

استخدام خوارزمية (Memetic) لجدولة مهام
نظام متعدد المعالجات

الدكتور طارق علي أشهبية

كلية تقنية المعلومات - جامعة بنغازي - الجماهيرية العربية الليبية

الدكتور قصي حميد السلامي

كلية بغداد للعلوم الاقتصادية الجامعة

ملخص

تلعب عملية جدولة المهام دور مهم في تقليل وقت الانتظار اللازم لانتظار المهام في صفوف الانتظار لأجل المعالجة في وحدة المعالجة المركزية (CPU) داخل الحاسوب، حيثُ تصل هذه المهام وتُجدول قبل دخولها إلى أحد المعالجات مع إمكانية التشارك في البيانات الموجودة على الذاكرة المشتركة، ويزداد الأمر أهمية لعملية جدولة مجموعة من المهام أمام n من المعالجات المتوازية. لعلاج مشكلة الجدولة على المعالجات المتوازية تستخدم خوارزمية ميميتك (Memetic Algorithm(MA)) التي هي نسخة مطورة من حاصل دمج الخوارزمية الوراثية (Genetic Algorithm(GA)) وخوارزمية التصليب المُحاكي (Simulated Annealing(SA)).

قدمت (MA) النتائج الفعالة وأفضل الحلول لتخصيص المهام على نظام متعدد المعالجات المتوازية المتجانسة، وذلك باعتمادها على مبدء المستوى الأعلى (Top level) والأسفل (Bottom level) التي قدمت أقل القيم عند حساب قيم المعاملات (Make span) و (Flow time)، وذلك عند المقارنة بين العديد من خوارزميات جدولة متعدد المعالجات.

Abstract:

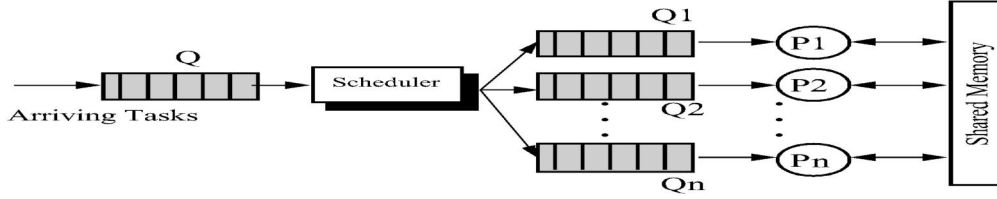
Tasks scheduling plays a very big role in minimizing waiting time, which is required for tasks or jobs in queuing for processing in the computer's (Central Processing Unit (CPU)), where these task are scheduled before they enter one of the processors with the ability to shear existing data in the sheared memory, and this is more important when an (n) tasks are scheduled (bunch of tasks) in parallel processing. Memetic Algorithm (MA) used to solve the scheduling problem in parallel processing, which is a form of genetic algorithm (GA) that are combined with other forms of local search simulated annealing (SA).

MA was introduced an effective results and a best solution for assigning the tasks to the homogeneous parallel multiprocessor system, which dependent on Tlevel and Blevel and they have minimum value when computed the parameters (Make span & Flow time), during the comparison between many of multiprocessors scheduling algorithms.

كلمات مفتاحيه: جدولة المهام، الخوارزمية الجينية (GA)، خوارزمية التصليب المحاكى (SA)، خوارزمية ميميتك (MA).

١. مقدمة:-

تظهر مشاكل الجدولة في العديد من التطبيقات المختلفة، عندما نحتاج إلى إتخاذ قرار لترتيب مهام معينة (كالبرامج الفرعية المتوازية على الحاسوب، نقل الزبائن أو البضائع، تنظيم جداول الامتحانات في الجامعات، ألخ) والتي يجب أن تنفذ بالإعتماد على مصادر محددة (كالعدد المحدد من المعالجات، المركبات، الموظفين، ألخ) مع قيود أخرى (كالتوقيت عند وجود شرط لإنهاء مهمة في وقت محدد أو أن بعض المهام ينتهي وقت الخدمة لها بعد الحصول على مخرجات مهمة أخرى، ألخ)،^[2] حيثُ يُسمح لأي مجموعة من المهام للوصول إلى النظام، كما يوضحهُ شكل رقم (١).



شكل رقم (١) يوضح نظام معالجات متوازية [4]

إن أسلوب الجدولة الكفاء لمعالجة البرامج المتوازية في أنظمة تعدد المعالجات يقلل وقت التنفيذ ويزيد معدل الأداء، ويتم تصنيف هذه الجدولة إلى أصناف مختلفة اعتماداً على خصائص المهام المطلوب جدولتها، خصائص أنظمة متعدد المعالجات وعلى مقدار توفر المعلومات.[6]

تُستعمل خوارزمية (MA) بشكل أساسي لتوجيه عملية البحث حيث يُمكن استعمالها مع بقية خوارزميات البحث المحلي، وفي حالة مزجها مع بعض نحصل على تصميم جديد لخوارزمية هجينة تطويرية.[2]

١,١ مشكلة البحث:- تتمثل في كيفية جدولة مجموعة المهام الواصلة إلى نظام متعدد المعالجات لمعالجة البرامج المتوازية، وذلك لحل مشكلة وقت التنفيذ الطويل ومحاولة الاستفادة من المعالج، والبحث عن العقد التي يجب أن تحقق الشروط التالية:-

١. يجب أن تُعرف وتحقق علاقات الأسبقية بين المهام لتسهيل ترتيب تنفيذ المهام.
٢. كل مهمة تظهر مرة واحدة في الجدولة.
٣. معرفة كلفة الاتصال (وهي كلفة إرسال الرسائل من مهمة على معالج معين إلى معالج آخر)، حيث إن كلفة الاتصال تكون صفر بين أي مهمتين على نفس المعالج.
٤. يُسمح بعمل نسخ من نفس المهمة ليتم تخصيصها على عدة معالجات في نفس الوقت لتقليل كلف الاتصال ووقت الجدولة.

٥. يتكون نظام متعدد المعالجات من عدد محدد من المعالجات المرتبطة بشكل كامل.
مشكلة جدولة المهام الفورية في أنظمة تعدد المعالجات هو أن يُقرر متى ولأي معالج تُعطى المهمة لتنفيذها.

١,٢ هدف البحث :- إن الهدف من جدولة متعدد المعالجات هو إيجاد الخوارزمية المثلى التي تقلل وقت التنفيذ الكلي (وقت التنفيذ + وقت الانتظار أو التعطل) للمهام الفرعية التي تتنافس لإتجازها، وهذا هو الأساس لتحسين إنجاز أنظمة تعدد المعالجات المتوازية، وذلك بتقسيم المهمة إلى مجموعة من المهام الفرعية وتمثيلها كرسوم دائرية مُوجهة ((Directed Acyclic Graph(DAG))، وإيجاد الجدولة لـ (DAG) لمعالجة البرامج المتوازية والذي يقلل وقت التنفيذ والمعالجة ويزيد منفعة المعالج، ولتحقيق هذا الهدف سيتم استخدام خوارزمية ميميتك (Memetic) للتغلب على بعض عيوب الجدولة [4]، من خلال الفرضيات التالية:-

- أي نظام متعدد المعالجات يتضمن (m) من المعالجات المنفصلة المتجانسة (سرعة ومواصفات المعالج متشابهة).
- يتم تمثيل المهام بشكل (DAG).

- مدة التنفيذ المتوقعة معلومة لكل المهام الفرعية وكذلك مدة تحويل البيانات المتوقعة بين المهام الفرعية.
- جدولة المهام لتقليل :

$$\text{Minimize } \underset{j=1}{\overset{n}{\text{Max}}} [\text{finish time } (V_j)] \quad \dots (1)$$

حيث إن الجدولة تُحدد ما هي المهمة الفرعية التي ستنفذ على معالج ما، وما الوقت المطلوب لتنفيذها على ذلك المعالج.^[7]

٢. الجانب النظري:

إن مفهوم الجدولة هو تخصيص مجموعة المهام إلى المصادر المختلفة للحصول على الأداء الأمثل، وذلك بتوزيع المهام بين المعالجات بأسلوب يحافظ على قيود الأسبقية ويقلل وقت التنفيذ الكلي T. إن قيمة T تعتمد على التخصيص المُعطى للمهام وعلى بعض سياسات الجدولة، لتحويل المهام المطلوب معالجتها من حالة الاستعداد (Ready) إلى التنفيذ (Run) في المعالج المناسب.^[٧]

يعمل المُجدول (Scheduler) على معالجة المهام الواصلة للمعالج بأسلوب معين، فإذا عمل المُجدول على أخذ مهام مُعينة من مجاميع معروفة فإن الجدولة تدعى بالجدولة المحددة (Deterministic)، وإذا أخذ مهام بدون تحديد لأي مجموعة تنتمي من بين المهام الكلية فإن المُجدول يدعى (Stochastic). تدعى الوحدة القابلة للجدولة بالمهمة، وإن مولد المهام (Task Generator) قد يكون أي نظام يعمل على تجهيز المُجدول بهذه المهام، تحت افتراض إن عدد المهام هي أكثر بكثير من عدد المعالجات المتاحة، وإن كل مهمة تحتاج إلى وقت مُعين للتنفيذ ومكان مُحدد تتوجه إليه بأمر من المُجدول.^[6]

٢,١ الخوارزمية الوراثية (GA): إن الخوارزمية الوراثية (الجينية) هي طريقة من طرق الكشف والبحث، يُمكن تصنيفها كإحدى طرق الخوارزميات التطورية (Evolutionary Algorithms) التي تعتمد على تقليد عمل الطبيعة من منظور داروين^(١)، وهي أيضاً تقنية بحث تستعمل لإيجاد حلول مضبوطة أو تقريبية لتحقيق الأمثلية، وتصنف كبحوث عالمية استدلالية (Search Heuristics Global)، وهي أيضاً فئة معينة من الخوارزميات التطورية المُعروفة كذلك بالحساب التطوري (Evolutionary Computation) التي تستخدم التكنولوجيا المستوحاة من البيولوجية التطورية (Evolutionary Biology) مثل التوريث والطفرات والاختيار والتجهين (Crossover).^[11] تعتبر الخوارزميات الوراثية من التقنيات الهامة في البحث عن الخيار الأمثل من مجموعة حلول متوفرة لتصميم معين، ويعتمد مبدأ داروين في الاصطفاء حيث تقوم هذه المعالجة الوراثية

^١ تشارلز روبرت داروين (Charles Robert Darwin)^[٧] اشتهر بنظرية التطور ومبدأ الانتخاب الطبيعي، حول نشأة الإنسان. قام بدراسة التحول في الكائنات الحية عن طريق الطفرات وطور نظريته الشهيرة في الانتخاب الطبيعي عام ١٨٣٨ م. التطور هي عملية أدت لظهور جماعات المتعضيات الحية، بشكل عام يؤدي التطور لظهور خلل جديد ومنجدد من جيل لآخر، يؤدي في النهاية إلى تغيير كافة مواصفات النوع قيد التطور مما يؤدي إلى نشوء نوع جديد من الكائنات الحية. مصطلح نشوء عضوي (Organic Evolution) أو النشوء البيولوجي يستخدم غالباً لتفريق هذا المصطلح عن استعمالات أخرى.

بتمرير المزايا المثلى من خلال عمليات التوالد المتعاقبة، وتدعيم هذه الصفات، وتكون لهذه الصفات القدرة الأكبر على دخول عملية التوالد، وإنتاج ذرية مثلى وبتكرار الدورة الوراثية تتحسن نوعية الذرية تدريجياً.

٢,٢ خوارزمية التصلب المحاكى (SA): هي طريقة محاكاة رياضية مبدؤها مستمد من طريقة التصلب المستخدمة كإحدى طرق معالجة المعادن حرارياً، في طريقة المعالجة هذه تتناوب دورات التبريد البطيء وإعادة التسخين (التصلب) بحيث يتم الوصول إلى وضعية تكون الطاقة في أدنى حد ممكن، فهي تهتم بتحقيق الأمثلة الإجمالية في فضاء الحل من خلال اختبار التغيرات العشوائية التي تطرأ على الحلول الفردية، حيث يُقبل أي تغيرات تُزيد من قيمة الأمثلة (Fitness value). يتم البدء بتغيرات ذات نسب عالية نسبياً، والتي تقل بمرور الوقت على طول وقت جدول المهام المُعطى، حيث تُستخدم خوارزمية (SA) مع خوارزمية (GA) القياسية لإجراء ذلك.^[7]

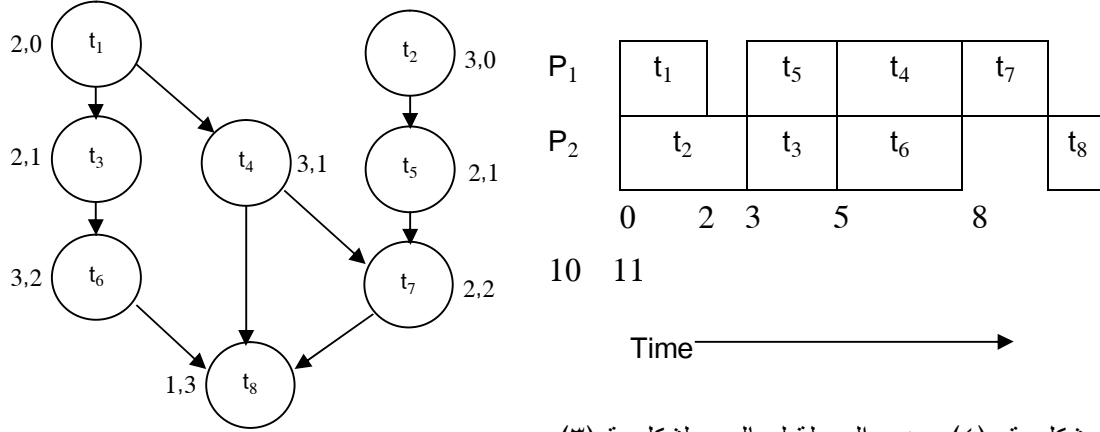
٢,٣ خوارزمية ميميتك (MA): تعتبر (MA) إحدى طرق تحقيق الأمثلة الهجينة والتي يُمكن الحصول عليها من دمج طرق البحث المحلية والعالمية (Local & Global Search) فعند الحصول على جين نقي أثناء البحث المحلي، يتم استبدال النمط الجيني والظاهري للعينة القديمة والسماح فقط بتحديث المعامل الهدف.^[3] إن (MA) هي صيغة مطوره من صيغ (GA) والتي تم دمجها مع غيرها من صيغ البحث المحلي مثل (SA). إن فكرة اختيار ميزة معينة من بين مئات أو آلاف المتغيرات أو الميزات أصبح من الأمور المهمة التي تم التطرق لها بعد التطور السريع لتقنيات قواعد البيانات والحاسوب.^[7]

لمعالجة قواعد البيانات الضخمة التي تعجز عن معالجتها الحواسيب التقليدية، يتم إزالة الميزات العاطلة التي ليس لديها علاقة فيما بينها، وذلك لتحسين أداء الخوارزميات ويُخفف من التكلفة الحسابية عن طريق اختيار الميزة المبتكرة لاختيار الجين الذي يستعمل مفهوم (MA) بشكل خاص، وتدعى هذه الخوارزمية بـ Wrapper-Filter (Feature Selection Algorithm (WFFSA)).^[10]

ففي مشاكل أنظمة تعدد المعالجات، يجب على البرنامج المُعطى أن يجدول في نظام تعدد المعالجات لتقليل وقت تنفيذ البرامج والسرعة في تنفيذه مع أداء أفضل. لقد ظهر استخدام أنظمة تعدد المعالجات للسيطرة على العمليات في المصانع النووية بسبب قابليتها العالية الأداء.

٢,٤ نموذج النظام المستخدم: أي نظام متجانس متعدد المعالجات مُكون من مجموعة (m) من المعالجات المنفصلة $P=\{p_1, p_2, \dots, p_m\}$ المرتبطة من خلال شبكة اتصالات متكاملة ذات صلات متماثلة، ولا يحق لأي مهمة أملاك مُعالج مُعين، بينما يحق للمعالجات الاتصال فيما بينها لمعالجة المهمة عن طريق الشبكة، وسيتم الاعتماد على الجدولة المحددة حيث يفترض أملاكنا جميع المعلومات الضرورية حول المهام وعلاقات الأسبقية التي تربط بينها وأوقات التنفيذ لكل مهمة.

٢,٥ رسم المهمة بيانياً : إن مشكلة رسم جدولة المهام بيانياً ((Task Graph(TG) وبشكل مثالي في نظام متعدد المعالجات مع P من المعالجات، هو أن نخصص المهام الحسابية إلى المعالجات لكي نحافظ على علاقات الأسبقية ونُكمل المهمة بأقصر وقت مُمكن. شكل رقم (٣) و (٤) مثال توضيحي لهذه الحالة.^[7]



شكل رقم (٣) يوضح رسم جدولة

شكل رقم (٤) يوضح الجدولة لمعالجين لشكل رقم (٣) باستخدام مخططات جانب

شكل رقم (٣) يوضح إن الجدولة هي المتجه $s = \{s_1, s_2, \dots, s_n\}$ حيث $s_j = \{t_{i_1}, \dots, t_{i_n}\}$ ، وأن s_j هي مجموعة من المهام n_j المجدولة على المعالج p_j ، وإن وقت التنفيذ الكلي المتحصل من عملية الجدولة يدعى (Make-span)، ويدعى وقت إنتهاء آخر مهمة بوقت إنتهاء الجدولة (FT)، حيثُ يعني الرقم ٣,٠ في شكل رقم (٣) بأن المهمة t_2 في المستوى ٠ وتحتاج ٣ وحدات زمنية، إما فتعني بأن المهمة t_6 في المستوى ٢ وتحتاج ٣ وحدات زمنية.

إن الجدولة في شكل رقم (٤) تقابل الرسم البياني لشكل رقم (٣) حيثُ إن وقت التنفيذ ينتهي في الوحدة الزمنية ١١، وإن الحدود الدنيا لوقت إنتهاء التنفيذ لأي جدولة مبينه على طول المسار الحرج، في شكل رقم (٣) نلاحظ إن $TG=(V,E)$ حيثُ :-

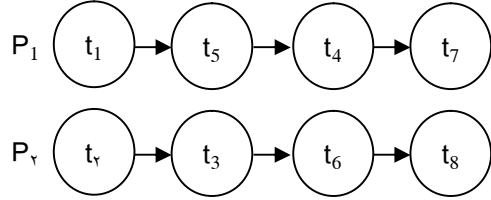
- t_i هي المهمة عند العقدة السابقة، و t_j هو المهمة عند العقدة اللاحقة لكل t_i إذا $e_{ij} \in E$.
- t_i يسبق t_j ، و t_j هو أبن t_i في حالة إن هناك مجموعة من العقد الموجهة من t_i إلى t_j .
- $Pred(t_i)$ هي مجموعة العقد التي تسبق المهمة t_i .
- $Succ(t_i)$ هي مجموعة العقد التي تلحق المهمة t_i .
- $E_t(t_i)$ هو وقت التنفيذ للمهمة t_i .

يُمكن تعريف دالة إرتفاع رسم المهمة وفق العلاقة :-

$$Height(t_i) = \begin{cases} 0 & ,if \quad Pred(t_i) = \emptyset \\ 1 + \max height(t_j), & otherwise \quad t_j \in Pred(t_i) \end{cases} \quad \dots (2)$$

والتي تستخدم لوصف أسبقية العلاقات غير المباشرة بين المهام، فإذا كانت المهمة t_i تسبق t_j فهذا يعني $height(t_i) < height(t_j)$ وإذ لم يكن هناك مسار يربط بين المهمتين فإنه ليس هناك أسبقيات للعلاقات بينهما ويتم تنفيذها بشكل عشوائي.

يُمكن تمثيل الجدولة على شكل مجموعة من قوائم المهام الحسابية، كما في شكل رقم (٥)، وهذه القوائم تقابل عمليات التنفيذ على



شكل رقم (٥) يوضح تمثيل القوائم في

المعالج، وترتيب المهام في كل قائمة يُشير إلى ترتيب التنفيذ في المعالج، هذا الترتيب يسمح لنا بمعرفة العلاقات الأسبقية بين المهام التي تنفذ في المعالج ونهمل العلاقات بين المهام التي تنفذ في معالجات أخرى.

٢,٦ القيم الأولية للكورموسومات (Population): - يتم توليد القيم الأولية في الخوارزمية الوراثية بشكل عشوائي من عقد البحث، حيث يتم افتراض شرط الارتفاع التالي على توليد الجدولة: (قائمة المهام ضمن كل معالج جدولته منظم في ترتيب تصاعدي للارتفاعات).^[7]

مثال ذلك رسم المهمة الموضح في شكل (٣)، يبين أن t_5 ($height=1$) في المستوى الأول وهي تسبق المهمة t_8 ($height=3$) التي هي في المستوى الثالث، وفي حالة أنهما سينفذان في نفس المعالج فإن t_5 تسبق t_8 ويتم تنفيذها أولاً كونها في المستوى الأول، بينما t_5 و t_6 ليس بينهما علاقة أسبقية لهذا يتم تنفيذهما بأي ترتيب. يُمكن استخدام الخوارزمية التالية لتوليد القيم الأولية:

Algorithm Generate-Schedule // {Generates a schedule of task graph TG for multiprocessor system with p processors}

1. [Initialize] compute height for every task in TG.
2. Separate tasks according to their height.
3. Loop p-1 times for each of first p-1 processor, do step 4.
4. Form the schedule for a processor.
5. Last processor assigns remaining tasks in the set to last processor.

لقد استخدمت تقنية التزاحم (Crowding) في الخوارزميات الوراثية لإبقاء التنوع في الكورموسومات ولمنع التقارب الغير مرغوب للحلول المحلية المثلى، ويشمل ذلك تزواج كل نسل مع فرد مماثل في الكورموسومات الحالي ويقرر أي من الاثنين الذي سيبقى في الكورموسومات الجديد.^[8, 6]

٢,٧ دالة الأمثلية **Fitness Function**: - هي دالة الهدف التي نرغب بجعلها أمثل ما يمكن لهذه المشكلة، والتي تستخدم لتقييم عقد البحث والسيطرة على العمليات الوراثية. لمشكلة جدولة متعدد المعالجات، يُمكن افتراض أن هناك عوامل مثل الطاقة الإنتاجية، وقت إنهاء المهمة ومنفعة المعالج، أن الخوارزمية الوراثية مستندة على انتهاء وقت الجدولة (S) والذي يُعرف بالشكل:-

$$\text{Flow time: } FT(S) = \max_{P_j} ftp(p_j) \dots (3)$$

$$\text{Make span: } \sum_{P_j} ftp(p_j) \dots (4)$$

حيثُ $ftp(p_j)$ هو وقت الانتهاء لأخر مهمة في المعالج p_j ، ولتعظيم دالة الأمثلية فأننا بحاجة إلى تحويل وقت الانتهاء إلى صيغة التعظيم، يتم ذلك بتعريف القيمة المثلى للجدولة (S) وهي $C_{max} - FT(S)$ ، حيثُ C_{max} هو الحد الأعلى لوقت انتهاء التنفيذ، وإن الجدولة المثالية ستملك أقل وقت انتهاء وأكبر قيمة مثلى مقارنة مع الجدولة الأخرى.^[4]

٢,٨ عوامل التشغيل الوراثية (Genetic Operators) : تتمثل في تكوين عقد بحث جديدة مستندة على البيانات الحالية لعقد البحث، وذلك بدمج التراكيب الجديدة من عقدي البحث للحصول على عقدة واحدة، وتفترض وجود علاقات أسبقية ما بين المعالجات، بالإضافة إلى ضمان اكمال انجاز المهام وتفردتها بالجدولة، وقد يتم دمج مجموعة من الأجزاء المثالية في جدولة متعدد المعالجات لتكوين جدولة مثالية.^[7]

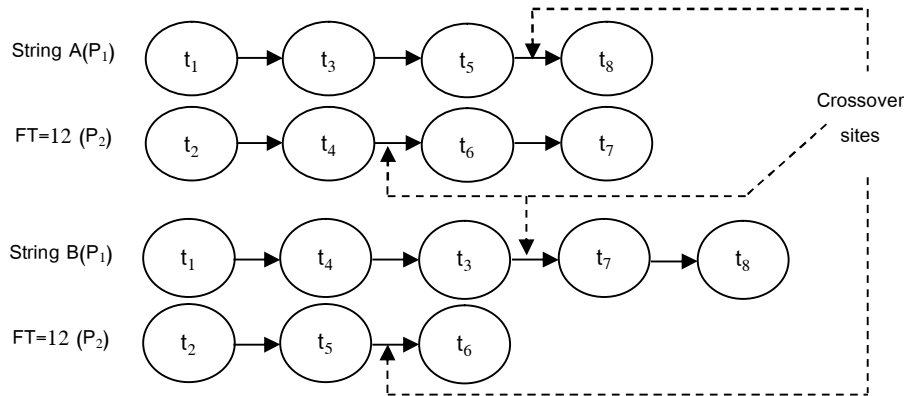
٣. الجانب التطبيقي:-

٣,١ الانتقال Crossover :- يُمكن أن تُكون سلسلتين جديدتين بتبادل الأجزاء بين السلسلتين مع ضمان بقاء تحقق علاقات الأسبقية واكتمال تنفيذ المهام وتفردتها بعد إجراء عملية الانتقال، كما في شكل رقم (٦ و ٧) وذلك بعمل التالي :^[7]

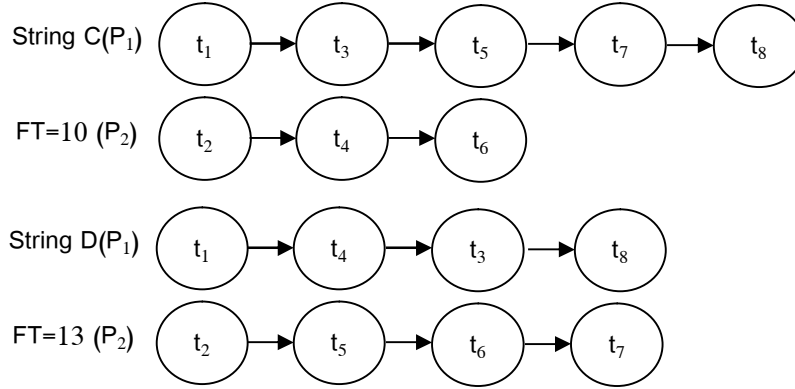
١. نختار موقع الانتقال حيثُ يُمكن أن نقطع القوائم إلى نصفين.

٢. أبدال الأنصاف الأولى للمعالج P_1 للسلسلة A و B.

٣. أبدال الأنصاف الأولى للمعالج P_2 للسلسلة A و B.



شكل رقم (٦) يوضح سلسلتين للعلمية المنقولة



شكل رقم (٧) يوضح توليد سلسلتين جديدتين بعد إجراء العملية المنقولة

٣,٢ الاستنساخ (Reproduction):- هي عملية استنساخ بيانات لسلاسل جديدة بالاعتماد على القيم المثلى لبيانات السلاسل القديمة، وإن معيار الاختيار هو أن السلسلة ذات القيمة المثلى العالية يكون لديها الفرصة الأعلى للبقاء في السلسلة الجديدة، ويُمكن إجراء تغيير بين العقد ذات المستوى الواحد خلال جدول متعدد المعالجات.^[7]

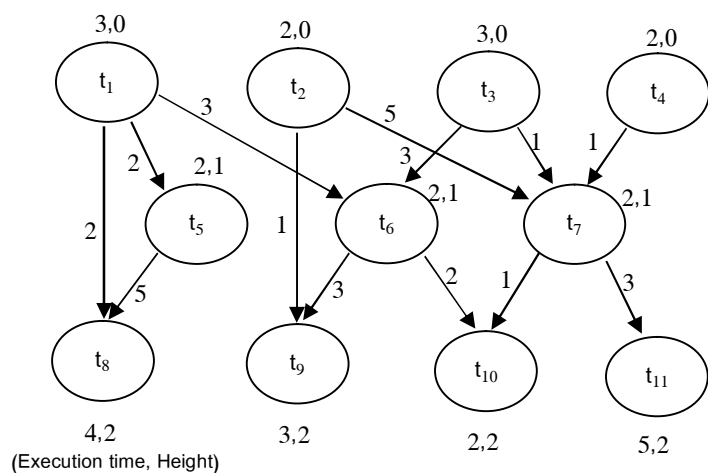
٣,٣ استخدام الخوارزمية الوراثية :-

//Algorithm Find-Schedule

1. Initialize.
2. Repeat step 3 to 8 until algorithm is convergent.
3. Compute fitness value for each string in the initial population.
4. Perform reproduction, store string with highest fitness values in Best-string.
5. Perform crossover.
6. Perform mutation.
7. Preserve the best string in Best-String.

تقوم طريقة الخوارزميات الجينية على توليد حلول جديدة تولد حلولاً من احتمالات مشفرة على الشكل المعروف بالكروموسوم أو المورث، الكروموسومات تجمع أو تتغير لإنتاج الأفراد الجدد، وهي مفيدة لإيجاد الحل الأمثل للمعضلات المتعددة الأبعاد التي يمكن أن تشفر القيم للمتغيرات المختلفة فيها على شكل الكروموسوم.^[11]

٣,٤ شرط الانتهاء:- إن أفضل حل للبيانات المكتسبة بواسطة هذه الخوارزمية لا يتغير بعد عدد محدد من الأجيال، ولكن يفترض هنا إن الخوارزمية تتقارب للحل الأمثل.^[7]



شكل رقم (٨) مثال آخر لرسم جدولة المهام DAG

شكل رقم (٨) هو مثال آخر لرسم جدولة المهام (DAG)، حيث يتم تعريف المستوى الأعلى $Tlevel(T_i)$ بأنه طول أطول مسار في رسم جدولة المهام من أي مهمة إلى (T_i) متضمناً كلف التنفيذ لـ (T_i) ، وبشكل مشابه نُعرف المستوى الأدنى $Blevel(T_i)$ بأنه طول أطول مسار من أي مهمة (T_i) إلى خروج تلك المهمة متضمناً كلف التنفيذ لـ (T_i) ، ويُمكن حسابهما من الصيغ التالية: [4]

$$Tlevel(T_i) = \max_{T_j \in pred(T_i)} \{Tlevel(T_j) + W_j + c_{ij}\} \dots (5)$$

$$Blevel(T_i) = W_i + \max_{T_j \in succ(T_i)} \{Blevel(T_j) + c_{ij}\} \dots (6)$$

Task No.	Execution Time	Completion Time	Bottom Level	Top Level	Order of execution According to Execution Time	Order of execution According to Completion Time	Order of execution According to Bottom-Level	Order of execution According to Top-Level
1	3	3	16	0	7	2	2	1
2	2	2	17	0	1	1	1	2
3	3	3	14	0	8	3	3	3
4	2	4	13	0	2	4	4	4
5	2	5	11	5	3	5	5	5
6	2	7	8	6	4	7	7	6
7	2	7	10	7	5	6	6	7
8	4	9	4	12	10	8	9	10
9	3	12	3	11	9	9	10	9
10	2	14	2	10	6	11	11	8
11	5	12	5	12	11	10	8	11

جدول رقم (١) يوضح أولوية تنفيذ المهام بالاعتماد على وقت تنفيذها، وقت أنتهاءها، المستوى الأدنى والأعلى

P1	T2	T4	*	*	T6	T9					
Time	0	2	4	7	10	12	15				
P2	T1	T5	#		*	T11	*	*	T10		
Time	0	3	5		11	14		19	21	22	24
P3	T3		*	*	T7	*		*		T8	
Time	0	3		8	9	11	13		18		22

شكل رقم (٩) يوضح كيفية تخصيص المهام للمعالجات باستخدام مبدأ المستوى الأدنى Blevel

Make span = 24 time units

Flow time = 3+2+3+4+5+12+11+22+15+24+19 = 120 time units

P1	T1	T4	*	*	T7	*	T8					
Time	0	3	5		10	11	13		18	22		
P2	T2	#	*	T5	#	*	#	*	T10	*	T11	
Time	0	2	3	5	7	8	10	13	14	16	19	24
P3	T3	*	T6	*	T9							
Time	0	3	6	8	9	12						

شكل رقم (10) يوضح كيفية تخصيص المهام للمعالجات باستخدام مبدأ المستوى الأعلى Tlevel

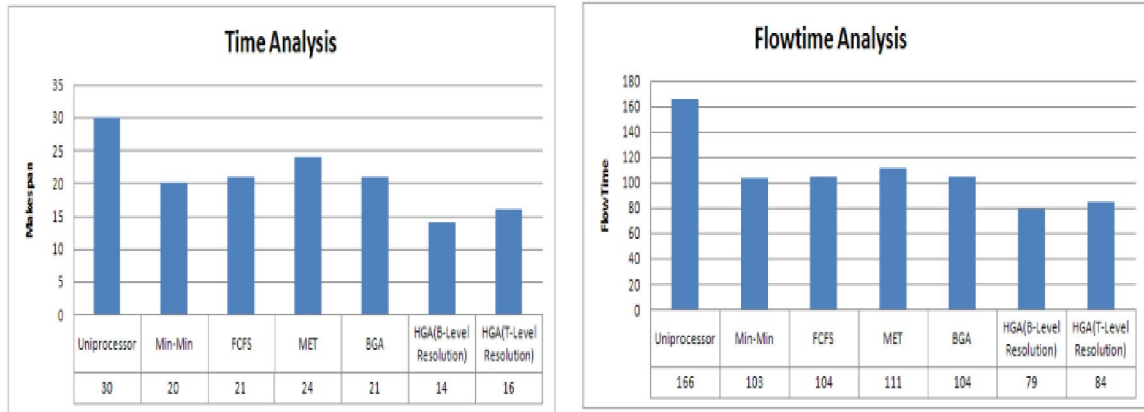
Make span = 24 time units

Flow time = 3+2+3+5+7+8+13+22+12+16+24 = 115 time units

حيث * تشير إلى وقت الإتصال (Communication time) و # تشير إلى وقت الأنتظار (Waiting time).

وبمقارنة النتائج أعلا مع نتائج عدة خوارزميات جدولة^[6]، نحصل على شكل رقم (١١) الذي يوضح قيم

المعاملات (Make span) و (Flow time). [4 pp.195]



شكل رقم (١١) يوضح قيم المعاملات (Make span) و (Flow time) لعدة خوارزميات جدولة

٣,٥ خوارزمية جدولة متعدد المعالجات الفعالة المستندة على الخوارزمية الوراثية :- يتم الاعتماد على التوليد العشوائي لتطبيق الاستنساخ، حيث إن كل سلسلة من البيانات يحتل حجم نسبي من القيمة المثلى، ويتم توليد الأرقام العشوائية لتقرير أي سلسلة ستكون في التوليد الوراثي القادم، حيث إن السلسلة ذات الامثلية الأعلى يكون لها النصيب الأكبر في الاختيار للتوليد القادم.^[7]

يُمكن إجراء تعديل بسيط على عملية الاستنساخ الأساسية وذلك بالسماح للسلسلة الجيدة في عملية التوليد الحالية للعبور إلى السلسلة الوراثية الجديدة، هذا التعديل سيزيد من أداء الخوارزمية الوراثية، والتي تنجز عملية الاستنساخ لبيانات السلسلة (POP) وتولد بيانات سلسلة جديدة (NEWPOP) بالصيغة التالية:-

1. Initialize: let NPOP ← number of string in POP.
2. Construct: NSUM ← sum of all the fitness value of the string in POP, form NSUM slots and assign string to the slots according to the fitness value of the string.
3. Loop NPOP – I times: Do step 4 NPOP – I times.
4. Generate a random number between I and NSUM and use it to index into the slots to find the corresponding string, add this string to NEWPOP.
5. Add the best string: add the string with the highest fitness value in POP to NEWPOP.

٣,٦ خوارزمية جدولة متعدد المعالجات المعدلة المستندة على خوارزميات Memetic :- لهذا الغرض استخدمت خوارزمية (SA) وهي تتعلق بتقنية الأمثلية الإجمالية التي تفترض إن فضاء البحث هو باختبار التغيرات العشوائية على الحل الفردي، حيث يتم قبول أي تغيير يزيد من القيمة المثلى، ويتم ذلك باستخدام (SA) مع (GA) القياسية، حيث يتم البدء بنسب عالية من التغيرات وتتناقص بمرور الوقت على طول وقت الجدولة. إن الدمج بين خوارزميتي (GA) و (SA) يولد لنا خوارزمية (Memetic)، وفي الحقيقة هي تحسين لأداء هذه الخوارزميتين، ويمكن استعمالها مع العمليات الوراثية الأخرى.^[7]

Algorithm Modified-Find-Schedule

1. Initialize.
2. Repeat steps 3 to 8 until algorithm is convergent.
3. Compute fitness values for each string in the initial population.
4. Arrange the chromosomes according to decreasing order of their fitness values.
5. Discard p lowest chromosomes.
6. Perform reproduction on the remaining chromosomes. Store string with highest fitness values in Best-String
7. Perform crossover.
8. Perform mutation.
9. Preserve the best string in Best-String.

تعتمد قيمة p في هذه الخوارزمية على العدد الكلي من الكروموسومات التي تم توليدها كبيانات أولية وعلى الشروط النهائية. أما خوارزمية (MA) فلقد مرت بعدة أجيال^[5]، فالجيل الأول كان يدعى (Hybrid Algorithms)، وهي ناتج دمج للبحث العالمي الأساسي والمرحلة التطويرية من خلال الشفرة التالية:-

Procedure Memetic Algorithm**Initialize:** Generate an initial population;**While** Stopping conditions are not satisfied **do**

Evaluate all individuals in the population.

Evolve a new population using stochastic search operators.

Select the subset of individuals, Ω_{il} that should undergo the individual improvement procedure.**For** each individual in Ω_{il} **do**Perform individual learning using meme(s) with frequency or probability of f_{il} for a period of t_{il} .

Proceed with Lamarckian or Baldwinian learning.

End for**End while**

أما الجيل الثاني لـ (MA) فكان يدعى (Multi-meme) أو (Hyper-heuristic) وهو يعرض مبادئ الإرسال واختيار الـ (Memetic) في تصميمها، وتعتبر مادة الـ (Memetic) هي نوع من أنواع الشفرات الجينية، ويدعى الجيل الثالث (Co-evaluation) أو (Self-generation) والذي هو يعتمد على البحث المحلي كأساس لإكمال الحلول المرشحة للجيل الجديد ضمن النظام التطوري.^[9]

٤. الاستنتاجات :

إن مشكلة جدولة المهام التي ستنفذ على نظام متعدد المعالجات هو أحد أكثر المشاكل صعوبة على المعالجات المتوازية، ويسبب امتلاك الخوارزمية الوراثية بعض العيوب، تم دمجها مع خوارزمية التصليب المحاكى لتحسين الأداء العام للنظام، وهذا ما أطلق عليه بخوارزمية ميميتك (MA)، حيث قدمت (MA) أفضل النتائج وأقلها باعتمادها على مبدء المستوى الأعلى (Tlevel) والأسفل (Blevel) بمقدار ١٤ و ١٦ وحدة عند حساب قيم المعامل (Make span) وبمقدار ٨٤ و ٧٩ وحدة عند حساب قيم المعامل (Flow time)، وذلك عند المقارنة بين العديد من خوارزميات جدولة متعدد المعالجات. فلقد قللت (MA) فترات توقف (CPU) وذلك عن طريق تخصيص المهمة القادمة وبشكل سريع للمعالج العاطل ومنع عملية التزامن التي قد تحدث بين المهام.

٥. التوصيات :

من أهم التوصيات التي يُمكن الإشارة إليها :-

١. اعتماد بيانات حقيقية لوصول المهام لمتعدد المعالجات مأخوذة من شركات عملاقة كشركة Intel و AMD لتقييم سلوك خوارزمية (MA) واعتمادها كطريقة جدولة فعالة.
٢. دراسة طرق جدولة متعدد المعالجات المستخدمة حالياً ومحاولة تطوير خوارزمية هجينة أسوه بخوارزمية (MA)، مع عمل مقارنة لتحديد الخوارزمية الأكثر كفاءة في استغلال وقت المعالج.

References

- 1- Albert Y. Zomaya, Matthew Clements, and Stephan Olariu, **A Framework for Reinforcement-Based Scheduling in Parallel Processor Systems**, *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, VOL. 9, NO. 3, MARCH 1998, 249.
- 2- Boris Mitavskiy, Jun He, **A Polynomial Time Approximation Scheme for a Single Machine Scheduling Problem Using a Hybrid Evolutionary Algorithm**, Aberystwyth University, U.K., February 9, 2012, <http://arxiv.org/abs/1202.1708v1>
- 3- Evgenia A. K., Varvara G. A. and Kyriakos C. G., **An Asynchronous Metamodel-Assisted Memetic Algorithm for CFD-based shape optimization**, National Technical University of Athens, Greece, *Engineering Optimization*, Vol. 00, No. 00, September 2010, 1–21.
- 4- Kamaljit K., Amit C. and Gurvinder S., **Heuristics Based Genetic Algorithm for Scheduling Static Tasks in Homogeneous Parallel System**, *International Journal of Computer Science and Security (IJCSS)*, Volume (4): Issue (2), pp.183, India, 2010.
- 5- Kendall G. and Soubeiga E. and Cowling P., **Choice function and random hyper-heuristics**, *4th Asia-Pacific Conference on Simulated Evolution and Learning SEAL 2002*: 667--671.
- 6- Kusay H. Alsalamy, **The use of operations research techniques to improve queues scheduling efficiency to enhance computer processing speed**, Ph.D. Thesis, Omdurman Islamic University, Faculty of Economics and Political Science, Statistics Department, 2012.
- 7- Sanjay R Sutar, Jyoti P. Sawant, Jyoti R. Jadhav, **Task Scheduling For Multiprocessor Systems Using Memetic Algorithms**, INDIA, 2006, <http://www.comp.brad.ac.uk/het-net/tutorials/P27.pdf>.
- 8- Severino F. Galan, Ole J. Mengshoel, **Generalized Crowding for Genetic Algorithms**, <http://www.sigevo.org/gecco-2010/paper-presentation-schedule.html>.
- 9- Smith J. E., **Coevolving Memetic Algorithms: A Review and Progress Report**, *IEEE Transactions on Systems Man and Cybernetics - Part B* 37 (1) 2007: 6--17.
- 10- Zexuan Zhu, Y. S. Ong and M. Dash, **Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework**, *IEEE Transactions on Systems, Man and Cybernetics - Part B* 37 (1) (2007): 70--76.
- 11- <http://en.wikipedia.org/wiki/>, 2010.