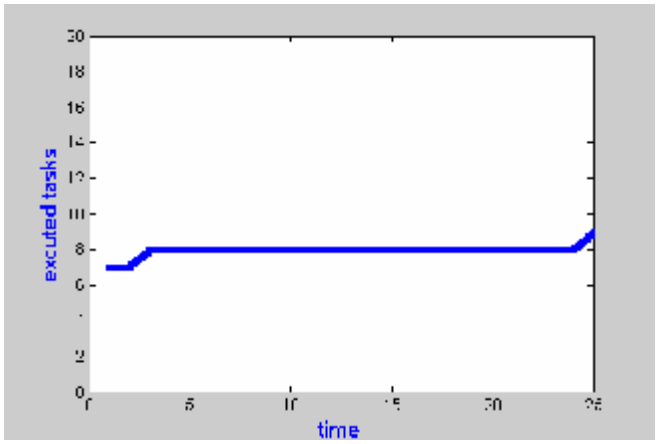
8.الاستنتاجات: إن استخدام الخوارزمية الجينية كان فعالاً جدا في الحصول على نتائج جيدة وهذا واضح من الجدول (1) والذي يستعرض تزايد واضح في عدد المهام المنجزة خلال زمن معين بعد تكرارات بسيطة للخوارزمية الجينية.ولا يمكن إهمال ملاحظة الدور الفعال للطفرة في تحسين النتائج والتسريع بالحصول على الحل الأفضل فمن ملاحظة الجدول (1) و الأشكال (11-14)يمكن الانتباه إلى أن الطفرة أدت إلى إنتاج قيم جيدة بعد تكرارات بسيطة لم تحقق في حالة عدم استخدامها.



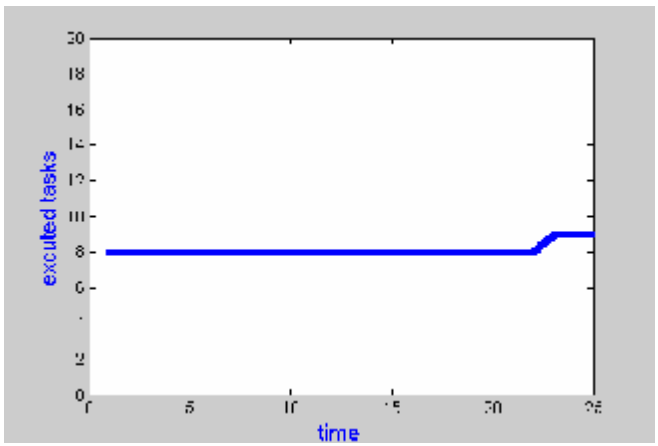
الشكل(11): يوضح عدد المهام المنفذة خلال 25 وحدة زمنية بعد تكرارين للخوارزمية المقترحة(دون طفرة)



الشكل(12):يوضح عدد المهام المنفذة خلال 25 وحدة زمنية بعد أربعة تكرارات (دون طفرة)



الشكل(13):يوضح عدد المهام المنفذة خلال 25 وحدة زمنية بعد تكرارين للخوارزمية (باستخدام الطفرة)



الشكل(14):يوضح عدد المهام المنفذة خلال 25 وحدة زمنية بعد أربعة تكرارات للخوارزمية (باستخدام الطفرة)

الجدول (1): يوضح النتائج التي تم الحصول عليها من تكرارات مختلفة للخوارزمية المقترحة وفي الحالتين باستخدام الطفرة وعدمه.

المهام المنفذة بالترتيب العشوائي في 25 وحدة زمن(بالطفرة)	المهام المنفذة بعد تكرارين في 25 وحدة زمن(بالطفرة)	المهام المنفذة بعد أربعة تكرارات في25 وحدة زمن(بالطفرة)	المهام المنفذة بالترتيب العشوائي في 25 وحدة زمن(دون طفرة)	المهام المنفذة بعد تكرارين في 25 وحدة زمن (دون طفرة)	المهام المنفذة بعد أربعة تكرارات في 25 وحدة زمن(دون طفرة)
8	9	9	8	9	9
8	8	9	8	8	9
8	8	9	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	8	8	8	8
8	8	7	8	8	8
8	8	7	8	8	8
8	7	7	8	8	8
8	7	7	8	7	8
7	7	7	8	7	7

المصادر

[1] Ansari, N.; E. S. H. Hon, and Hong Ren.(1994) "A Genetic Algorithm for multiprocessor scheduling ",**IEEE transaction on parallel and distributed system**,Vol.5,No.2,PP.113-120.

[2] El-Gendy, S. M. (1994) "Task Allocation Using Genetic Algorithms", MS Thesis, University of Louisville.

[3] Gonçalves, J.F.; J. J. de Mendes, and G. C. Maurício (September 2002) "A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem", **AT&T Labs Research Technical Report** TD-5EAL6J**.**

[4] Horn, J.; N. Nafpliotis (2001) "A Genetic Algorithm and Optimization Technique",**IEEE Transactions on Parallel and Distributed Systems**, Vol.8, No.3, PP.0272_284 , http://www.cs. unr.edu/~sushil/papers/conference/newpapers/2001/physics/atomic Processes/poster.pdf

[5] Hou, E. S.; N. Ansari, and H. Ren (1994) "Genetic Algorithm for Multiprocessing Scheduling", **IEEE Transactions on Parallel and Distributed Systems** , Vol. 5, No. 2, PP.113-120.

[6] Kidwell, M.D. and D. J. Cook (1994) "Genetic Algorithm for Dynamic Task Scheduling", **IEEE 13th Annual International Phoenix Conference on Computers and Communications**,PP.61-67.

[7] Kim, Y. C. and Y. S. Hong (1993) "Task Allocation Using A Genetic Algorithm in Multicomputer Systems ", **Proceedings of the 10th IEEE Region Conference on Computer Communication, Control and Power Engineering, NJ**, PP. 258-261.

[8] Lee,Y._H. and C. Ghen (2003) "A modified Genetic Algorithm for Task Scheduling in multiprocessor systems " , Taiwan**, Department of computer Sciences & information Engineering**, **National chiao Tung University**.

[9] Liu,C.L. and J. Layland (1973) "Scheduling algorithms for multiprogramming in A Hard Real-time Environment", **J. ACM**, Vol.20 , No. 1, 1973, pp. 46-61.

[10] Mahmood ,A. "A Hybrid Genetic Algorithm for Task Scheduling in multiprocessor Real Time systems", Bahrain, **Department of computer science ,University of Bahrain**.

[11] Meijer,M. (24th Feb 2003) "Scheduling parallel processes using Genetic Algorithm", **Master thesis in the filed of artificial intelligence, section autonomous systems** . Technical University of Madrid.http://www.ii,s.sinica.edu.twcthpc2003papers CTHPC – 18.pdf

[12] Michael L. Pinedo (2006)**Planning and scheduling in manufactur--ing and services** ,Series: Springer series in operations research and financial engineering,xvi,506p.hardover isban -20:0-387-22-198-0.

[13] Mitchell,M.(1996) **An introduction to genetic algorithms**, In Proceedings of the First European Conference on Artificial Life, Cambridge, MA: MIT Press.

[14] Pai_Chou Wang and W. Kofhage (October 1995) "Process Scheduling With Genetic Algorithm", **Preceding of the 7th IEEE symposium on parallel and distributed** ,PP. 635-641

[15] Rebreyend, P. ; A. Ferreira , and R.C.Correa (1999) "Scheduling multiprocessor Tasks with Genetic Algorithm", **IEEE transaction and parallel and distributed systems**,Vol.10,No.8,88, PP.825-837.

[16] Ramamritham , K.; J.A. Stankovic, and P.F Shiah (1990) "Efficient scheduling Algorithms for Real-time Multiprocessor Systems", **IEEE Transactions on Parallel and Distributed Systems**,Vol.1, No. 2, PP.184-194.

[17] Rinehart, M.; V. Kianzad, and S.S. Bhattacharyya (2004) "A Modular Genetic Algorithm for Scheduling Task Graphs", Technical Report UMTACS_TR_institute for advanced computer sciences, **University of Mayland at college park**. http://www.science.Uva. nlresearchscspapersarchivemei\er2004a.pdf.