

تطوير خوارزمية جديدة لجدولة مهام متعددة المعالجات المباشرة باستخدام تقنيات بحوث العمليات

Developing New Online Multiprocessor Scheduling Algorithm using Operations Research Techniques

د. قصي حميد السلامي¹ ، أ. أحمد محمد رؤوف²
¹ جامعة جيهان - قسم إدارة الأعمال - أربيل - العراق.
² جامعة بنغازي - كلية تقنية المعلومات - قسم شبكات واتصالات الحاسوب - بنغازي - ليبيا.
qalsalami72@Yahoo.com

Received: 13.03.2013

Accepted: 21.08.2013

ABSTRACT- This paper details the process of developing a new algorithm for multi-processors task scheduling, and the simulation of such algorithm using one of queuing theory's models – from operations research science – to evaluate the behavior and efficiency of the new algorithm, and to compare simulation results with the ones from other common multi-processors task scheduling algorithms, using parameters from queuing theory in operations research. The comparison includes an analysis of CPU's behavior when receiving a number of jobs on four random-job-duration patterns (completely random, ascending random, descending random & inconstant random), using charts and diagrams, formed from each case's data using Excel 2010, which showed the convergence and divergence between those algorithms at the different patterns. The results of the comparison confirmed the efficiency of the new algorithm in enhancing and minimizing the waiting duration for each job at the total job queue of the CPU.

Keywords: Scheduling, Queuing Theory, Multi-Processors, Efficiency

المستخلص- تناقش هذه الورقة البحثية كيفية أعداد وعمل خوارزمية جديدة لجدولة مهام متعددة المعالجات أسوة ببقية خوارزميات الجدولة، ومن ثم العمل على محاكاة عمل تلك الخوارزمية بالاعتماد على إحدى نماذج نظرية صفوف الانتظار في علم بحوث العمليات لتقييم سلوك وكفاءة هذه الخوارزمية، ومقارنة نتائجها مع نتائج محاكاة البعض من طرق جدولة متعددة المعالجات المتعارف عليها، بهدف دراسة سلوك عمل المعالج في حالة استقباله عدداً من الأعمال على أربع فترات عشوائية مختلفة (عشوائية، تصاعدي، تنازلي، متقلب)، وتحليل سلوك كل حالة معتمدين على الأشكال البيانية باستخدام البرنامج الإحصائي (Excel 2010) التي تبين أماكن التقارب والاختلاف فيما بين هذه الخوارزميات خلال الفترات والقيم العشوائية المختلفة، حيث تمت المقارنة بالاعتماد على معاملات نماذج نظرية صفوف الانتظار، التي أظهرت كفاءة الخوارزمية المقترحة في تحسين سرعة معالجة الحاسوب زيادتها.

الكلمات المفتاحية: بحوث العمليات، صفوف الانتظار، خوارزمية KA، خوارزميات الجدولة، متعدد المعالجات.

تمهيد
تعتبر وحدة المعالجة المركزية الوحدة المسؤولة على تنفيذ كافة العمليات الحسابية والمنطقية (تنفيذ البرامج)، ويفترض في هذه الوحدة أن تكون عاملة طوال الوقت (ما أمكن)، وعادة تقاس كفاءة استخدام وحدة المعالجة (CPU Utilization) [3] كنسبة للمهام التي تقع في هذه الوحدة والتي هي في حالة تنفيذ [1] حيث تكون خوارزميات جدولة المهام مسؤولة على توزيعها على المعالجات.

مشكلة البحث: هناك جدل كبير حول أفضلية استخدام عدد من المعالجات على شريحة واحدة والتي تدعى بمعالج متعدد الأنوية (Multi-Core Processor) أو استخدام معالجات منفصلة (Separate Processors) على لوحة أم واحدة (Mother Board)، بشرط ضمان توفر ناقل بيانات بسرعة عالية

جداً الجدل يدور حول كيفية تحديد طريقة جدولة المهام التي تنتظر الدخول للمعالجات لأجل المعالجة، فالمشكلة هي كيفية إيجاد الخوارزمية المثلى التي تقلل وقت التنفيذ الكلي للمهام الفرعية التي تتنافس لإنجازها، وهذا ما سيتم دراسته والتركيز عليه بشكل مستفيض في هذا البحث.

هدف البحث: هو إجراء محاكاة لعمل متعدد المعالجات، وذلك بتطبيق أحد أساليب بحوث العمليات لتحسين كفاءة جدولة صفوف الانتظار (Queuing Theory)، بهدف زيادة سرعة معالجة الحاسوب وذلك بإعداد خوارزمية جديدة مقترحة لحل مشكلة جدولة المعالج أسوة ببقية خوارزميات الجدولة، ودراسة سلوكها ومدى تميزها عن بقية الخوارزميات.

خوارزميات جدولة متعدد المعالجات [9]

إن جدولة المهام هي عملية توزيع موارد المعالجات على المهام والنشاطات للوصول إلى الوضع المثالي لعمليات المعالجة. كأي عملية تأتي جدولة المهام لحل مشكلة ما، وتتحدد خصائص (مشكلة) جدولة المهام من العناصر الآتية: خصائص المهام والأنشطة المطلوب جدولتها، وخصائص المعالجات، والقيود على عمليات المعالجة، ومعايير الأمثلية المطلوب. تتوفر عدة طرق لجدولة العمليات وعادة ينفذ جدول العمليات طريقة واحدة منها، ومن بينها [2]:

خوارزمية جراهام لجدولة القوائم Graham's List Scheduling

التي ظهرت في أواسط الستينيات، إذ تقوم هذه الخوارزمية على إسناد أي مهمة جديدة إلى المعالج الأقل حملاً، على افتراض أن المهام تُعطى في قائمة متسلسلة.

خوارزمية بارتال Bartal's Algorithm

فكرة العمل هي الحفاظ على انعدام موازنة الأحمال بين المعالجات، والاحتفاظ بعدد منها مع عدد قليل من المهام (بشكل أكثر دقة حوالي 44.5% على الأقل من المعالجات يجب أن تكون قليلة المهام (خفيفة))، بحيث أنه في حالة ظهور مهمة كبيرة سيكون من السهولة إسنادها إلى أحد المعالجات الخفيفة.

خوارزمية كارجر Karger's Algorithm

تعمل على اختيار معالج معين من ضمن قائمة موحدة وفقاً لشروط معينة تحقق عدم موازنة الأحمال على المعالجات، حيث تحتفظ الخوارزمية دائماً بقائمة محدثة للمعالجات مرتبة ترتيباً تصاعدياً وفقاً لارتفاعها.

خوارزمية ألبيرس Albers's Algorithm

تُحافظ الخوارزمية في أي لحظة على تحديث قائمتين: قائمة L_1 المكونة من $(m/2)$ للمعالجات الخفيفة، وقائمة L_2 المكونة من $(m/2)$ للمعالجات المشغولة، حيث يتم توجيه المهمة الجديدة إلى أقل المعالجات انشغالا إما في L_1 أو L_2 بناءً على النسبة بين مجموعي الارتفاعات في القائمتين، وتقوم بالاحتفاظ بقائمة بجميع المعالجات مرتبة ترتيباً تصاعدياً بناءً على الارتفاع في ذلك الوقت.

خوارزمية فليشر Fleischer's Algorithm

تقوم هذه الخوارزمية بالاحتفاظ بقائمة بجميع المعالجات مرتبة ترتيباً تنازلياً بناءً على الارتفاع بدلاً من الترتيب التصاعدي للخوارزميات السابقة.

إن مشكلة جدولة المهام هي عنصر أساسي لجعل أنظمة تعدد المعالجات المتوازية ذات أداء أفضل، ويتم ذلك بأن يُقرر متى ولأي معالج تُعطى المهمة لتنفيذها. إن مفهوم الجدولة ببساطة هو تخصيص مجموعة المهام إلى المصادر المختلفة للحصول على الأداء الأمثل، ويتم ذلك بتوزيع المهام بين المعالجات بأسلوب يحافظ على قيود الأسبقية ويقلل وقت التنفيذ الكلي (Total time (T)). [10]

تُدعى الوحدة القابلة للجدولة بالمهمة، وإن مولد المهام (Task Generator) يعمل على تجهيز الجدول بهذه المهام، تحت افتراض أن عدد المهام هي أكثر بكثير من عدد المعالجات

المتاحة. إن مولد المهام قد يكون أي نظام يعمل على تزويد مجموعة المهام إلى الجدول، وأن كل مهمة تحتاج إلى وقت معين للتنفيذ ومكان مُحدد تتوجه إليه بأمر من الجدول. [5]

لقد عمل [8] Susanne A. في عام 2000 على إجراء تجارب عملية لإختبار كفاءة أربع خوارزميات جدولة هي (List, Bartal, Albers, Karger) وذلك باستخدام أولاً بيانات لمهام حقيقية مأخوذة من عدة شركات حاسوب عملاقة، وثانياً بيانات تم توليدها من عدة دوال احتمالية، لدراسة مدى تأثير ذلك على سلوك هذه الخوارزميات فيما بينها، وقد لاحظ أن ازدياد معدل التحميل على المعالجات يعتمد على مقدار المهام الواصلة للنظام، هل هي أقل أو قريبة أو أكثر من مقدار معدل التحميل لأن ذلك يؤثر بشكل مباشرة على أسلوب عمل كل خوارزمية وكيفية جدولة المهام على المعالجات. أن الشيء المميز الذي تم ملاحظته، أن جميع الخوارزميات تميل إلى حالة الاستقرار في سلوكها بشكل سريع عند استخدام البيانات الاحتمالية، بخلاف حالة الاستقرار المتأخرة التي قد يصل إليها النظام في حالة استخدام البيانات الحقيقية، ويعود السبب إلى أن البيانات الاحتمالية تسلك سلوك محدد يعتمد على الدالة الاحتمالية المستخدمة، بخلاف البيانات الحقيقية التي قد تسلك سلوك عشوائي لا يميل إلى دالة محددة، وقد أشار في نهاية دراسته إلى كفاءة خوارزميتي (Bartal & Albers) بسبب سلوكها المستقر مقارنة ببقية الخوارزميات.

وقد قارن [7] Raihan A. في 2004 بين النسبة التنافسية لخمس من خوارزميات الجدولة هي (List, Bartal, Albers, Karger, Fleischer)، وقد بين كيفية تقسيم المعالجات إلى مجموعات ذات حمل ثقيل والأخرى ذات حمل خفيف لأسباب تتعلق بكل خوارزمية، وفي نهاية دراسته أشار إلى كفاءة (Fleischer & Bartal) ويتبعها (Albers & List).

وناقش قصي السلامي [2] في 2012 كيفية سلوك الخوارزميات الخمسة السابقة مستخدماً نموذجين من نماذج نظرية صفوف الانتظار في بحوث العمليات لتحديد الخوارزمية المثلى، فعمل أولاً على محاكاة نموذج $(M/M/C/GD/\infty/\infty)$ ، وثانياً نموذج $(M/M/C):(NPRP/\infty/\infty)$ ذات نمط خدمة لعدم أحقية أولوية تقديم الخدمة ومقارنتها مع النتائج المأخوذة من محاكاة نموذج المعالجات متعددة النواة، وقد توصل إلى أن أفضل قيم للمعاملات قيد الدراسة لـ (Fleischer) قد كانت عند 10 و 15 دورة فلم ينافسها غير (List) بفارق بسيط، أما في حالة 5 دورات فلقد تميزت (Bartal) وأقتربت منها (Albers) بفارق الكسور العشرية ومنافسهما الشديد هي (List).

خوارزمية KA المقترحة

لقد جاءت فكرة استحداث خوارزمية KA المقترحة (1) من خلال مراقبة الدراسة التطبيقية لسلوك الخوارزميات الأربع (List, Bartal, Albers, Fleischer) من محاكاة النموذج الأول $(M/M/C/GD/\infty/\infty)$ باستخدام 9 معاملات، ومحاكاة

¹ جاء الأسم المختصر للخوارزمية المقترحة (KA) من الحرف الأول لأسم الباحث (KUSAY) والحرف الأول من أسم زميله (AHMAD).

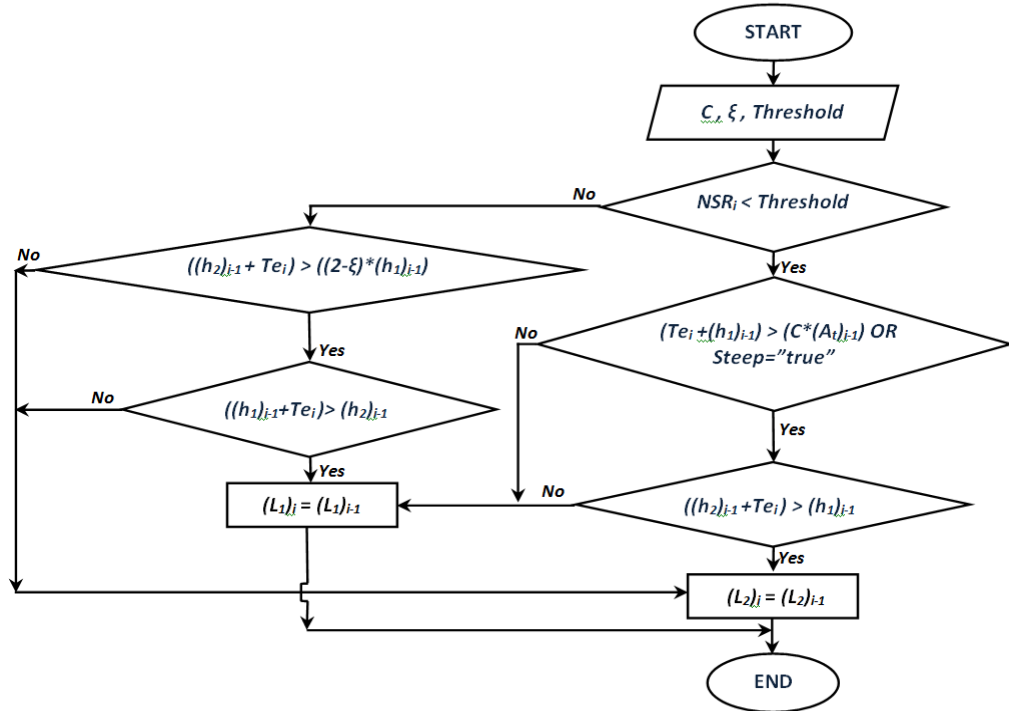
النموذج الثاني $(NPRP/\infty/\infty): (M_i/M/C)$ باستخدام 15 مُعامل، تم ملاحظة كفاءة عمل خوارزمية (Fleischer) في المرتبة الأولى ويتبعها خوارزمية (Bartal)، وأن هناك انقلاب يحدث في قيمة معامل عدم اتزان الاحمال ((Non-Stable Rate(NSR) لتحديد المعالج الأخف حملاً، وذلك لتكليفه بتنفيذ المهمة القادمة، وهذا الانقلاب يحدث للخوارزميتين بالشكل التالي: (2 pp.118)

لخوارزمية (Fleischer) يحدث عند $\text{Min.}(0.123) \leq \text{NSR} \leq \text{Max.}(0.913)$

لخوارزمية (Bartal) يحدث عند $\text{Min.}(0.5349) \leq \text{NSR} \leq \text{Max.}(0.977)$

جدول 1: يوضح معنى كل رمز مستخدم في خوارزمية KA

الرمز	المعنى	الرمز	المعنى
NSR_i	معامل عدم اتزان الاحمال	$(L_1)_i$	قائمة بأسماء المعالجات الثقيلة
Te_i	وقت الخدمة المطلوب على المعالج	$(L_2)_i$	قائمة بأسماء المعالجات الخفيفة
$(h_1)_i$	ارتفاع صف انتظار القائمة الثقيلة للمعالج الأول	$(A_i)_i$	معدل ارتفاع صف الانتظار لجميع المعالجات
$(h_2)_i$	ارتفاع صف انتظار القائمة الخفيفة للمعالج الثاني	$Steep_i$	متغير منطقي قيمته true أو false
$C=1.8$	ثابت	$Threshold$	متغير تتغير قيمته وفق نسق معين
$\xi=0.5$	ثابت يمثل قيمة Epsilon		



شكل 1: يوضح المخطط الانسيابي لخوارزمية KA

مستخدم في الخوارزميتين كما في جدول 1، أما شكل رقم (1) فإنه يوضح المخطط الانسيابي للخوارزمية المقترحة:

If $(NSR_i < Threshold)$ then

If $\{ (Te_i + (h_1)_{i-1}) > (C * (A_i)_{i-1}) \}$ OR $\{ Steep = "true" \}$ then

If $\{ (h_2)_{i-1} + Te_i > (h_1)_{i-1} \}$ then $(h_2)_{i-1}$ Else $(L_1)_{i-1}$

Elseif $\{ ((h_2)_{i-1} + Te_i) > ((2-\xi) * (h_1)_{i-1}) \}$ then

If $\{ (h_1)_{i-1} + Te_i > (h_2)_{i-1} \}$ then $(L_1)_{i-1}$ Else $(L_2)_{i-1}$

If $\{ (h_1)_{i-1} + Te_i \} > (h_2)_{i-1} \}$ then $(L_1)_{i-1}$ Else $(L_2)_{i-1}$

Else $(L_2)_{i-1}$

لهذا شغلت خوارزمية (Fleischer) عند الزمن $t=1$ وتستمر بالعمل إلى أن يحدث هناك فرق كبير بين توزيع الأحمال على المعالجين (حسب مبدأ عمل هذه الخوارزمية)، ولكن لا يُسمح للوصول إلى الحد الأدنى لها وهو (Min. 0.123) بل إلى القيمة $(NSR \geq 0.3)$ ، لتشتغل بدورها (Bartal) وتستمر الخوارزمية الأخيرة بالعمل إلى أن يقل الفرق بين توزيع الأحمال على المعالجين (حسب مبدأ هذه الخوارزمية)، ولكن لا يُسمح للوصول إلى الحد الأعلى لها وهو (Max. 0.977) بل إلى القيمة $(NSR \leq 0.6)$ ، ليرجع الأمر إلى سابقة وتعمل (Fleischer) على تكبير الفارق بين توزيع الأحمال على المعالجين، عندها يتم استخدام (Bartal) لتقليل الفارق وهكذا تستمر هذه الخوارزمية المقترحة بالعمل على هذا المبدأ. سيتم دراسة سلوك خوارزمية KA ومعرفة كفاءتها في توزيع المهام على المعالجين أسوة ببقية الخوارزميات الأربع، والوقوف على نقاط القوة والضعف فيها، وذلك باستخدام التحليلات الإحصائية بالاعتماد على 15 معامل من المعاملات قيد الدراسة والتي استخدمت لتقييم الخوارزميات الأربعة السابقة وهي :

$\rho_1, \rho_2, \rho, Wq_1 * \mu, Wq_2 * \mu, Wq * \mu, Lq_1, Lq_2, Lq, Ws_1 * \mu, Ws_2 * \mu, Ws * \mu, Ls_1, Ls_2, Ls$.

حيث تم أخذ أنواع مختلفة لعدد دورات المعالج $(No. of cycles)$ ، تعتمد في توليدها على القيم الافتراضية لتجربة المحاكاة باستخدام ثلاث قيم مختلفة لعدد الدورات هي $(5, 10 \text{ and } 15 \text{ cycles})$ ، تم تطبيقها على خوارزمية KA.

وتم افتراض وصول (164) مهمة مطلوب جدولتها على المعالجات المتاحة⁽²⁾. تحتاج هذه المهام إلى وقت تنفيذ بمعدل (2000) دورة معالج خلال (645) وحدة زمنية (هذه القيمة متغيرة وتعتمد على مقدار عدد الدورات هل هي 5 أو 10 أو 15)، وللحصول على النتائج المطلوبة فقد تم استخدام برنامج أكسل (Excel 2010) لتقديم المخططات الإحصائية اللازمة لتحليل نتائج عملية المحاكاة التي تم القيام بها، حيث تم توليد القيم العشوائية للأوقات اللازمة لعملية محاكاة عدد الدورات التي تحتاجها كل مهمة لغرض إنجازها على

المعالج (Time in service)، وهي قيم محصورة ما بين $(1 \leq \text{time in service} \leq 30)$ ^[4]، وتم الاعتماد على الدالة Randbetween (bottom, top) في برنامج أكسل لإرجاع رقم عشوائي صحيح محصور بين 1 و30، ويتم إرجاع رقم عشوائي صحيح جديد في كل مرة يتم فيها حساب أي خلية من

خلابا برنامج أكسل، علماً أن هذا البرنامج يعتمد ضمناً على العديد من الدوال الاحتمالية لتوليد الأرقام العشوائية، وما يخصنا الدالة Uniform (low, high) المحصورة بين المدى [0,1] حسب الصيغة التالية :

$Uniform (low, high) = low + (high - low) * RAND()$

حيث تم توليد أربع أشكال مختلفة لقيم عدد الدورات التي قد يصادفها المعالج (تمثل أسلوب الضغط على المعالج) وهي (عشوائية، تصاعدية، تنازلية، متقلبة)، ومن ثم إدخال المهام الواسلة كلاً حسب عدد الدورات التي تحتاجها وحسب نوع الخوارزمية إلى برنامج أكسل ليتم بالنتيجة جدول هذه المهام على أحد المعالجات.

لقد تم تطبيق (12) حالة دراسية على نموذج العشوائية $(M_i/M/C):(NPRP/\infty/\infty)$ ، بسبب استخدام 4 توزيعات للقيم العشوائية (عشوائي، تصاعدي، تنازلي، متقلب High-Low) للفترة التي تحتاجها المهام القادمة للمعالجة داخل المعالجات، وكل توزيع قيم قد تعاملت مع 3 قيم مختلفة لعدد الدورات التي يحتاجها المعالج لمعالجة المهام القادمة، وهي (5، 10 و 15) دورة/وحدة زمنية، معتمدين على 15 معامل تخص نموذج صف الانتظار لتقييم سلوك وكفاءة خوارزمية KA.

إن نموذج صف الانتظار المشار إليه أعلاه يفترض أن جميع الزبائن (المهام) يمتلكون نفس توزيع وقت الخدمة بغض النظر من يملك الأحقية المطلقة، وأن كل قنوات الخدمة تملك نفس التوزيع الأسّي لوقت الخدمة بمعدل خدمة μ ، وأن الوصول إلى صف الانتظار K ذو الأولوية الأكبر يكون بمعدل وصول λ_k ويتبع توزيع بواسون لكل قيم $(k=1,2,...,m)$ ، لهذا يُمكن الحصول على مقاييس الفعالية التالية لصف الانتظار K بالشكل التالي.^[6]

$$W_q^{(k)} = \frac{L}{(1 - S_k)}$$

where $S_0 \equiv 0$ a

وللحصول على حالة الاستقرار يفضل أن تكون $\rho_i = \frac{\lambda_i}{c\mu} < 1$ كذلك فإن $E\{\xi_0\} = \frac{1}{c\mu(\rho^{-c}(c-\rho)(c-1)! \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + 1)}$ وأن وقت الانتظار المتوقع لأي زبون في صف الانتظار وبقية المقاييس هي:

$$W_q = \sum_{k=1}^m \frac{\lambda_k}{\lambda}$$

$$L_q = \lambda \cdot W_q$$

$$L_q^{(k)} = \lambda_k \cdot W_q^{(k)}$$

$$W_s^{(k)} = W_q^{(k)} +$$

²ملاحظة: تم دراسة سلوك النظام بافتراض وصول 50 و100 و150 مهمة، إلى أن استقر سلوك النظام عند 164 مهمة، حيث تم وصول 164 مهمة خلال (645cycles) فيحالة (15cycles)، وتسلكها لتوزيعات العشوائية نفسها لسلوكياتها في الدورات (5, 10cycles/times) وتختلف عنها فقط بوصول 164 مهمة خلال (215cycles) فيحالة (5cycles)، ووصول 164 مهمة خلال (430cycles) فيحالة (10cycles).

W_s

معاملات كثافة الحركة (p_1, p_2, p) المعدلة:

نلاحظ من الأشكال ذات الأرقام (2(A,B,C)) أن هناك علاقة عكسية بين عدد دورات المعالج وأياً من قيم p_1 أو p_2 ، وهذا يعني أنه يُفضل أن تزداد عدد الدورات الخاصة بتنفيذ كل عملية، فيزيادها يقل الضغط على المعالج ويستطيع انجاز عمليات أكثر ويقل معدل كثافة الحركة فيه، فالفرق الكبير بين قيم p_1 و p_2 نجده متزامن مع عدد الدورات 10 و 15 كما في الأشكال رقم (2(A,B))، ويمكن تجميع الأشكال السابقة بشكل (3) الذي يوضح جميع قيم p لجميع أنواع توزيع القيم العشوائية Te لجميع عدد الدورات لخوارزمية KA، حيث يُمكن تمييز العلاقة بين عدد الدورات وقيمة معدلات كثافة الحركة بكل سهولة.

القيم المتوقعة لجميع أنواع أوقات الانتظار $Wq(t)$

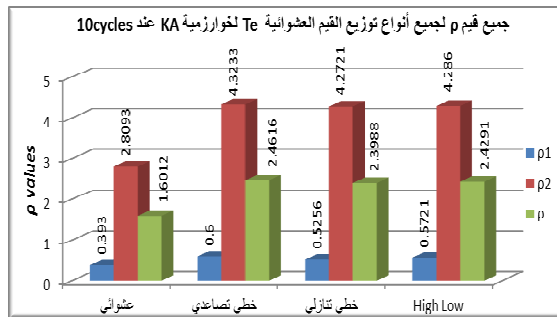
فيما يتعلق بالزمن الذي تنتظره الدورة الواحدة من التعليم لكل ثانية (cycles/time) في المعالج، نلاحظ أن KA تسلك سلوكاً عكسياً واضح بعلاقة عدد الدورات وقيمة وقت الانتظار فكلما ازداد عدد الدورات قلت قيمة وقت الانتظار بغض النظر عن نوع التوزيع (عشوائي، تنازلي، تصاعدي، متقلب)، كما في شكل (4).

القيمة المتوقعة لعدد المهام التي يقدم لها الخدمة Lq

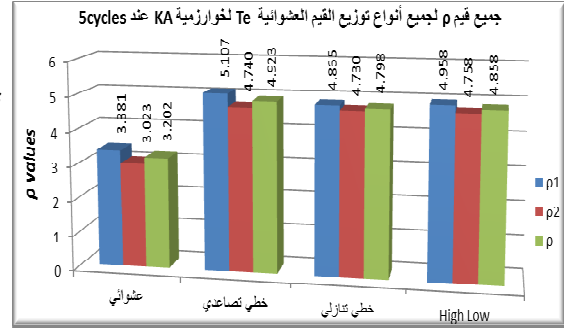
فيما يتعلق بطول صف الانتظار (Lq) للمهام التي تنتظر تلقي الخدمة أمام المعالج، نلاحظ أن القيمة المتوقعة لعدد المهام لخوارزمية KA تسلك سلوكاً طردياً مع ازدياد عدد الدورات، بغض النظر عن نوع توزيع القيم العشوائي، فكلما ازداد عدد الدورات ازداد عدد المهام المنتظرة تلقي الخدمة أمام المعالج، كما في شكل (5).

القيمة المتوقعة لوقت انتظار مهمة ما في النظام Ws :

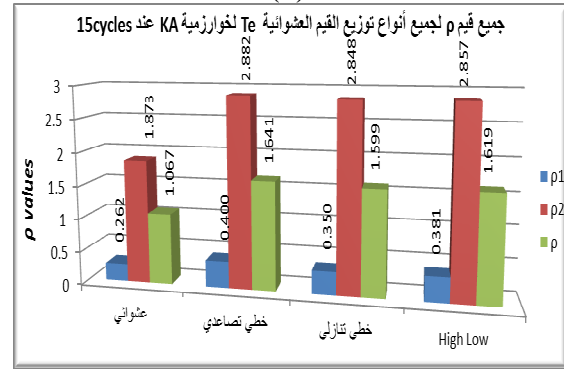
أن القيمة المتوقعة لزمن انتظار مهمة ما في النظام لـ KA قد حافظت على مستوى متقارب، وهو أدنى مستوى مقارنة ببقية الخوارزميات بغض النظر عن عدد الدورات ونوع التوزيع العشوائية، إلا حالة واحدة شاذة للتوزيع التنازلي لعدد الدورات 15 قد ارتفعت فيها قيمة وقت الانتظار إلى حد عالٍ، كما في شكل (6).



(A)



(B)



(C)

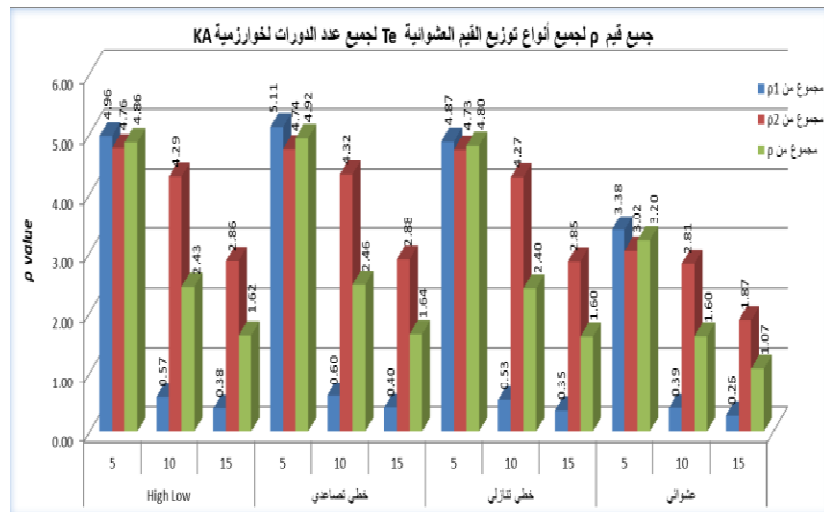
شكل 2: يوضح سلوك معامل كثافة الحركة لخوارزمية 10 cycles (B) 5 cycles (A) KA 15 cycles (C)

القيمة المتوقعة لطول صف الانتظار داخل النظام Ls

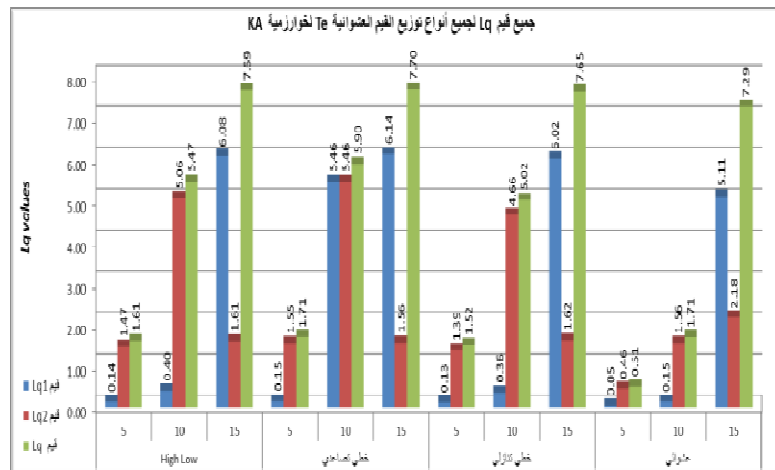
بالنسبة لطول صف الانتظار داخل النظام نلاحظ أن خوارزمية KA تسلك سلوكاً عكسياً مع ازدياد عدد الدورات، بغض النظر عن نوع توزيع القيم العشوائي فكلما ازداد عدد الدورات قل عدد المهام المنتظرة تلقي الخدمة أمام المعالج، كما في شكل (7).

تحليل نتائج بيانات الانحراف المعياري لجميع الخوارزميات

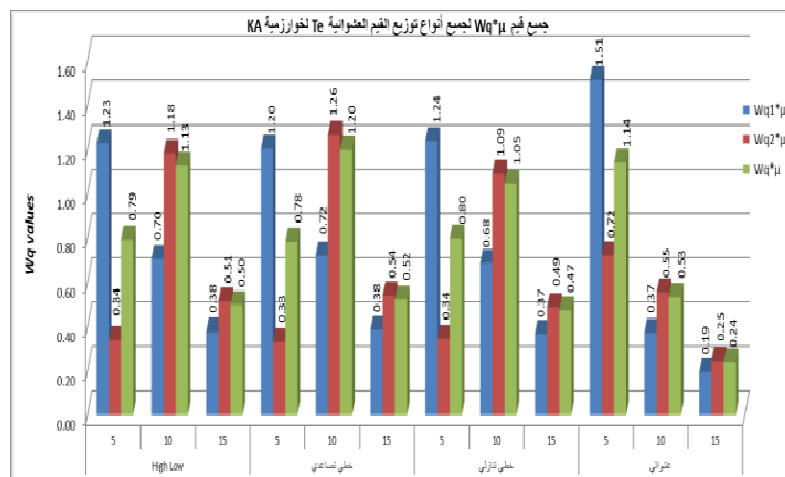
أن سلوك المعامل (Threshold) وتقلب قيمه متشابه في سلوكه (إلى حد بعيد)، وهو متعلق بكل نوع توزيعه سواء كانت (عشوائي، تصاعدي، تنازلي، متقلب) بغض النظر عن عدد الدورات، والذي يُمكن توضيحه في جدول رقم (2). لقد تم أخذ جميع الحالات الدراسية لسلوك المعامل (Threshold) لخوارزمية KA ومقارنتها مع الخوارزميات الأربعة (List, Bartal, Albers, Fleischer)، حيث أن قيمة هذا المعامل تتغير حسب نوع توزيع القيم العشوائية وكما موضح في جدول رقم (2)، ونلاحظ أن لدينا (8 و 6 و 5 و 4) حالات للتوزيع (العشوائية، المتقلبة، التصاعدية، التنازلية) على التوالي، أي تم دراسة 23 حالة دراسية لخوارزمية KA ومقارنتها مع الخوارزميات الأربعة، مع مراعاة أن كل خوارزمية لها قيمها الخاصة حسب (عدد دورات المعالج ونوع توزيع القيم العشوائية)، بهذا يُمكن تحليل نتائج بيانات الانحراف المعياري لجميع الخوارزميات للمعاملات التالية :



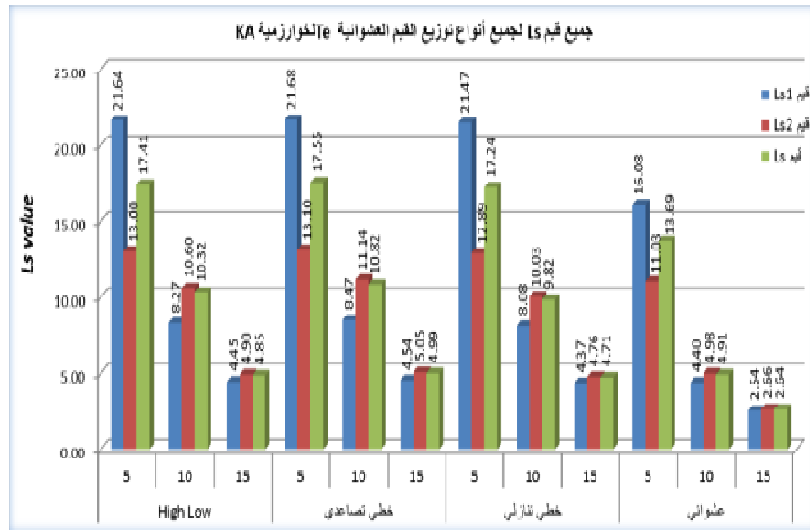
شكل 3: يوضح جميع قيم p لجميع أنواع توزيع القيم العشوائية Te لجميع عدد الدورات لخوارزمية KA



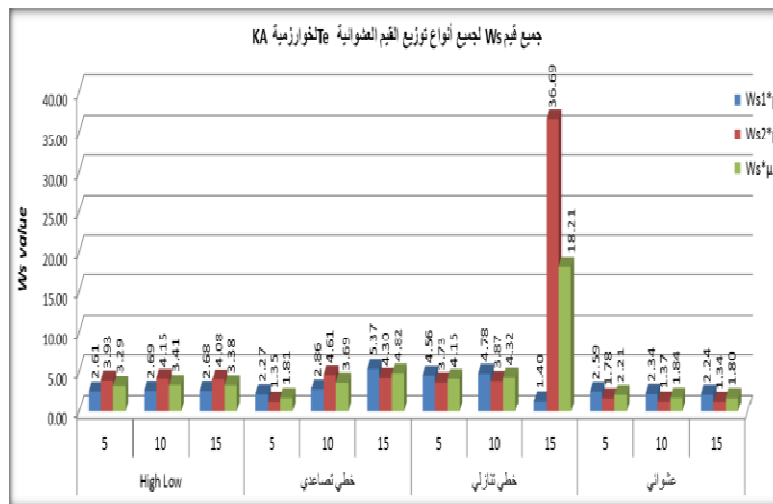
شكل 4: يوضح جميع قيم Wq لجميع أنواع توزيع القيم العشوائية Te لجميع عدد الدورات لخوارزمية KA



شكل 5: يوضح جميع قيم Lq لجميع أنواع توزيع القيم العشوائية Te لجميع عدد الدورات لخوارزمية KA



شكل 6: يوضح جميع قيم Ws لجميع أنواع توزيع القيم العشوائية Te لجميع عدد الدورات لخوارزمية KA



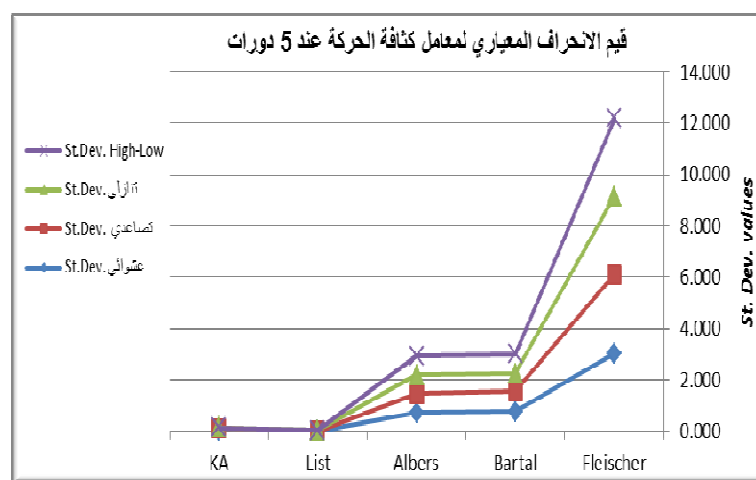
شكل رقم (7) يوضح جميع قيم Ls لجميع أنواع توزيع القيم العشوائية Te لجميع عدد الدورات لخوارزمية KA

1. كثافة الحركة (ρ_1, ρ_2, ρ)
 لقد أثبتت خوارزمية (List) بامتلاكها أقل انحراف معياري لمعاملات كثافة الحركة مساوية للصفر أو قريبة منه لجميع حالات عدد الدورات (5, 10 & 15) ولمختلف توزيع القيم العشوائية، وهذا يعني أن هنالك ($3 \times 4 = 12$) حالة مقارنة بين أنواع الخوارزميات الأربعة مقابل أفضل القيم لخوارزمية KA، وأن KA قد تطابقت أو كانت قريبة جداً من خوارزمية (List) في 8 حالات من أصل 12 حالة، ولم يصل إليهما أي من خوارزميات (Bartal, Albers & Fleischer)، ويمكن استنتاج الأشكال رقم (8(A,B,C)) الآتية.

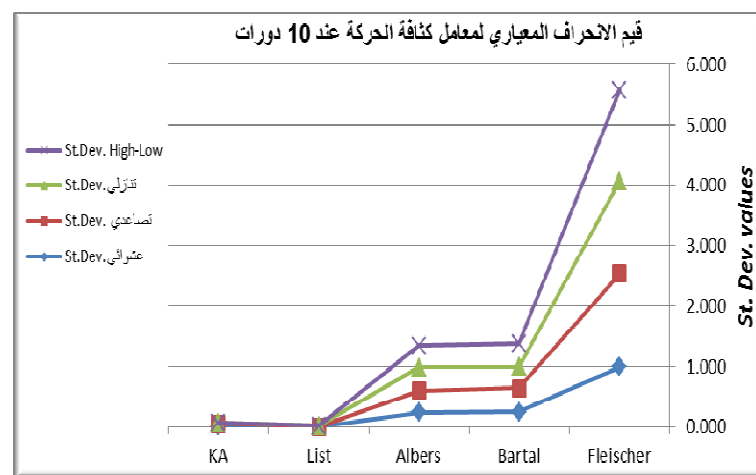
2. وقت الانتظار في صف الانتظار (Wq_1, Wq_2, Wq)
 لقد أثبتت خوارزمية KA أنها الأفضل بالنسبة لقيمة الانحراف المعياري لمعاملات وقت الانتظار من بين جميع الخوارزميات، فهي الأفضل على الإطلاق في 8 حالات دراسية من أصل 12، كان منها 5 حالات اجتازت بها جميع و3 حالات تطابق مع خوارزمية (Fleischer) عند توزيع القيم التنازلية لعدد الدورات 10 و 15، و 4 حالات اقتربت بشكل كبير وبفارق الكسور البسيطة عن خوارزمية (Bartal) التي أعطت أفضل النتائج عند الـ 5 دورات، وتقايرت منها خوارزمية (Albers)، ويمكن استنتاج الأشكال رقم (9(A,B,C)).

جدول 2: يوضح كيفية تقلب قيم المعامل (Threshold) لجميع القيم العشوائية

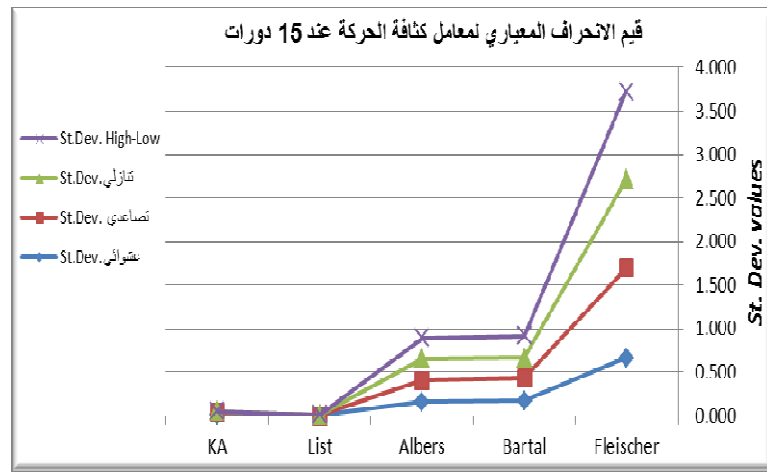
عشوائي	متقلب	تصاعدي	تنازلي
Thresh. ≤ 0	Thresh. ≤ 0	Thresh. ≤ 0	Thresh. ≤ 0
$0 < \text{Thresh.} \leq 0.2$	$0 < \text{Thresh.} \leq 0.05$	$0 < \text{Thresh.} \leq 0.4615$	$0 < \text{Thresh.} \leq 0.513$
$0.2 < \text{Thresh.} \leq 0.479$	$0.05 < \text{Thresh.} \leq 0.5$	$0.4615 < \text{Thresh.} \leq 0.7$	$0.513 < \text{Thresh.} \leq 0.8695$
$0.479 < \text{Thresh.} \leq 0.58$	$0.5 < \text{Thresh.} \leq 0.7$	$0.7 < \text{Thresh.} \leq 0.84$	$0.8695 < \text{Thresh.}$
$0.58 < \text{Thresh.} \leq 0.651$	$0.7 < \text{Thresh.} \leq 0.913$	$0.84 < \text{Thresh.}$	
$0.651 < \text{Thresh.} \leq 0.768$	$0.913 < \text{Thresh.}$		
$0.768 < \text{Thresh.} \leq 0.875$			
$0.875 < \text{Thresh.}$			



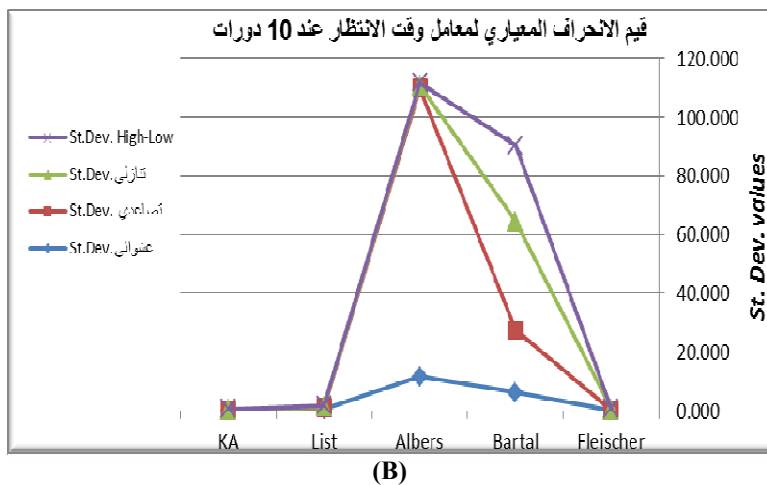
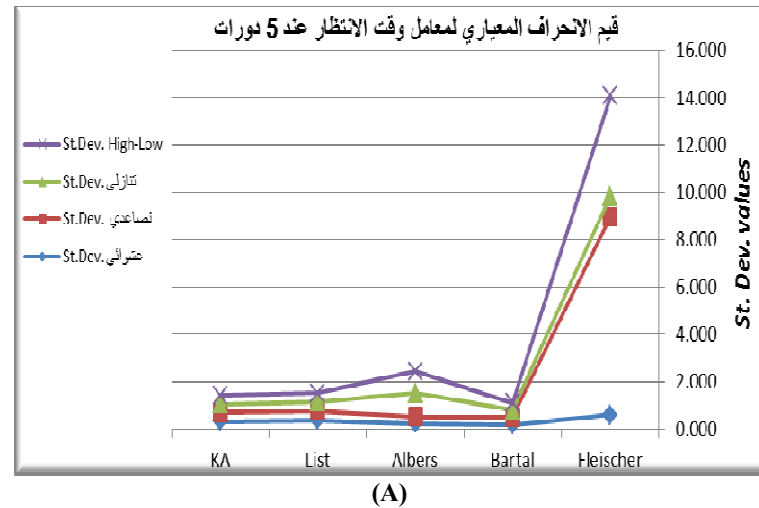
(A)

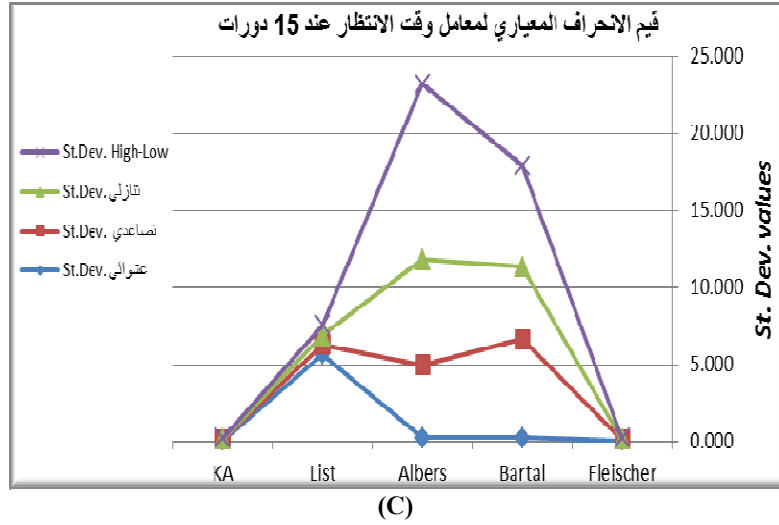


(B)



شكل 8: يوضح قيم الانحراف لمعاملات كثافة الحركة عند (A) 5 دورات (B) 10 دورات (B) 15 دورة





شكل 9: يوضح قيم الانحراف لمعاملات وقت الانتظار عند (A) 5 دورات (B) 10 دورات (C) 15 دورة

3. عدد المهام عند صف الانتظار (Lq_1, Lq_2, Lq) أثبتت خوارزمية KA كفاءتها بإعطائها أقل عدد مهام تنتظر في صفوف الانتظار على الإطلاق مقابل جميع الخوارزميات، حيث أعطت أقل قيمة للانحراف المعياري وقدرة (2.093, 0.641, 0.192) في حالة (5 و 10 و 15) دورة معالج على التوالي، وجميعها في حالة توزيع القيم العشوائي، وتطابقت خوارزمية (Fleischer) مع KA في حالتين من أصل 12 عند عدد الدورات (10 و 15) وكلاهما في حالة التوزيع التنازلي، ويُمكن استنتاج الأشكال رقم (10(A,B,C)).

4. وقت انتظار المهام داخل النظام (Ws_1, Ws_2, Ws) تقدمت خوارزمية KA على الجميع في حالة عدد الدورات 10 و 15 دورة معالج، فمن أصل 12 حالة دراسية تقدمت KA بشكل مطلق بـ 5 حالات، وتطابقت مع (Fleischer) في 3 حالات وتقدمت (Bartal) ويتبعها بشكل متقارب (Albers) في 4 حالات، وكانت KA مقاربه لهم بفارق الكسور العشرية البسيطة، ويُمكن استنتاج الأشكال رقم (11(A,B,C)).

5. (Ws_1, Ws_2, Ws) طول صف الانتظار داخل النظام امتازت خوارزمية KA بامتلاكها أقل قيمة للانحراف المعياري لطول صف الانتظار داخل النظام على الإطلاق على بقية الخوارزميات عند الـ 10 و 15 دورة معالج، في 6 حالات من أصل 12 وتطابقت مع (Fleischer) في حالتين، أما خوارزمية (Bartal) ويتبعها (Albers) فتقدمت في 4 حالات عند الـ 5 دورات للمعالج واقتربت KA منهما بفارق الكسور العشرية البسيطة، ويُمكن استنتاج الأشكال رقم (12(A,B,C)).

تحليل سلوك المعاملات قيد الدراسة

بالنظر إلى جميع حالات عدد الدورات نجد أن أفضل قيم للمعاملات قيد الدراسة لخوارزمية KA قد كانت عند 5 دورات فلم ينافسها غير خوارزمية (List) بفارق بسيط، أما في حالة

10 و 15 دورة فلقد اقتربت (Fleischer) من KA بفارق الكسور العشرية ولكن لم تجتازها في أي نوع من توزيعات القيم العشوائية (عشوائي، تصاعدي، تنازلي، متقلب).

من خلال الرسوم الإحصائية لسلوك قيم المعاملات قيد الدراسة المبينة في الأشكال رقم (13(A,B,C)) والتي تم اختيارها بشكل عشوائي من بين 23 حالة دراسية لمعامل (Threshold) لخوارزمية KA التي تم الإشارة إليها سابقاً، نجد أن سلوك هذه المعاملات ثابت ومتقارب إلى حد كبير ومرتبطة مع عدد الدورات بغض النظر عن نوع توزيع القيم العشوائية التي لم تؤثر على سلوك هذه القيم، حيث نلاحظ أن الأشكال السابقة متشابهة من حيث طبيعة سلوكها، وفي حالة متابعة كل معامل على حدا لتبين أنه يسلك سلوكاً متشابهاً تقريباً.

مثال ذلك لعدد الدورات 10 و 15 دورة ولمختلف توزيع القيم العشوائية نلاحظ الأشكال رقم (13(B,C)) لتبين أنها متشابهة رغم اختلاف توزيع القيم العشوائية.

الاستنتاجات CONCLUSIONS

لقد تم تصميم وتنفيذ خوارزمية KA الخلطة من خوارزميتي (Fleischer) و (Bartal)، وهي أكثر كفاءة منهما وذلك بسبب الدور الفعال لأساليب بحوث العمليات وما قدمته نظرية صفوف الانتظار من تحسين لعمل خوارزميات جدولة المعالج. بالنظر إلى جميع حالات عدد الدورات نجد أن أفضل قيم للمعاملات قيد الدراسة لـ KA قد كانت عند 5 دورات فلم ينافسها غير خوارزمية (List) بفارق بسيط، أما في حالة 10 و 15 دورة فلقد اقتربت خوارزمية (Fleischer) من KA بفارق الكسور العشرية ولكن لم تجتازها في أي نوع من توزيعات القيم العشوائية.

شكل رقم (14) يلخص عدد الحالات التي تميزت بها كل خوارزمية لتحليل نتائج الانحراف المعياري لـ 12 حالة دراسية تم تطبيقها على أفضل نتائج لخوارزمية KA، في حين يلخص

جدول رقم (3) العلاقة بين المعاملات قيد الدراسة مع عدد الدورات لجميع الخوارزميات لنموذج صف الانتظار وما يقابلها من الخوارزمية المقترحة.

التوصيات RECOMMENDATIONS

استخدام خوارزمية KA أسوة ببقية خوارزميات الجدولة لحل مشكلة جدولة الوصول المباشر للمهام، وذلك باعتماد بيانات حقيقية لوصول المهام مأخوذة من شركات عملاقة كشركة (Intel) و (AMD) لتقييم سلوك خوارزمية KA، واعتمادها كطريقة جدولة فعالة.

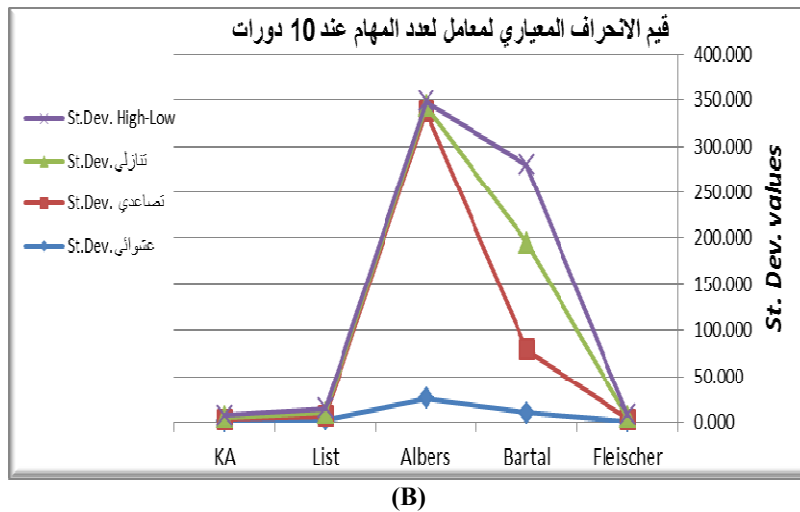
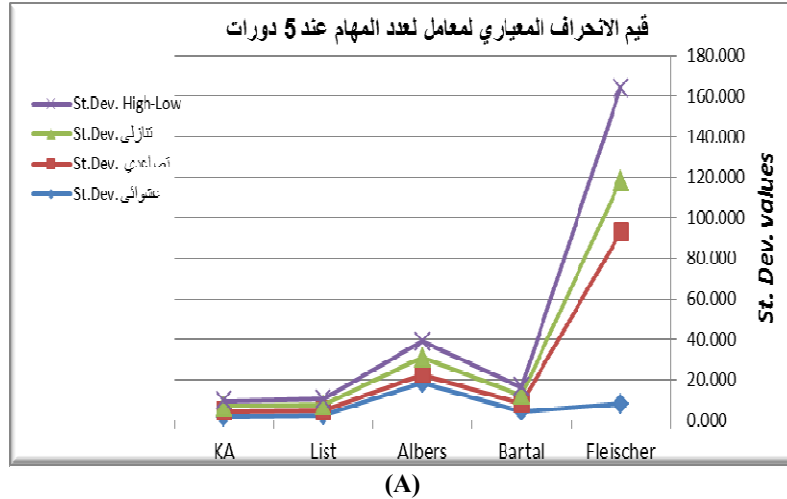
بسبب التحسين الذي قدمته بحوث العمليات لخوارزميات الجدولة، نوصي باستخدامها خاصة في الحالات التي تحاول إيجاد الأسلوب الأمثل لمواردها.

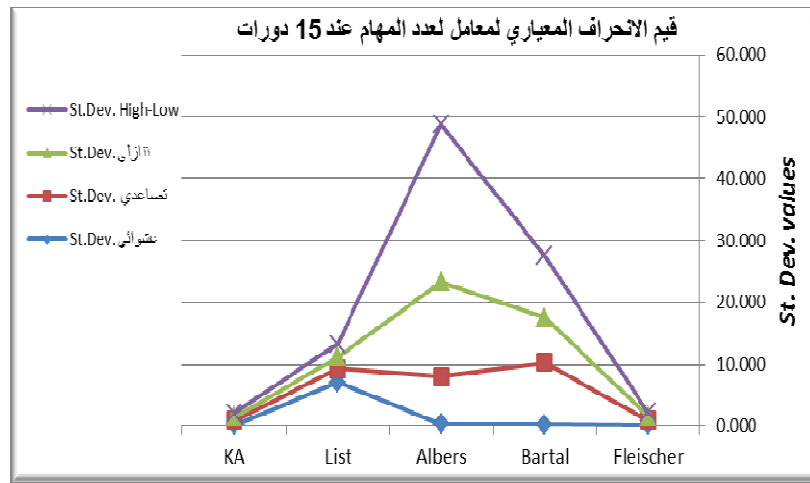
الدراسات المستقبلية FUTURE STUDIES

- حيث أن خوارزمية KA قد أظهرت أفضل النتائج نتيجة دمج خوارزميتين فقط، لذا يقترح الباحث عمل خوارزمية

مزيج من جميع الخوارزميات الأربعة، أو كل اثنان أو ثلاث معاً وذلك بعد دراسة سلوك المعاملات قيد الدراسة، ومن ثم تطوير شرط عمل الخوارزمية الجديدة.

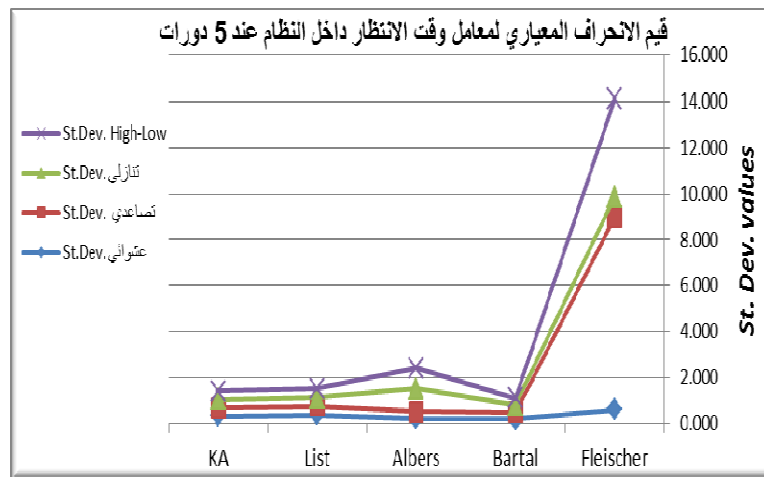
- استخدام أسلوب آخر من أساليب بحوث العمليات لتطوير مجال عمل الحاسوب والمجالات الأخرى لما قدمته من تحسين وتطوير لخوارزميات جدولة المعالج، كاستخدام البرمجة الديناميكية (Dynamic Programming) لتقسيم البرامج المعقدة (التي يتم تنفيذها على شبكات الحاسوب العملاقة) إلى مجموعة برامج صغيرة تنفذ كل منها على جانب، ومن ثم يتم تجميع الحلول الفرعية لتكون الحل الأمثل للمشكلة الأصلية.
- أعداد دراسة بحثية لمحاكاة عمل نماذج شبكات صفوف الانتظار وخاصة ما يتعلق بالحاسوب مثل شبكة الطوابير المفتوحة والمختلطة ونموذج الخادم المركزي، كمحاولة لتحسين أسلوب عملها معتمدين على علم بحوث العمليات.



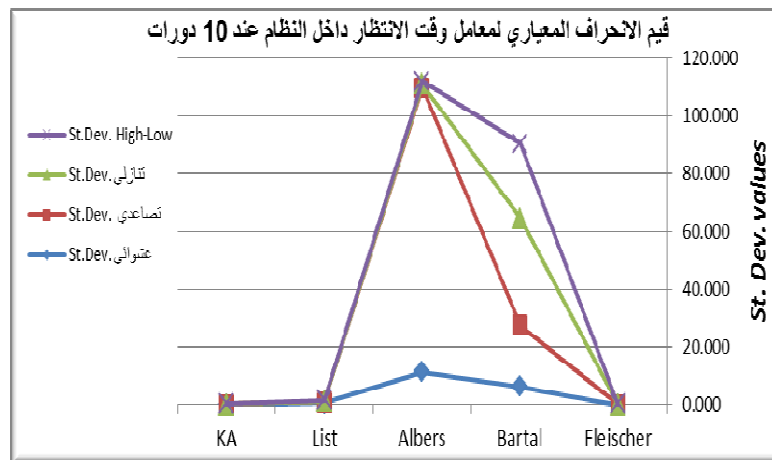


(C)

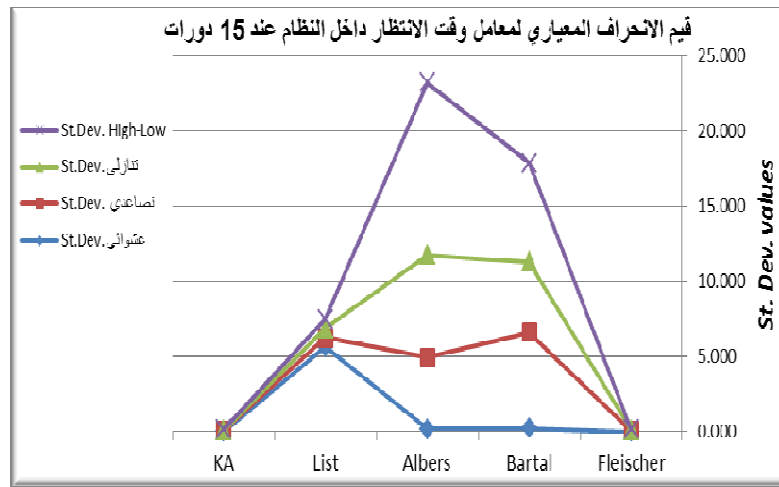
شكل 10: يوضح قيم الانحراف لمعاملات عدد المهام عند (A) 5 دورات (B) 10 دورات (C) 15 دورة



(A)

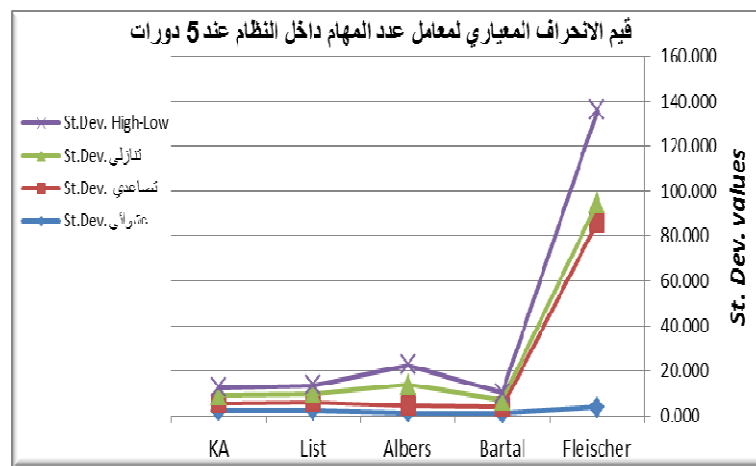


(B)

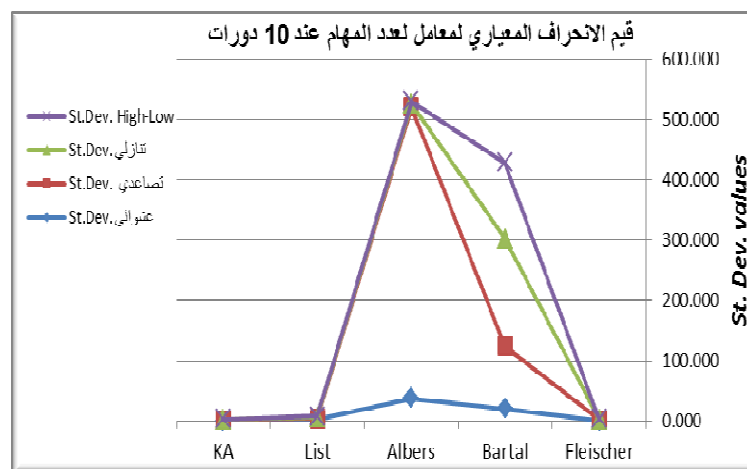


(C)

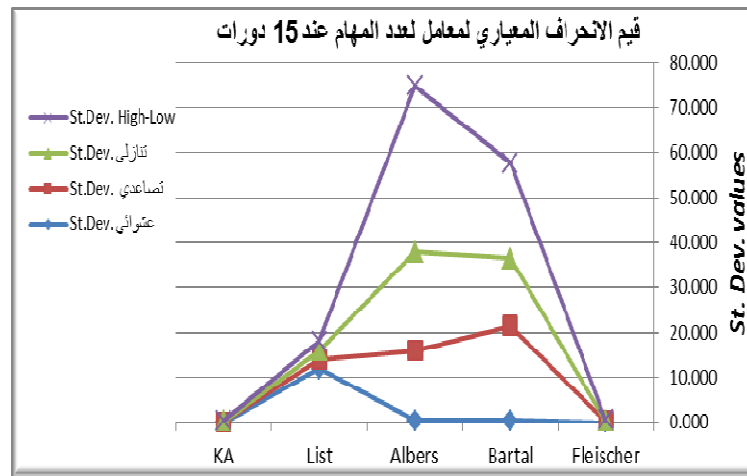
شكل 11: يوضح قيم الانحراف المعياري لمعاملات وقت الانتظار عند 5 دورات (A) دورات 10 دورات (B) دورات 15 دورة



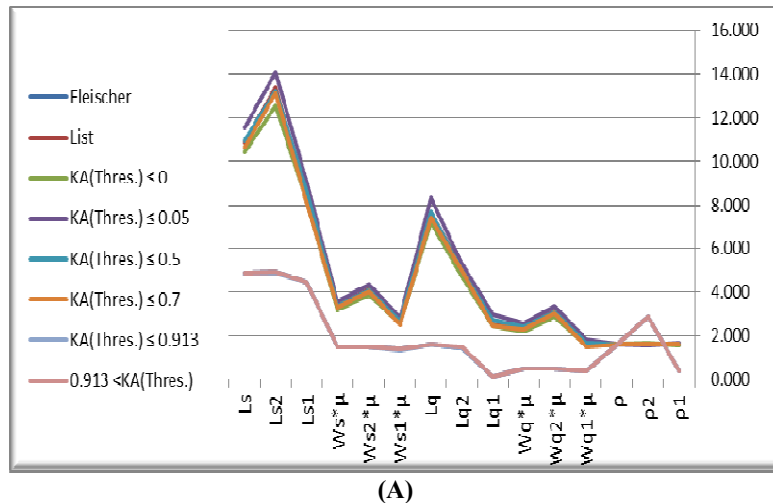
(A)



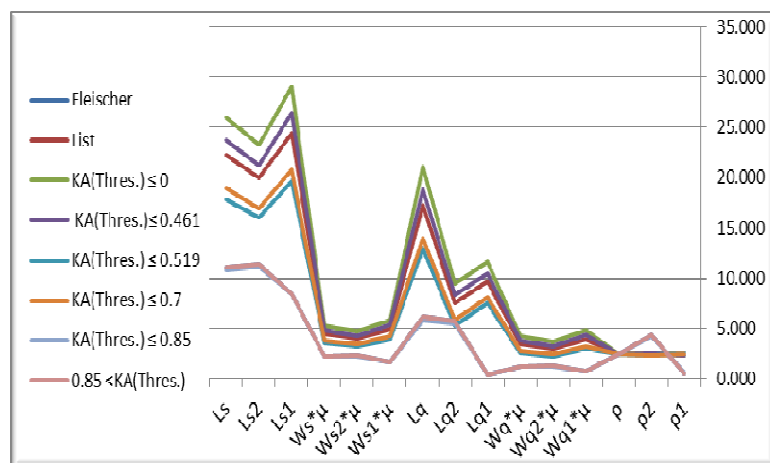
(B)



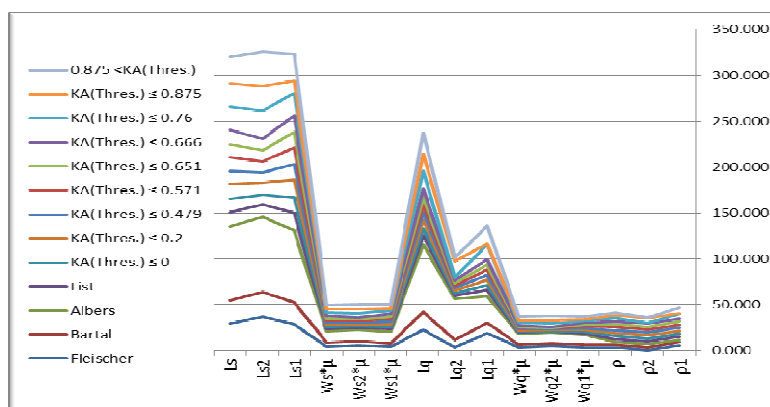
شكل 12: يوضح قيم الانحراف المعياري لمعاملات عدد المهام داخل (A) النظام عند 5 دورات (B) النظام عند 10 دورات (C) النظام عند 15 دورة



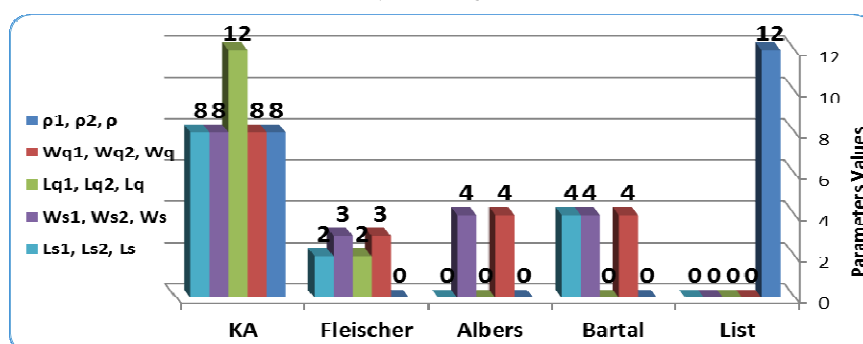
(A)



(B)



شكل 13: سلوك معاملات قيد الدراسة لجميع الخوارزميات مقابل جميع حالات KA عند (A) دورات 5 (B) دورات 10 (B) دورات 15 للتوزيع المتقلبة مع حذف القيم الشاذة (C)



شكل 14: يوضح عدد الحالات التي تميزت بها كل خوارزمية لتحليل نتائج بيانات الانحراف المعياري

جدول 3: يوضح العلاقة بين المعاملات قيد الدراسة مع عدد الدورات لجميع الخوارزميات

المعامل	Graham's List	Bartal	Albers	Fleischer	KA
p1 p2 p	ذات علاقة عكسية مع عدد الدورات	ذات علاقة عكسية مع عدد الدورات	ذات علاقة عكسية مع عدد الدورات	ذات علاقة عكسية مع عدد الدورات	ذات علاقة عكسية مع عدد الدورات
Wq1 Wq2 Wq	مقارنة لـ Bartal طردية معتدل طردية معتدل	أعلى قيمة شاذة أعلى قيمة شاذة أعلى قيمة شاذة	علاقة عكسية مقارنة لـ Bartal مقارنة لـ Bartal	علاقة عكسية عكسي معتدل عكسي معتدل	علاقة عكسية واضحة تمتاز عن الجميع
Lq1 Lq2 Lq	أعلى قيمة شاذة أعلى قيمة شاذة علاقة متقلبة	أعلى قيمة شاذة أعلى قيمة شاذة أعلى قيمة شاذة	علاقة عكسية علاقة طردية أعلى قيمة شاذة	علاقة عكسية علاقة عكسية عكسي حاد	علاقة طردية واضحة تمتاز عن الجميع
Ws1 Ws2 Ws	أعلى قيمة شاذة علاقة طردية طردية معتدل	أعلى قيمة شاذة أعلى قيمة شاذة أعلى قيمة شاذة	علاقة عكسية أعلى قيمة شاذة أعلى قيمة شاذة	علاقة عكسية علاقة عكسية علاقة عكسية	ذات سلوك مستقر وهو الأقل من بين جميع الخوارزميات
Ls1 Ls2 Ls	أعلى قيمة شاذة أعلى قيمة شاذة علاقة متقلبة	أعلى قيمة شاذة أعلى قيمة شاذة أعلى قيمة شاذة	عكسي حاد مقارنة لـ Bartal مقارنة لـ Fleischer	عكسي حاد عكسي حاد عكسي حاد	علاقة عكسية واضحة تمتاز عن الجميع

المصادر REFERENCES

- [1] زياد القاضي، (2004) المفاهيم الأساسية في أنظمة التشغيل، الطبعة الأولى، مكتبة المجتمع العربي للنشر والتوزيع، صفحة 29.
- [2] قصي حميد السلامي، (2012) استخدام أساليب بحوث العمليات لتحسين كفاءة جدولة صفوف الانتظار لزيادة سرعة معالجات الحاسوب، أطروحة دكتوراه، جامعة أم درمان الإسلامية، كلية الاقتصاد والعلوم السياسية، قسم الإحصاء.
- [3] قصي حميد السلامي، أحمد محمد رؤوف، دراسة تطبيقية لخوارزميات جدولة المهام المباشرة لمتعدد المعالجات باستخدام تقنيات بحوث العمليات، مؤتمر الدراسات العليا السنوي الرابع، جامعة النيلين، كلية الدراسات العليا، 2012.
- [4] محمد إبراهيم العدوي، المعالجات الدقيقة Microprocessors (2000) Intel80186 – Intel8086 – Z80 – Intel8085 –) Pentium – Intel80486 – Intel80386 – Intel80286)، الدار الدولية للاستثمارات الثقافية، مصر، الطبعة الأولى.
- [5] Albert Y.Z., Matthew C., and Stephan O.Iariu, (1998) A Framework for Reinforcement-Based Scheduling in Parallel Processor Systems, IEEE Transactions on Parallel and distributed systems, Vol.9, No.3, pp.249.
- [6] Hamdy A. Taha, Operations Research: An Introduction, by Pearson Education, Inc., Eighth Edition, pp.545, 2007.
- [7] Raihan Al-Ekram, (2004) Online Multiprocessor Scheduling. <http://arnetminer.org/dev.do?m=downloadpdf&url=http://arnetminer.org/pdf/PDFFiles2>.
- [8] Peter Brucker, (2007) Scheduling Algorithms, Fifth Edition, Springer Berlin Heidelberg New York, pp.317.
- [9] Susanne Albers, Bianca Schröder, (2002), An Experimental Study of Online Scheduling Algorithms, ACM Journal of Experimental Algorithmics, vol(7), pages 154-160.
- [10] Zexuan Zhu, Y. S. Ong and M. Dash, (2007), Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework, IEEE Transactions on Systems, Man and Cybernetics - Part B 37 (1), pp.70-76.