

العنوان:	دراسة تطبيقية لخوارزميات جدولة المهام المباشرة لمتعدد المعالجات باستخدام تقنيات بحوث العمليات
المصدر:	مجلة الدراسات العليا
الناشر:	جامعة النيلين - كلية الدراسات العليا
المؤلف الرئيسي:	السلامي، قصي حميد
مؤلفين آخرين:	روؤف، أحمد محمد(م. مشارك)
المجلد/العدد:	مج2
محكمة:	نعم
التاريخ الميلادي:	2013
الصفحات:	278 - 254
رقم MD:	608625
نوع المحتوى:	بحوث ومقالات
اللغة:	Arabic
قواعد المعلومات:	IslamicInfo, HumanIndex, EcoLink, EduSearch
مواضيع:	وحدة المعالجة المركزية ، هندسة البرمجيات
رابط:	http://search.mandumah.com/Record/608625

دراسة تطبيقية لخوارزميات جدولة المهام المباشرة لمتعدد

المعالجات باستخدام تقنيات بحوث العمليات

An Applied Study of Online Multiprocessor Scheduling Algorithms Using of Operations Research Techniques

قصي حميد السلامي¹ ، أحمد محمد رؤوف²

1. كلية بغداد للعلوم الاقتصادية الجامعة-قسم هندسة البرمجيات - جمهورية العراق-

Email: qalsalami72@Yahoo.com

2. جامعة بنغازي-كلية تقنية المعلومات-قسم شبكات واتصالات الحاسوب-الجمهورية العربية الليبية

مستخلص الورقة

تعتبر وحدة المعالجة المركزية الوحدة المسؤولة عن تنفيذ كافة العمليات الحسابية والمنطقية (تنفيذ البرامج)، ويفترض في هذه الوحدة أن تكون عاملة طوال الوقت (ما أمكن)، وعادة ما تقاس كفاءة استخدام وحدة المعالجة CPU Utilization كنسبة للمهام التي تقع في هذه الوحدة والتي هي في حالة تنفيذ (5) حيث تكون خوارزميات جدولة المهام مسؤولة عن توزيعها على المعالجات.

على هذا الأساس تم محاكاة عمل عدة خوارزميات جدولة متعدد المعالجات لدراسة سلوك عمل المعالج في حالة استقباله عدداً من الأعمال (Jobs) وعلى أربع فترات عشوائية مختلفة (عشوائي، تصاعدي، تنازلي، متقلب)، ومن ثم تم تحليل 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 نظرية صفوف الانتظار في بحوث العمليات.

Abstract:

Central Processing Unit (CPU) is the most important part of computer systems, which is responsible of executing all arithmetic and logical operations - in another words program execution - and for that reason it is supposed to be active for most of its operating cycle. The CPU utilization is a measurement of the ratio between jobs that are in execution, and the total jobs CPU is handling. The scheduling algorithms are responsible of distributing such jobs on the processors of the CPU□□assure an efficient CPU utilization. Based on that, a simulation of a number of multi-processors scheduling algorithms has been made in order to study CPU's behavior when receiving a number of jobs on four random-job-duration patterns (completely random, ascending random, descending random & inconstant random), then each case was analyzed using charts, formed form each case's data using Excel 2010, which showed the convergence and divergence between those algorithms. Finally a comparison was done between those algorithms using parameters from queuing theory in operations research.

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

المقدمة

تناقش هذه الورقة البحثية عدداً من الخوارزميات المستخدمة لحل مشكلة (جدولة مهام متعدد المعالجات)، والتي يمكن وصفها باستخدام المعلومات التالية: -

- ❖ خصائص المهام: هناك عدد n من المهام، لكل منها موعد بدء وزمن معالجة محددين.
- ❖ خصائص المعالجات: هناك m من المعالجات التي تعمل بشكل متوازي (Parallel).
- ❖ قيود المعالجة: لا يوجد هناك إعادة ترتيب للمهام ويتم التعامل معها بشكل مباشر (فوري).
- ❖ الدالة المستهدفة: زمن العمل (Make-span) وهو الوقت المطلوب لإنجاز المهمة بشكل كامل.

مشكلة البحث: كيفية تحديد الخوارزمية المثلى لجدولة متعدد المعالجات التي تقلل وقت التنفيذ الكلي للمهام الفرعية التي تتنافس لإنجازها.

هدف البحث: هو إجراء محاكاة لعمل متعدد المعالجات، وذلك بدراسة التقنيات والخوارزميات المتبعة حالياً في جدولة صفوف انتظار المهام أمام المعالجات، ومن بينها (3,6): -

- Graham's List Scheduling خوارزمية جراهام لجدولة القوائم
- Bartal's Algorithm خوارزمية بارتال
- Karger's Algorithm خوارزمية كارجر
- Albers's Algorithm خوارزمية ألبيرس
- Fleischer's Algorithm خوارزمية فليستشر

ولتقييم سلوك هذه الخوارزميات استخدمت نظرية صفوف الانتظار في بحوث العمليات لتحديد الخوارزمية المثلى.

جدولة المهام

جدولة المهام (Tasks Scheduling) 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، خصائص المعالجات، القيود على عمليات المعالجة، ومعيار الأمثلة المطلوبة.

تتضمن خصائص المهام وقت المعالجة اللازم لتنفيذها، موعد البدء لكل مهمة، زمن التنفيذ الأقصى وأهمية التنفيذ. أما خصائص المعالجات فتتلخص في عدد المعالجات، سرعة المعالجة وإعدادات عمل هذه المعالجات (التي قد تكون مثلاً متوازية أو تسلسلية). هناك العديد من القيود التي تفرض على تنفيذ عمليات المعالجة مثل أسبقيات التنفيذ، غرامات التأخير، إمكانية تغيير ترتيب التنفيذ، وطريقة التعامل مع المهام عند وصولها (التعامل الفوري (المباشر) والتعامل غير المباشر).

جدولة المهام المباشرة

في هذا النوع من خوارزميات جدولة المهام، لا يكون لدى مجدول المهام (Scheduler) المعلومات الكاملة عن كامل دفعة المهام المدخلة عند بدء التنفيذ، بدلاً من ذلك فإنه يحصل على معلومات جزئية عند دخول كل مهمة على حدة (أي أنه يستقي المعلومات عند إدخال المهام واحدة بعد أخرى)، وعليه أن يتصرف بناء على هذه المعلومات الجزئية للقيام بجدولة هذه المهام، أيضاً يكون لكل مهمة في الجدولة المباشرة موعد بدء يتم عنده بالضبط إرسال المهمة إلى المجدول.

هناك تصنيفان للجدولة المباشرة: الجدولة المباشرة الاستقرائية (Clairvoyant) التي يعطى فيها المجدول زمن المعالجة للمهمة حالما يتم البدء في تنفيذها، والجدولة المباشرة غير الاستقرائية (Non-clairvoyant) والتي لا يكون فيها لدى المجدول أي فكرة عن وقت المعالجة اللازمة لأي مهمة.

إن عدم وجود معلومات حول دفعة المهام المدخلة في الجدولة المباشرة تجعل من المستحيل على المجدول أن يضمن نفس الوضع المثالي لجميع حالات الإدخال، لذلك فمعظم الأبحاث في هذا المجال تعمل على إيجاد خوارزميات جدولة تضمن الوصول إلى أقرب نتيجة مثالية يمكن تحقيقها باستخدام خوارزميات الجدولة غير المباشرة، وإن أداء خوارزمية الجدولة المباشرة يقاس من خلال ما يعرف بالتحليل التنافسي (Competitive Analysis)، والذي هو عبارة عن طريقة لحساب النسبة بين الدالة التي تستهدفها خوارزمية الجدولة المباشرة إلى تلك الخاصة بالخوارزمية الأمثل.

خوارزميات الجدولة

قائمة جراهام (Graham's List Scheduling)

ظهرت هذه الخوارزمية في أواسط الستينيات، وتقوم على إسناد أي مهمة جديدة إلى المعالج الأقل حملاً على افتراض أن المهام تعطى في قائمة متسلسلة، وهو ما يتفق مع طريقة وصول المهام في الجدولة المباشرة، وتفترض أن النسبة التنافسية لهذه الخوارزمية تقدر بـ $(2-1/m)$ لكل قيم m ، ولحساب هذه النسبة يتم افتراض أن زمن المعالجة للمهمة i هو p_i ، وأن

$$W = \sum_{i=1}^n p_i$$

هو مجموع زمن المعالجة لجميع المهام، وإذا كان زمن العمل للمجدول المثالي هو (Opt) فإن زمن المعالجة الكلي المتوفر للمجدول المثالي هو $m \cdot \text{Opt}$ من ذلك نجد أن: -

وتحت افتراض أن A هو زمن العمل لخوارزمية مجدول القائمة، فسيكون لكل مهمة K لها زمن معالجة P_K بحيث ينتهي عند نهاية زمن العمل، بهذا لن يكون هناك أي معالج شاغر قبل $(A-P_K)$ ، وذلك لأنه إذا تحقق ذلك سيجب إسناد المهمة K إلى ذلك المعالج، لذلك فإن جميع المعالجات ستكون مشغولة من الزمن. إلى $(A-P_K)$. من ذلك يمكن إيجاد النسبة التنافسية كالتالي: -

$$\therefore W \geq m (A - p_k) + p_k$$

$$\therefore W \geq m A - p_k (m - 1)$$

$$\Rightarrow A \leq \frac{W}{m} + p_k \frac{m-1}{m}$$

$$A \leq O p t + O p t \frac{m-1}{m}$$

$$\therefore A = (2 - \frac{1}{m}) O p t \Rightarrow \frac{A}{O p t} = (2 - \frac{1}{m})$$

خوارزمية بارتال (Bartal's Algorithm)

وضع بارتال خوارزمية جديدة أعطت نسبة تنافسية قدرها 1,986 تقريبا لأي $(m \geq 70)$ (3)، حيث جاءت كحل لمشكلة رئيسية كانت تواجه خوارزمية جدولة القائمة بشكل مستمر، فإذا كان هناك $(1-m)$ ، مهمة موزعة على m من المعالجات (أي أن كل معالج لديه $m-1$ من المهام بحيث أن كل مهمة حجمها 1، وجاءت مهمة جديدة حجمها $(m-1)$)، فستقوم خوارزمية القائمة بإسناد المهمة الجديدة الكبيرة إلى أحد المعالجات المشغولة أصلا (بمعنى آخر سيكون لأحد المعالجات زمن عمل قدره $(2(m-1))$)، إن هذه المشكلة تحدث لأن خوارزمية القائمة تعمل دائما على (موازنة) الأحمال على جميع المعالجات، وعندما تأتي مهمة كبيرة فإنها ستبعد زمن العمل كثيرا عن القيمة المثالية.

طريقة العمل تتلخص في أنه إذا جاءت مهمة صغيرة فسيتم أولاً فحص نسبة المعالجات الخفيفة، إذا كانت النسبة بعد إسناد المهمة الجديدة لأي منها لا تقل عن 44,5% فإنها (المهمة الجديدة) ستسند لأحد المعالجات الخفيفة، وإلا فإنها ستسند مباشرة إلى أحد المعالجات المشغولة (لأن الهدف هو الحفاظ على نسبة المعالجات الخفيفة)، وعندما تأتي مهمة كبيرة فسيتم إسنادها مباشرة إلى أحد المعالجات الخفيفة، على الرغم من أن ذلك يخرق القاعدة ولكن إسنادها إلى أحد المعالجات المشغولة سيؤدي إلى زيادة كبيرة في زمن العمل، مما لا يترك أمام الخوارزمية المجال إلى إسنادها إلى أحد المعالجات الخفيفة.

إن خوارزمية بارتال مصممة لعدد m من المعالجات أكبر من الحد المعين (m_0) ، ولو تم افتراض $\epsilon = \frac{1}{m_0}$

سنجد أن النسبة التنافسية ستكون مساوية لـ $(2-\epsilon)$ ، وأيضا يمكننا تعريف $\epsilon \in (0,1)$ بحيث أن $m \geq \frac{1}{\epsilon}$ ستكون المتممة لـ m .

نفترض أن الارتفاع h لأي معالج هو مجموع زمن المعالجة للمهام المسندة إلى 5، وعليه يجب أن تتوفر للخوارزمية قائمة بجميع المعالجات مرتبة ترتيبا تصاعديا حسب h ، ويجب أن تحدث هذه القائمة بشكل مستمر، فإذا تم توزيع عدد t من المهام، فسيكون لدينا قائمتان: قائمة المعالجات الخفيفة L_t التي تحوي m من المعالجات مع ارتفاعاتها، وقائمة المعالجات المشغولة H_t التي تحتوي على $(m-\epsilon m)$ مع ارتفاعاتها، وبافتراض أن $M(x)$ هي قيمة متوسط الارتفاعات للقائمة x و $A(x)$ هي قيمة أقل ارتفاع ضمن القائمة x ، فيمكن كتابة الخوارزمية كالتالي:-

عند وصول المهمة $t+1$ ذات زمن المعالجة p_{t+1} نفذ:-

• إذا كان $A(L_t) + p_{t+1} \leq (2-\epsilon) \cdot M(H_t)$ أسند المهمة الجديدة إلى أول معالج في H_t .

• وإلا أسند المهمة إلى أول معالج في L_t .

لحساب النسبة التنافسية، افترض بارتال أن $(m_0 = 70)$ وأن جميع قيم $(0.445 + \epsilon) \in (0.445 - 0.5, 0.5)$ ، وأثبتت أن النسبة التنافسية للخوارزمية هي $\cong (2 - \epsilon) \cdot 1.986$.

خوارزمية كارجر (Kargel's Algorithm)

طور كارجر خوارزمية تتفوق على خوارزمية القائمة خصوصا عندما تكون $(m \geq 6)$ ، حيث تحصل على نسبة تنافسية قدرها 1,945 لجميع قيم m . وتعتبر خوارزمية كارجر في الواقع صيغة أكثر شمولية من صيغة بارتال، حيث أنه بدلا من تقسيم المعالجات في قائمتين (الخفيفة والمشغولة)، ومن ثم الاختيار بين أول معالج في أي من القائمتين، تعمل خوارزمية كارجر على اختيار معالج معين من ضمن قائمة موحدة وفقا لشروط معينة تحقق عدم موازنة الأحمال على المعالجات.

باعتبارها النسخة الأعم من خوارزمية بارتال، سيتم استخدام مصطلح الارتفاع h مرة أخرى، حيث ستحتفظ الخوارزمية دائما بقائمة محدثة للمعالجات مترتبة ترتيبا تصاعديا وفقا لارتفاعها، لنفترض أن الزمن t هو الزمن قبل جدول المهمة t^{th} مباشرة، على ذلك يكون M_i^t هو المعالج $(i+1)^{\text{st}}$ الأخف عند الزمن t (بمعنى أن M_0^t هو المعالج الأخف على الإطلاق عند الزمن t ، يليه $M_1^t \dots$ إلخ)، وسيكون بالطبع h_i هو ارتفاع هذا المعالج عند نفس الزمن (لكل القيم السابقة $(0 \leq i \leq m)$). بافتراض A_i^t هي متوسط الارتفاع للمعالجات الأقصر حتى الرقم i (هنا $m \leq i \leq m$ لأن $A_0^t = \infty$)، فسيكون A_m^t هو متوسط الارتفاع لجميع المعالجات المتوفرة عند الزمن t وبالتالي الحد الأدنى لـ Opt .

أما خطوات الخوارزمية فهي عند وصول المهمة t ذات زمن المعالجة p_t تسند المهمة الجديدة إلى المعالج الأكثر انشغالا M_k^t الذي يحقق الشرط $A_k^t \leq a \cdot h_k + p_t$.

يحدد المعامل a درجة عدم الاتزان في الأحمال بين المعالجات، ونلاحظ أن خوارزمية كارجر تستخدم متوسط الارتفاع للمعالجات الأخف قبل إضافة المهمة الجديد وليس بعدها. بالنسبة للقيمة التنافسية، فقد أثبت كارجر أنه لكل $(6 \leq m \leq 13000)$ ستكون هناك قيمة لـ a بحيث تكون $(2 - \frac{1}{m}) < \alpha < 1.943$ ، هذه القيمة ستجعل خوارزميته تنافسية بمقدار a على m من المعالجات.

خوارزمية ألبريس (Albers's Algorithm)

إن طريقة عمل هذه الخوارزمية أبسط من الخوارزميتين السابقتين، كما أنها تحسن النسبة التنافسية لتصبح 1,923 لأي $(m \geq 2)$. تقوم فكرة الخوارزمية على أنها (في أي لحظة) تحافظ على تحديث قائمتين: قائمة L_1 المكونة من $(m/2)$ للمعالجات الخفيفة، وقائمة L_2 المكونة من $(m/2)$ للمعالجات المشغولة، وسيتم توجيه المهمة الجديدة إلى أقل المعالجات انشغالا أما في L_1 أو في L_2 بناء على النسبة بين مجموعي الارتفاعات في القائمتين.

تقوم الخوارزمية بالاحتفاظ بقائمة بجميع المعالجات مرتبة ترتيباً تصاعدياً بناء على الارتفاع في ذلك الوقت. لنفرض أن الزمن T هو الزمن بعد جدول المهمة t^{th} مباشرة (عكس الخوارزمية السابقة)، سيكون M_k^t هو المعالج الـ K^{th} الأخف عند الزمن t ، أيضا نفرض أن h_k^t هو ارتفاع المعالج M_k^t $(1 \leq k \leq m)$ ، في أي زمن ستكون h_k^t هي زمن العمل الحالي.

لتكن H_1 هي مجموع الارتفاعات للمعالجات في القائمة L_1 و H_h هي مجموع الارتفاعات للمعالجات في القائمة L_h ، هكذا سيكون المجموع الكلي للارتفاعات $(H = H_1 + H_h)$ ، وتكن $C = 1.923$ ، و $i = (m/2)$ و $j = 0.29 m$ كذلك:-

$$\alpha = \frac{(c - 1)i - \frac{j}{2}}{(c - 1)(m - i)} \approx 0.686$$

ستكون خطوات الخوارزمية كالتالي:-

عند وصول المهمة t ذات زمن المعالجة p_t نفذ الآتي:-

$$\text{if } (H_1 \leq \alpha H_h) \text{ or } (h_m > h_m^{t-1}) \text{ and } (h_m > \frac{cH}{m})$$

أسند المهمة الجديدة إلى المعالج الأخف، وإلا أسند المهمة إلى المعالج $(i+1)$ الأخف.

خوارزمية فليستشر (Fleischer's Algorithm)

تتميز هذه الخوارزمية بتنافسية أكبر من الخوارزميات السابقة عندما يكون عدد المعالجات $(m \geq 64)$ ، حيث

تقترب النسبة التنافسية من

$$1 + \sqrt{\frac{1 + \ln 2}{2}} = 1.9201$$

كلما اقتربت $(m \rightarrow \infty)$ ، هذه تعد أفضل نتيجة ممكنة بالنسبة لخوارزميات جدولة مهام متعددة المعالجات المحددة

المباشرة حتى الآن.

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

التصاعدي للخوارزميات السابقة، فإذا كان t هو الزمن بعد جدولة المهمة t ، وكان M_k هو المعالج k^{th} الأكثر انشغالاً

عند الزمن t وارتفاعه h_k ، فسيكون M_1 و M_m هما المعالجان الأكثر انشغالاً والأقل انشغالاً على الترتيب عند

الزمن t ، وسيكون h_k هو زمن العمل عند نفس الزمن.

ليكن A_j هو متوسط الارتفاع للمعالجات (M_1, \dots, M_m) وأن $A = A_1$ (متوسط الحمل على جميع

المعالجات عند الزمن t)، أما خطوات الخوارزمية فستكون كالتالي:-

عند وصول المهمة t ذات زمن المعالجة p_t نفذ الآتي:-

• إذا كان الجدول في الوضع المنحدر (Steep) أو $A > c \cdot p_t + h_i^{t-1}$ أسند المهمة الجديدة إلى المعالج الأخف t

M_m .

• وإلا أسند المهمة إلى المعالج (i) الأكثر انشغالاً M_i^{t-1} .

مقارنة بين خوارزميات جدولة متعدد المعالجات

5,1 من منظور الدراسات السابقة

نلاحظ أن جميع الخوارزميات التي تقدم نسبة تنافسية أفضل من خوارزمية القائمة التقليدية، تعتمد على فكرة التوزيع غير المتساوي للمهام (الأحمال) على المعالجات، حيث يتم جدولة المهمة الجديدة إما إلى أقل معالج انشغالاً طالما أن ارتفاعه قصير كفاية أو إلى أحد المعالجات المشغولة طالما أن ذلك يحقق شرطاً ما (عدم اتزان الأحمال).
وكما في جدول رقم (1) فإنه يلخص أهم النقاط المتعلقة بالخوارزميات السابقة، علماً بأن C هي النسبة التنافسية المطلوبة، في حين يعرض جدول رقم (2) مقارنة لنتائج هذه الخوارزميات. (3)

جدول رقم (1) يوضح مقارنة بين خوارزميات جدولة متعدد المعالجات

الخوارزمية	اختيار المعالج	المحافظة على ارتفاع المعالج	قائمة المعالجات الخفيفة
بارتال	تختار المعالج الأخف من قائمة المعالجات الخفيفة أو من قائمة المعالجات الطويلة	تحافظ على ارتفاع المعالج صاحب المهمة الجديدة ليكون أقل من C مضروباً في متوسط ارتفاعات قائمة المعالجات الخفيفة قبل إسنادها له	قائمة المعالجات الخفيفة تحتوي على أول 44,5% من المعالجات الخفيفة، الباقي في قائمة المعالجات المشغولة
كارجر	تختار المعالج الأكثر انشغالاً من قائمة المعالجات الخفيفة	تحافظ على ارتفاع المعالج صاحب المهمة الجديدة ليكون أقل من C مضروباً في متوسط ارتفاعات قائمة المعالجات الخفيفة قبل إسنادها له	المعالجات الخفيفة يتم تحديدها ديناميكياً عند اختيار المعالج الذي سيتم إسناد المهمة إلى ه
ألبيرس	تختار المعالج الأخف من قائمة المعالجات الخفيفة أو من قائمة المعالجات الطويلة	تحافظ على الارتفاع الكلي لقائمة المعالجات الخفيفة ليكون أقل من C مضروباً في الارتفاع الكلي لقائمة المعالجات الطويلة بعد إسناد المهمة الجديدة للمعالج	قائمة المعالجات الخفيفة تحتوي على أول 50% من المعالجات الخفيفة، الباقي في قائمة المعالجات المشغولة
فليستشر	تختار المعالج الأخف من قائمة المعالجات الخفيفة أو من قائمة المعالجات الطويلة	تحافظ على ارتفاع المعالج صاحب المهمة الجديدة ليكون أقل من C مضروباً في متوسط ارتفاعات جميع المعالجات قبل إسنادها له	المعالجات الخفيفة يتم تحديدها بناء على قيمة تتحد من خلال m و C .

جدول رقم (2) يوضح مقارنة نتائج خوارزميات الجدولة

النسبة التنافسية	m	الخوارزمية
$c = 2 - \frac{1}{m}$	كل قيم m	جراهام
1,9860	$70 < m$	بارتال
1,9450	كل قيم m	كارجر
1,9230	$2 \leq m$	ألبيرس
1,9201	$\infty \leftarrow m$	فليستشر

لقد عمل Susanne A. (1) على إجراء تجارب عملية لاختبار كفاءة أربع خوارزميات جدولة هي (جراهام، بارتال، ألبيرس، كارجر) 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، وهل هي أقل أو قريبة أو أكثر من مقدار معدل التحميل لأن ذلك يؤثر بشكل مباشر 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.

من منظور بحوث العمليات

يمكن الاعتماد على نتائج محاكاة نموذج نظرية صفوف الانتظار في بحوث العمليات (96 PP.7) ذات أولوية تقديم الخدمة (NPRP/∞/∞): (M_i/M/C) مع النتائج المأخوذة من محاكاة نموذج المعالجات متعددة النواة، 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 and 15 cycles)، تم تطبيقها على أربع خوارزميات جدولة (Albers, Bartal, Fleischer, Graham' s list) وتم افتراض وصول (164) مهمة مطلوب جدولتها على المعالجات المتاحة (14). 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)، تعمل خلالها كل خوارزمية على اتباع الأسلوب الخاص بها في جدولة هذه المهام.

ولإتمام كل ما تقدم وللحصول على النتائج المطلوبة فقد تم استخدام برنامج (Microsoft Excel 2010) لتقديم المخططات الإحصائية اللازمة لتحليل نتائج عملية المحاكاة التي تم القيام بها، حيث تم توليد القيم العشوائية

(14) ملاحظة: تم دراسة سلوك النظام بافتراض وصول 50 و 100 و 150 مهمة، إلى أن استقر سلوك النظام عند 164 مهمة، الأشكال (A, B, C, D) توضح وقت خدمة العملية والتي على أساسها تم تثبيت عدد المهام الواصلة إلى النظام بالعدد 164 مهمة. تتضمن هذه الأشكال وصول 164 مهمة خلال (215cycles) في حالة (5cycles) ووصول 164 مهمة خلال (430 cycles) في حالة (10cycles).

للأوقات اللازمة لعملية محاكاة عدد الدورات التي تحتاجها كل مهمة لغرض إنجازها على المعالج (Time in service)، وهي قيم محصورة ما بين $(1 \leq \text{time in service} \leq 30)$ (2 pp.165)، وتم الاعتماد على الدالة $\text{Randbetween (bottom, top)}$ 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 خلايا برنامج أكسل، علماً أن هذا البرنامج يعتمد ضمناً على العديد من الدوال الاحتمالية لتوليد الأرقام العشوائية، وما يخصنا هو الدالة $\text{Uniform (low, high)}$ المحصورة بين المدى $[0,1]$ حسب الصيغة التالية:

$$\text{Uniform (low, high)} = \text{low} + (\text{high} - \text{low}) * \text{RAND}$$

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 تم إدخال المهام الواصلة كلاً على حدى وحسب عدد الدورات التي تحتاجها وحسب نوع الخوارزمية إلى برنامج أكسل ليتم بالنتيجة جدولة هذه المهام على أحد المعالجات. إن نموذج صف الانتظار المشار إليه سابقاً يفترض أن جميع الزبائن يمتلكون نفس توزيع وقت الخدمة بغض النظر عن 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 $(1, 2, \dots, m)$ ، $(k=)$ لهذا يمكن الحصول على مقاييس الفعالية التالية لصف الانتظار K بالشكل:-

$$W_q^{(k)} = \frac{E\{\xi_0\}}{(1 - S_{k-1})(1 - S_k)}, k = 1, 2, \dots, m \text{ where } S_0 \equiv 0 \text{ and } S_k = \sum_{i=1}^k \frac{\lambda_i}{c\mu} < 1, \text{ for all } k$$

$$E\{\xi_0\} = \frac{1}{c\mu(\rho^{-c}(c-\rho)(c-1)! \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + 1)}$$

وللحصول على حالة الاستقرار يفضل أن تكون $\rho_i = \frac{\lambda_i}{c\mu} < 1$ كذلك فإن $E\{\xi_0\} =$

وأن وقت الانتظار المتوقع لأي زبون في صف الانتظار وبقيّة المقاييس هي:-

$$W_q = \sum_{k=1}^m \frac{\lambda_k}{\lambda} \cdot W_q^{(k)}, \text{ and } L_q = \lambda \cdot W_q, \text{ Where } \lambda = \sum_{i=1}^m \lambda_i$$

$$L_q^{(k)} = \lambda_k \cdot W_q^{(k)}, \text{ and } L_s^{(k)} = \lambda \cdot W_s^{(k)}$$

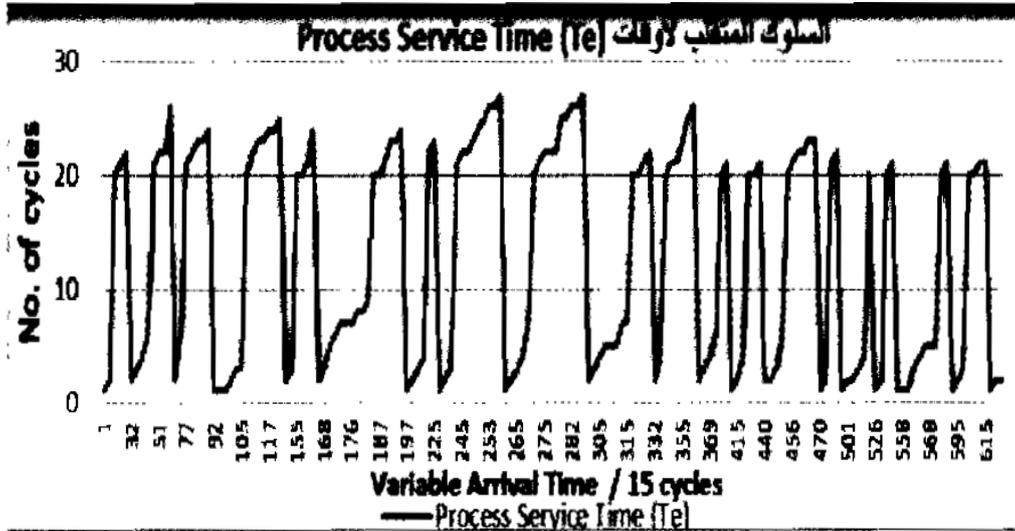
$$W_s^{(k)} = W_q^{(k)} + E_k\{t\}$$

$$W_s = \sum_{k=1}^m \frac{\lambda_k}{\lambda} \cdot W_s^{(k)}, \text{ and } L_s = \lambda \cdot W_s$$

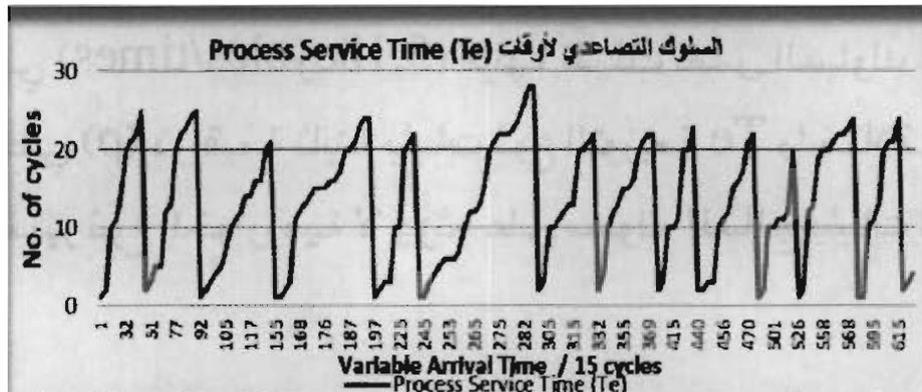
لقد تم تطبيق (48) حالة دراسية على نموذج $(M_i/M/C):(NPRP/\infty/\infty)$ ، بسبب استخدام 4 خوارزميات جدولة وهي (List, Bartal, Albers, Fleischer)، كل واحدة منها تعاملت مع 4 توزيعات للقيم العشوائية (عشوائي، تصاعدي، تنازلي، متقلب) للفترة التي تحتاجها المهام القادمة للمعالجة داخل المعالجات، وكل خوارزمية وتوزيع

قيم قد تعاملت مع 3 قيم مختلفة لعدد الدورات التي يحتاجها المعالج لمعالجة المهام القادمة، وهي (5, 10 and 15) دورة /وحدة زمنية، معتمدين على 15 معامل تخص نموذج صف الانتظار لتقييم سلوك وكفاءة الخوارزميات الأربع.

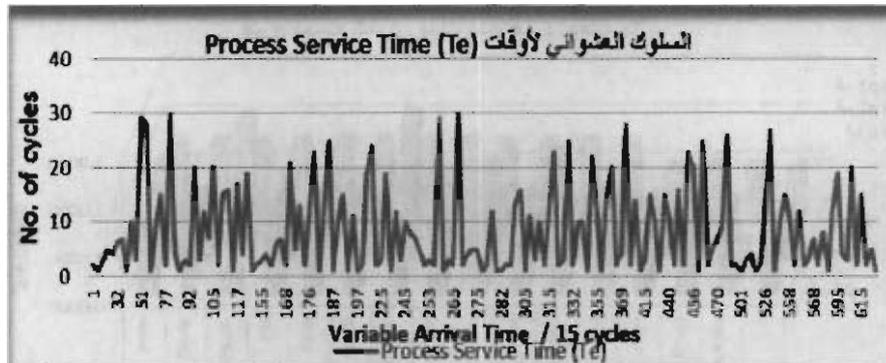
شكل رقم (1A) يوضح توزيع القيم العشوائي لأوقات تقديم الخدمة



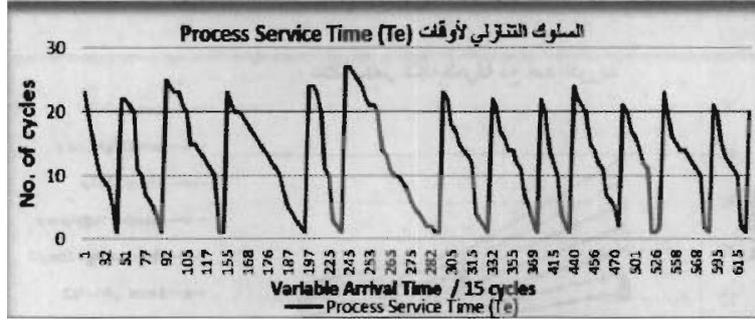
شكل رقم (1B) يوضح توزيع القيم التصاعدي لأوقات تقديم الخدمة



شكل رقم (1C) يوضح توزيع القيم التنازلي لأوقات تقديم الخدمة



شكل رقم (1D) يوضح توزيع القيم المتقلب لأوقات تقديم الخدمة



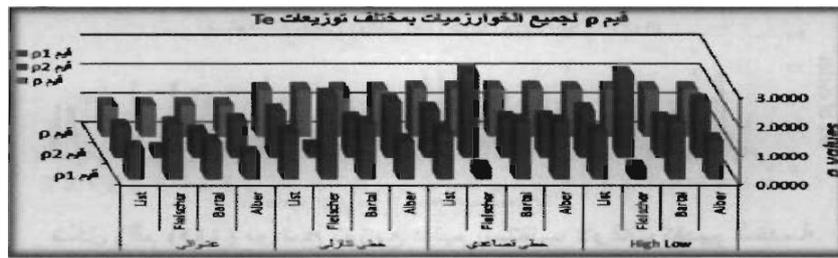
معاملات كثافة الحركة (p_1, p_2, p) المعدلة: -

نلاحظ من شكل رقم (2) أن هناك علاقة عكسية بين عدد دورات المعالج وأياً من قيم p أو p_1 أو p_2 ، وهذا يعني أنه يفضل أن تزداد عدد الدورات الخاصة بتنفيذ كل عملية، فبازديادها يقل الضغط على المعالج ويستطيع إنجاز عمليات أكثر ويقل معدل كثافة الحركة فيه.

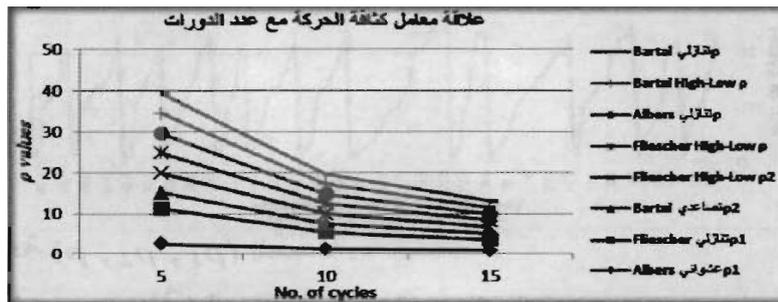
أما شكل رقم (3) فيوضح سلوك معامل كثافة الحركة لجميع الخوارزميات والتوزيعات Te مجتمعة معاً، ونلاحظ أن ازدياد معدل التحميل (الضغط) على المعالجات يعتمد على مقدار المهام الواصلة للنظام، هل هي أقل أو قريبة أو أكثر من مقدار معدل الخدمة التي يقدمها ذلك المعالج، لأن ذلك يؤثر بشكل مباشرة على أسلوب عمل كل خوارزمية وكيفية جدولة المهام على المعالجات.

توضح الأشكال (4 (A, B, C, D) كيفية سلوك معاملات كثافة الحركة لجميع الخوارزميات ولجميع أنواع توزيع القيم العشوائية ولعدد دورات (15 cycles/ times) فقط، أما بقية أنواع الدورات وهي (5,10 cycles/times) فإنها تسلك نفس السلوك، ونلاحظ أن قيمة معدل الحركة الكلي (p) ذو قيمة ثابتة بثبات نوع التوزيعة Te واختلاف نوع الخوارزمية، وهذا يدل على أن تغيير نوع الخوارزمية لا يؤثر على سلوك المعالج بثبات نوع التوزيعة وعدد الدورات.

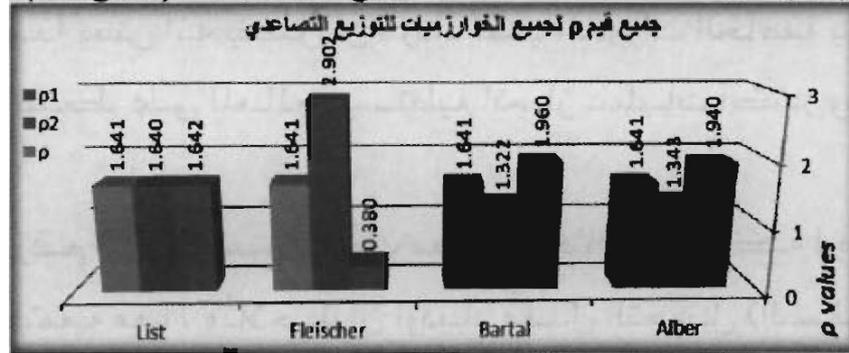
شكل رقم (2) سلوك معاملات كثافة الحركة لجميع الخوارزميات وبشكل منوع للتوزيعة



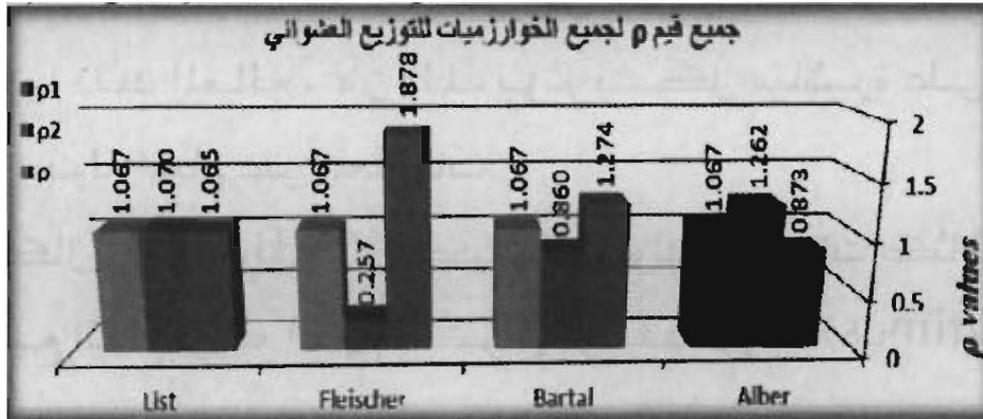
شكل رقم (3) سلوك معامل كثافة الحركة p لجميع الخوارزميات والتوزيعات Te مجتمعة معاً



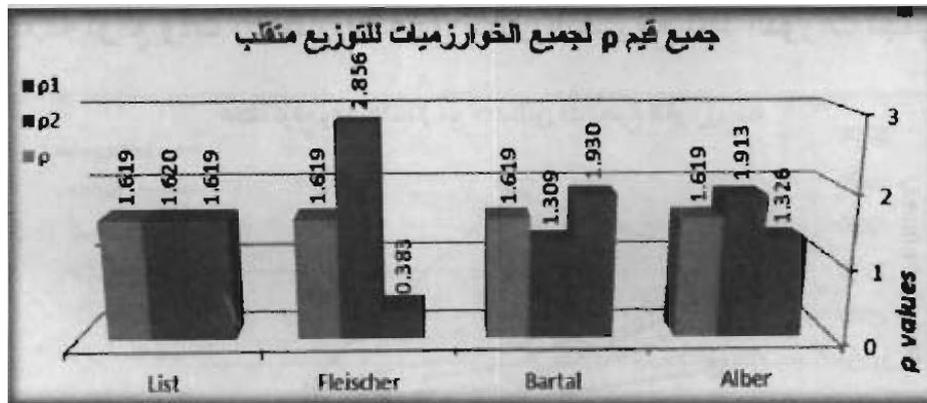
شكل رقم (4A) سلوك معاملات كثافة الحركة لجميع الخوارزميات (توزيع القيم عشوائي)



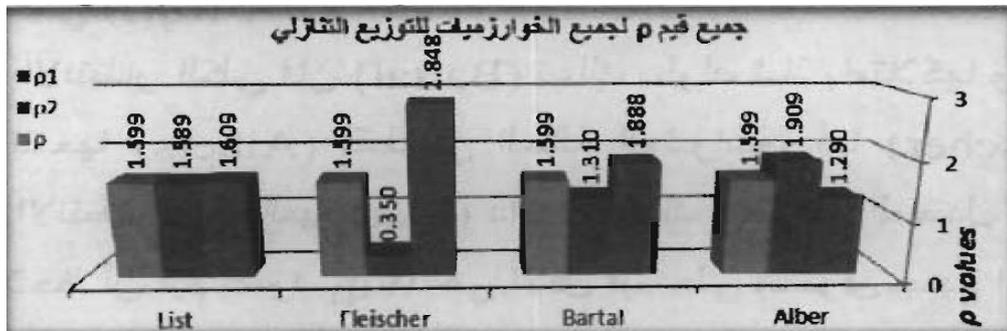
شكل رقم (4B) سلوك معاملات كثافة الحركة لجميع الخوارزميات (توزيع القيم تصاعدي)



شكل رقم (4C) سلوك معاملات كثافة الحركة لجميع الخوارزميات (توزيع القيم تنازلي)



شكل رقم (4D) سلوك معاملات كثافة الحركة لجميع الخوارزميات (توزيع القيم متقلب)



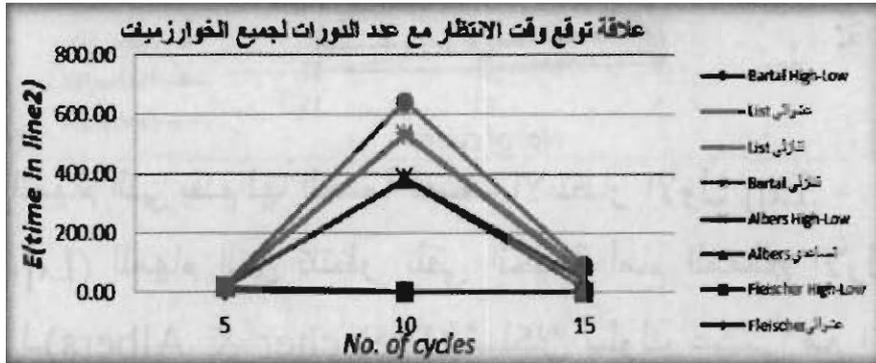
توقع وقت الانتظار لصف الانتظار الأول (T) wq_1 :-

فيما يتعلق بالزمن الذي تنتظره الدورة الواحدة من التعليم لكل ثانية (cycles/time) في صف الانتظار أمام المعالج الأول، فإن (Albers & Fleischer) تسلكان سلوك عكسي واضح تتبعها (Bartal) فقط في الحالة العشوائية، وتخالف (Bartal) الجميع بامتلاكها أعلى قيمة شاذة عند عدد الدورات 10 في التوزيع (تنازلي، تصاعدي، متقلب)، وتشابهها بهذا السلوك (list) ولكن بقيمة شاذة صغيرة كما في شكل (5).

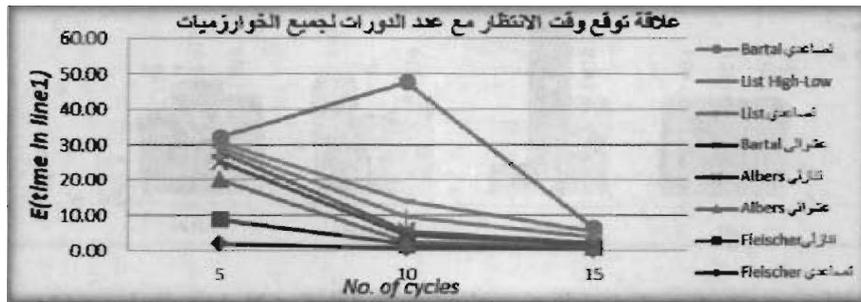
توقع وقت الانتظار لصف الانتظار الثاني (T) wq_2 = E (time in line 2) :-

فيما يتعلق بالزمن الذي تنتظره الدورة الواحدة من التعليم لكل ثانية في صف الانتظار أمام المعالج الثاني، فإن خوارزمية (Bartal) تسلك سلوك شاذ بامتلاكها وقت انتظار عال عند عدد الدورات 10، تتبعها خوارزمية (Albers) فقط في الحالة التصاعدية، أما (Fleischer) فذات سلوك عكسي معتدل لوقت الانتظار، وتخالفها خوارزمية (List) ذات السلوك الطردي المعتدل لوقت الانتظار، كما موضح في شكل رقم (6).

شكل رقم (5) علاقة توقع وقت الانتظار لصف الانتظار الأول مع عدد الدورات لجميع الخوارزميات

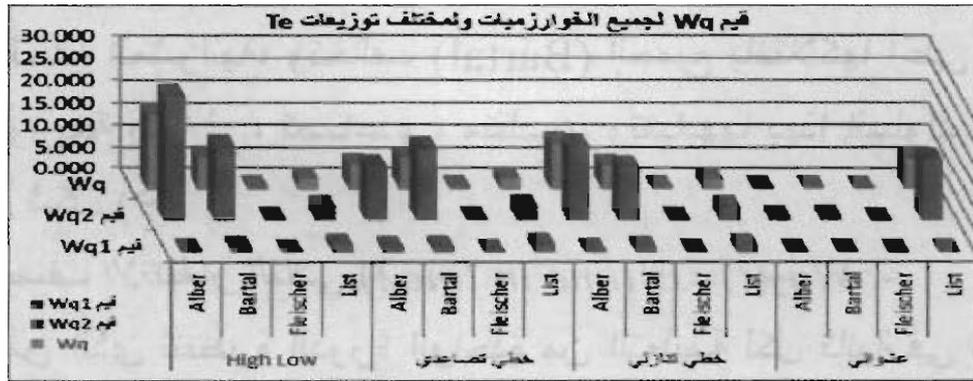


شكل رقم (6) علاقة توقع وقت الانتظار لصف الانتظار الثاني مع عدد الدورات لجميع الخوارزميات

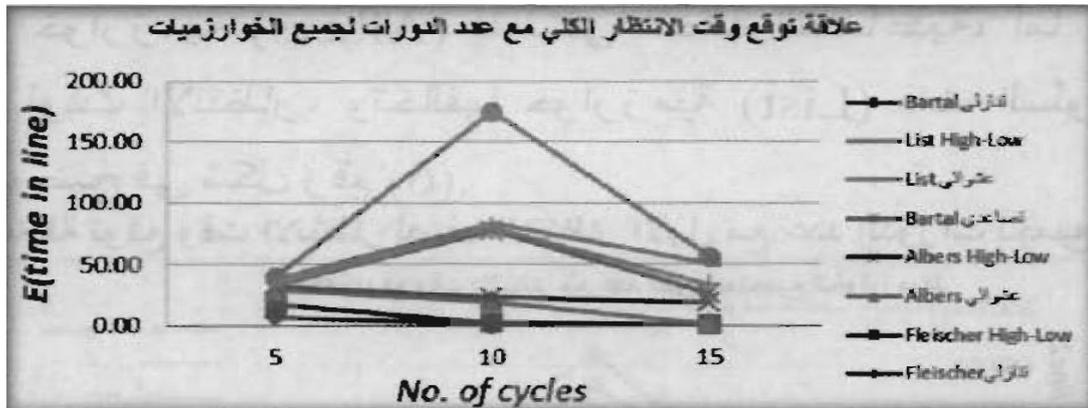
توقع وقت الانتظار الكلي (t) Wq :-

فيما يتعلق بوقت الانتظار الكلي فإن (Bartal) تسلك سلوك شاذ بامتلاكها وقت انتظار عال عند عدد الدورات 10، تتبعها (Albers) فقط في الحالة العشوائية، أما (Fleischer) فذات سلوك عكسي معتدل لوقت الانتظار، وتخالفها (List) ذات السلوك الطردي المعتدل لوقت الانتظار، كما في شكل رقم (7)، نلاحظ أن قيم عمود wq_1 هي الأقل قيم على مستوى جميع الخوارزميات وأنواع التوزيعات، كما يوضح شكل رقم (8) أن (Fleischer) تملك أقل توقع لوقت الانتظار عند صفوف الانتظار الفرعية والكلي لجميع أنواع الخوارزميات والتوزيعات.

شكل رقم (7) علاقة توقع وقت الانتظار الكلي مع عدد الدورات لجميع الخوارزميات



شكل رقم (8) سلوك وقت الانتظار Wq لجميع الخوارزميات والتوزيعات Te مجتمعة معاً



القيمة المتوقعة لعدد المهام التي يقدم لها الخدمة لصف الانتظار الأول Lq_1 :-

فيما يتعلق بـ (Lq_1) للمهام التي تنتظر تلقي الخدمة أمام المعالج الأول، نلاحظ أن القيمة المتوقعة لعدد المهام لـ (Fleischer & Albers) تسلكان سلوك عكسي مع ازدياد عدد الدورات، بغض النظر عن نوع توزيع القيم العشوائي فكلما ازداد عدد الدورات قل عدد المهام المنتظرة تلقي الخدمة أمام المعالج، أما (Bartal) فإنها تسلك سلوك شاذ بامتلاكها طول صف انتظار عال عند عدد الدورات 10 للتوزيع التنازلي، وتشبهها (List) بسلوكها الشاذ بامتلاكها طول صف انتظار معتدل عند عدد الدورات 10، كما موضح في شكل رقم (9).

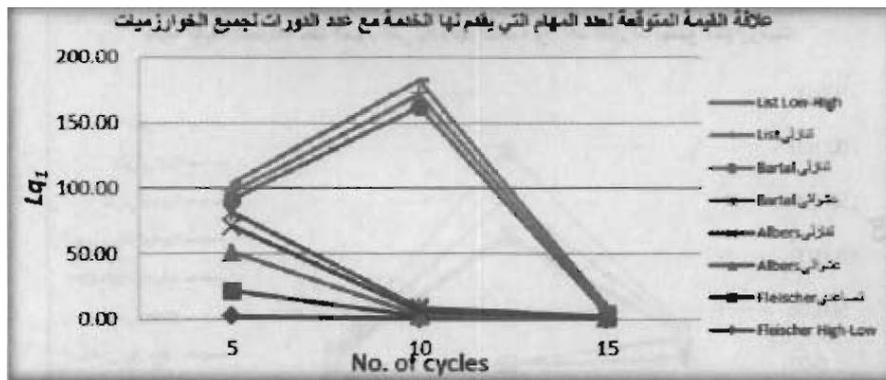
القيمة المتوقعة لعدد المهام التي تقدم لها الخدمة لصف الانتظار الثاني Lq_2 :-

بالنسبة لقيمة Lq_2 ، نلاحظ أن القيمة المتوقعة لعدد المهام لـ (Fleischer) تسلك سلوك عكسي مع ازدياد عدد الدورات، فكلما ازداد عدد الدورات قل عدد المهام المنتظرة تلقي الخدمة أمام المعالج، وعلى خلافها (Albers) التي تسلك سلوك طردي مع عدد الدورات، أما (Bartal) فإنها تسلك سلوك شاذ بامتلاكها طول صف انتظار عال عند 10 دورات للتوزيع التصاعدي، وتشبهها (List) بسلوكها بامتلاكها طول صف انتظار معتدل عند 10 دورات، كما في شكل (10).

شكل رقم (9) علاقة عدد المهام المنتظرة أمام صف الانتظار الأول مع عدد الدورات لجميع الخوارزميات



شكل رقم (10) علاقة عدد المهام المنتظرة أمام صف الانتظار الثاني مع عدد الدورات لجميع الخوارزميات



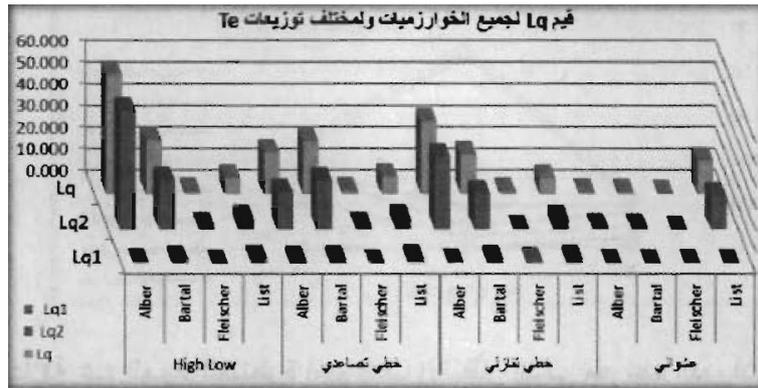
القيمة المتوقعة لعدد المهام التي يقدم لها الخدمة لصف الانتظار الكلي Lq : -

فيما يتعلق بـ (Lq) للمهام التي تنتظر تلقي الخدمة أمام المعالجين، نلاحظ أن (Fleischer) تسلك سلوك عكسي حاد مع ازدياد عدد الدورات، فكلما ازداد عدد الدورات قل عدد المهام المنتظرة تلقي الخدمة أمام المعالجين، أما (Bartal) فإنها تسلك سلوك شاذ بامتلاكها طول صف انتظار عال عند عدد الدورات 10، وتشبهها (Albers) بسلوكها الشاذ لحالة التوزيع التصادمية كما في شكل (11)، بخلاف الجميع فإن (List) غير مستقرة ومتقلبة بين قيمة وأخرى لجميع أنواع توزيعات القيم العشوائية، ونلاحظ أن قيم $Lq1$ تملك أقل قيم على مستوى جميع الخوارزميات وأنواع التوزيعات، من شكل رقم (12) نلاحظ أن (Fleischer) تملك أقل توقع لطول صف الانتظار عند صفوف الانتظار الفرعية والكلية لجميع أنواع الخوارزميات والتوزيعات.

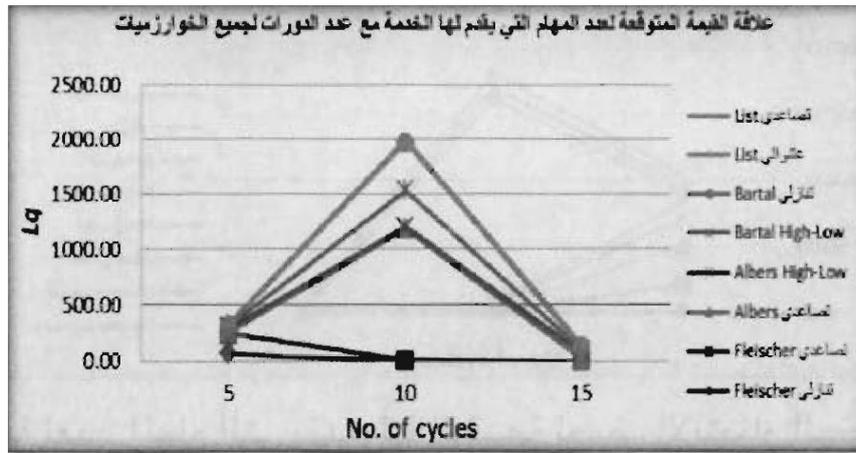
القيمة المتوقعة لوقت انتظار مهمة ما في النظام عند صف الانتظار الأول $Ws1$: -

فيما يتعلق بقيمة $(Ws1)$ والمحسوب بوحدة (cycles)، فسيتم ضربها بمعدل الخدمة (μ) وذلك لتحويل الوحدات إلى (cycles/ second). نلاحظ أن قيمة $(Ws1)$ لخوارزميتي (Fleischer & Albers) تسلكان سلوكاً عكسياً مع ازدياد عدد الدورات، أما (Bartal) فإنها تسلك سلوك شاذ بامتلاكها وقت انتظار عال عند عدد الدورات 10، وبخلاف الجميع فإن (List) غير مستقرة ومتقلبة بين قيمة وأخرى لجميع أنواع توزيعات القيم العشوائية، كما موضح في شكل رقم (13).

شكل رقم (11) علاقة عدد المهام المنتظرة أمام صف الانتظار الكلي مع عدد الدورات لجميع الخوارزميات



شكل رقم (12) سلوك وقت الانتظار Lq لجميع الخوارزميات والتوزيعات Te مجتمعة معاً

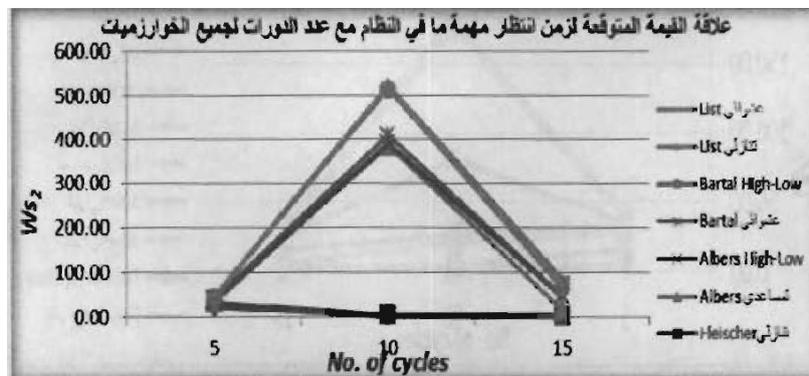


القيمة المتوقعة لوقت انتظار مهمة ما في النظام عند صف الانتظار الثاني Ws_2 :-

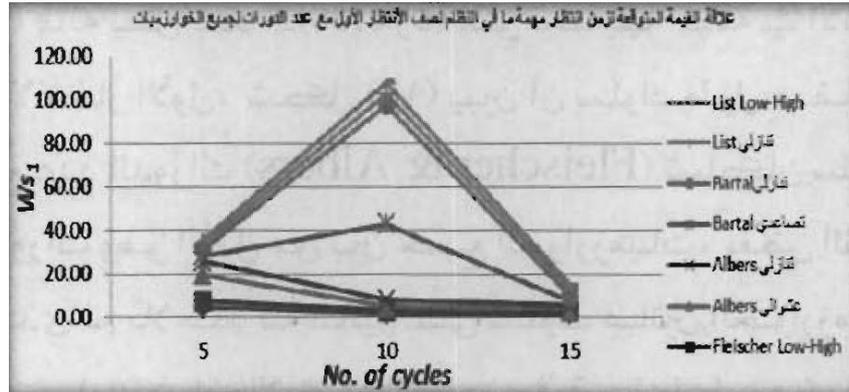
نلاحظ أن قيمة (Ws_2) (Fleischer لـ) تسلك سلوك عكسي مع ازدياد عدد الدورات، بخلاف (List) ذات السلوك الطردي بشكل بسيط لجميع أنواع توزيعات القيم العشوائية، أما (Bartal & Albers) فإنهما تسلكان سلوك شاذ بامتلاكها وقت انتظار عال عند عدد الدورات 10، كما في شكل (14)

شكل رقم (13) علاقة القيمة المتوقعة لزمن انتظار مهمة ما في النظام لصف الانتظار الأول مع عدد الدورات

لجميع الخوارزميات



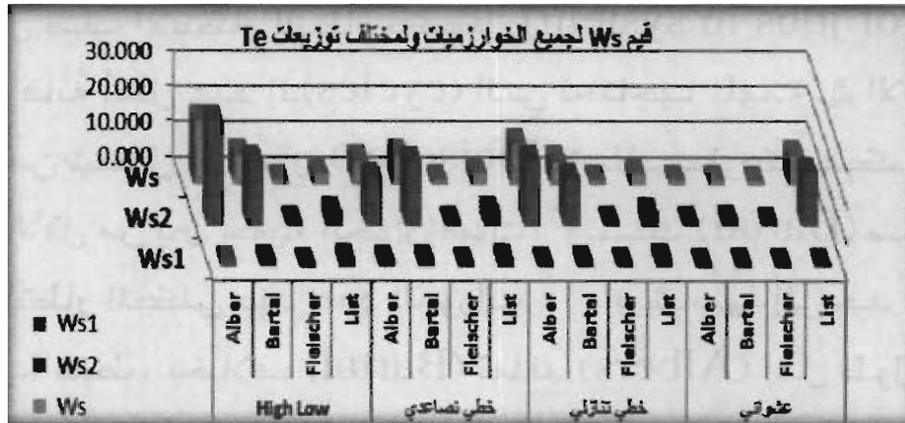
شكل رقم (14) علاقة القيمة المتوقعة لزمن انتظار مهمة ما في النظام لصف الانتظار الثاني مع عدد الدورات لجميع الخوارزميات

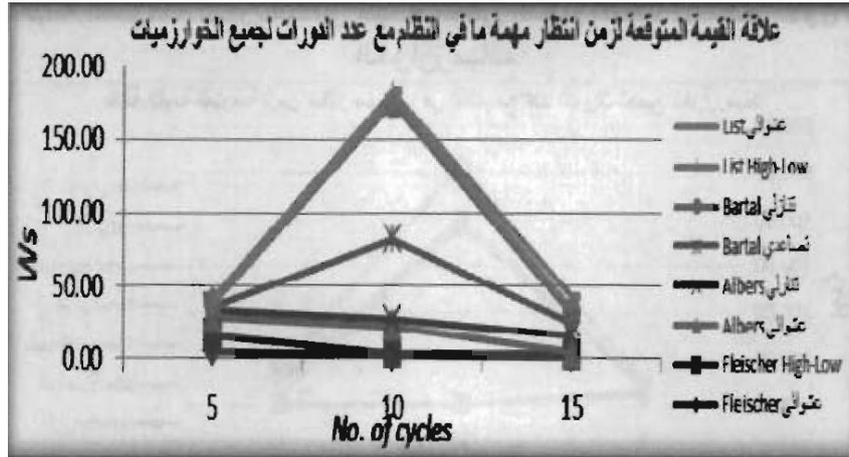


القيمة المتوقعة لوقت انتظار مهمة ما في النظام W_s :

نلاحظ أن قيمة W_s لـ (Fleischer) تسلك سلوكا عكسيا مع ازدياد عدد الدورات، بخلاف (List) ذات السلوك الطردي البسيط لجميع أنواع توزيعات القيم العشوائية، أما (Bartal) فإنها تسلك سلوك شاذ بامتلاكها وقت انتظار عالي عند عدد الدورات 10، ويشابهها بهذا السلوك (Albers) عند توزيع القيم العشوائي كما موضح في شكل رقم (15)، ونلاحظ أن قيم عمود W_s1 تملك أقل قيم على مستوى جميع الخوارزميات وأنواع التوزيعات، من شكل رقم (16) نلاحظ أن (Fleischer) تملك أقل توقع لطول صف الانتظار عند صفوف الانتظار الفرعية والكلية لجميع أنواع الخوارزميات والتوزيعات

شكل رقم (15) علاقة القيمة المتوقعة لزمن انتظار مهمة ما في النظام مع عدد الدورات لجميع الخوارزميات



شكل رقم (16) سلوك وقت الانتظار W_s لجميع الخوارزميات والتوزيعات Te مجتمعة معاً

القيمة المتوقعة لطول صف الانتظار داخل النظام والمقابل لصف الانتظار الأول (LS_1): -

بالنسبة لـ (LS_1) فإنه يمثل عدد الـ (cycles) التي تحتاجها المهمة في الانتظار داخل النظام والقادمة من صف الانتظار الأول، شكل (17) يبين أن سلوك طول صف الانتظار متغير من خوارزمية لأخرى مع عدد الدورات (Fleischer & Albers) تسلكان سلوكاً عكسياً حاد مع ازدياد عدد الدورات وهو الأقل من بين جميع الخوارزميات، بغض النظر عن نوع توزيع القيم العشوائي الذي لم نلاحظ له تأثير على سلوك هاتين الخوارزمتين، بينما تسلك (Bartal) كالعادة سلوك شاذ بالارتفاع الكبير في قيمة طول صف الانتظار عند عدد الدورات 10، ويشابهها إلى حد بسيط (List) فهي تملك علاقة متقلبة تختلف حسب نوع توزيع القيم العشوائية وعلى العموم فهي متقلبة في مدى متقارب وليس هناك تطرف يذكر في سلوكها مقارنة مع خوارزمتي (Albers & Bartal).

القيمة المتوقعة لطول صف الانتظار داخل النظام والمقابل لصف الانتظار الثاني (LS_2): -

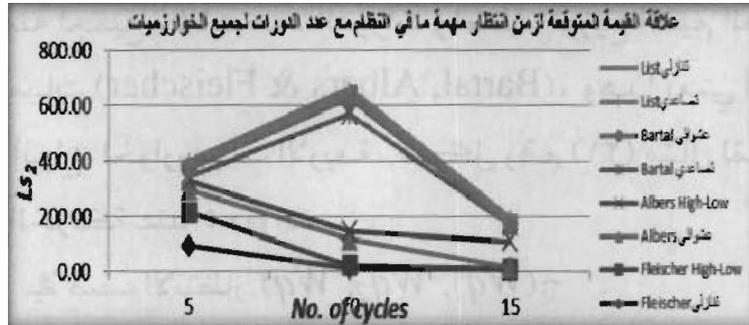
أن قيمة (LS_2) لـ (Fleischer) تسلك سلوكاً عكسياً حاد مع ازدياد عدد الدورات وهو الأقل من بين جميع الخوارزميات، بينما تسلك (Bartal & List) سلوكاً شاذاً بارتفاع قيمتها عند عدد الدورات 10، ويشابهها إلى حد بسيط (Albers) عند توزيعه القيم العشوائية فقط، كما في شكل رقم (18).

القيمة المتوقعة لطول صف الانتظار للنظام ككل (E (no. of jobs in system))

بالنسبة لـ (LS) فإنه يمثل عدد الـ (cycles) التي تحتاجها المهمة في الانتظار داخل النظام ككل، ونلاحظ من شكل (19) أن (Fleischer) تسلك سلوكاً عكسياً حاداً مع ازدياد عدد الدورات وهو الأقل من بين جميع الخوارزميات، وتسلك (Bartal) سلوكاً شاذاً بارتفاع قيمة طول صف الانتظار الكلي عند عدد الدورات 10، ويشابهها إلى حد بسيط (List) عند توزيعه القيم (متقلب) فقط، بخلاف (Bartal) تملك (Albers) أقل طول صف انتظار عند عدد الدورات 10، وهذا مغاير لجميع الخوارزميات فيما جرى لعلاقتها مع عدد الدورات.

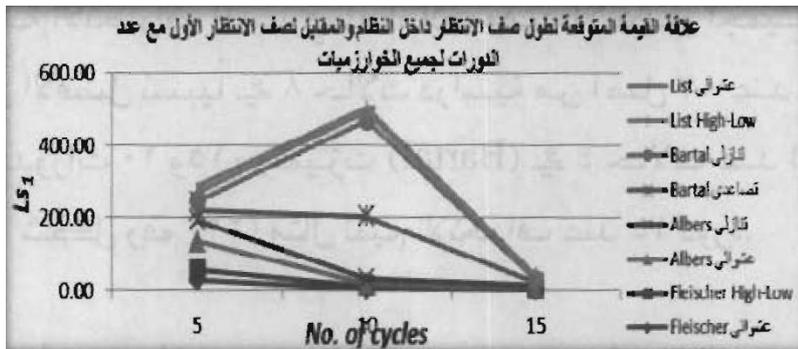
من شكل رقم (20) نلاحظ أن قيم عمود LS_1 تملك أقل قيم على مستوى جميع الخوارزميات وأنواع التوزيعات، وأن (Fleischer) تملك أقل توقع لطول صف الانتظار عند صفوف الانتظار الفرعية والكلبي لجميع أنواع الخوارزميات.

علاقة القيمة المتوقعة لزمن انتظار مهمة ما في النظام مع عدد الدورات لجميع الخوارزميات

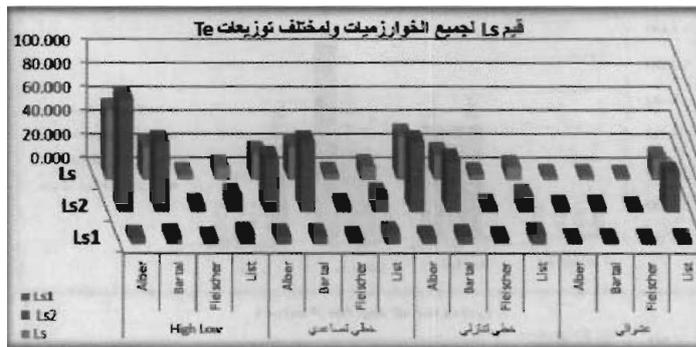


علاقة القيمة المتوقعة لطول صف الانتظار داخل النظام والمقابل لصف الانتظار الأول مع

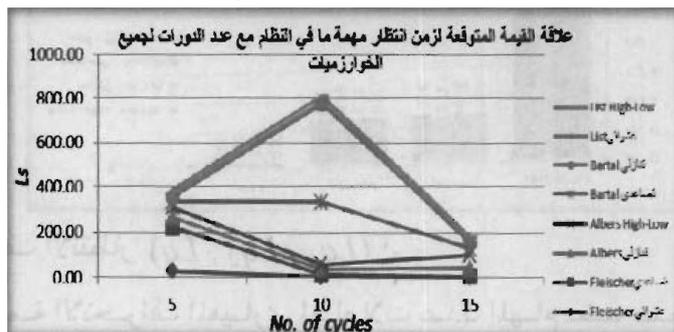
عدد الدورات لجميع الخوارزميات



قيم LS لجميع الخوارزميات ولمختلف توزيعات Te



علاقة القيمة المتوقعة لزمن انتظار مهمة ما في انتظام مع عدد الدورات لجميع الخوارزميات



تحليل نتائج الانحراف المعياري لجميع الخوارزميات :-

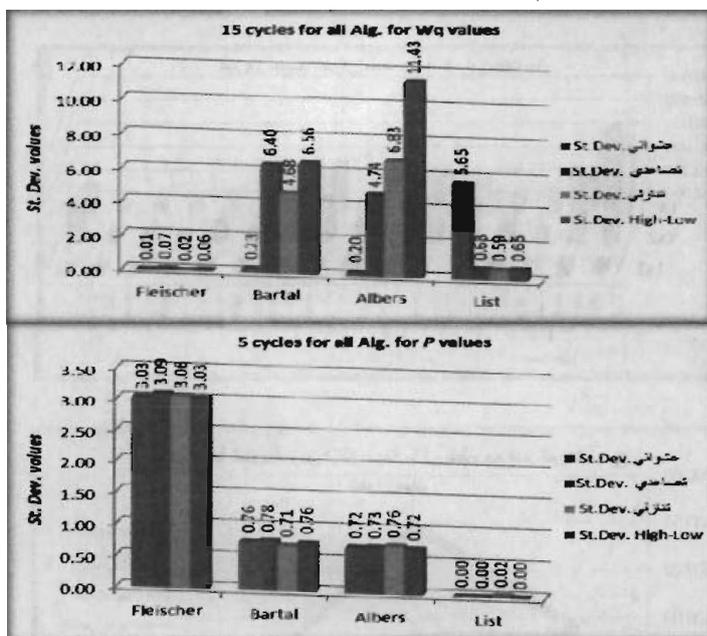
كثافة الحركة (p1, p2, p) :-

لقد أثبتت (List) بامتلاكها أقل انحراف معياري لمعاملات كثافة الحركة مساوية للصفر أو قريبة منة لجميع حالات عدد الدورات ولمختلف توزيع القيم العشوائية ولم يصل إليها أي من خوارزميات (Bartal, Albers & Fleischer)، وهذا يعني أن هنالك (3x4=12) حالة مقارنة بين أنواع الخوارزميات الأربعة، شكل رقم (21) مثال لقيم الانحراف المعياري لمعاملات كثافة الحركة عند 5 دورات.

وقت الانتظار في صف الانتظار (Wq1, Wq2, Wq) :-

بالنسبة لقيمة الانحراف المعياري لمعاملات وقت الانتظار لجميع الخوارزميات، فإن (Fleischer) هي الأفضل نسبيا في 8 حالات دراسية من أصل 12 عند جميع قيم التوزيعات العشوائية لعدد الدورات 10 و 15، وتميزت (Bartal) في 4 حالات عند الـ 5 دورات، وتقاربت منها (Albers)، شكل رقم (22) مثال لقيم الانحراف عند 15 دورة.

شكل رقم (22) مثال لقيم الانحراف المعياري لمعاملات وقت الانتظار عند 15 دورة



عدد المهام عند صف الانتظار (Lq1, Lq2, Lq) :-

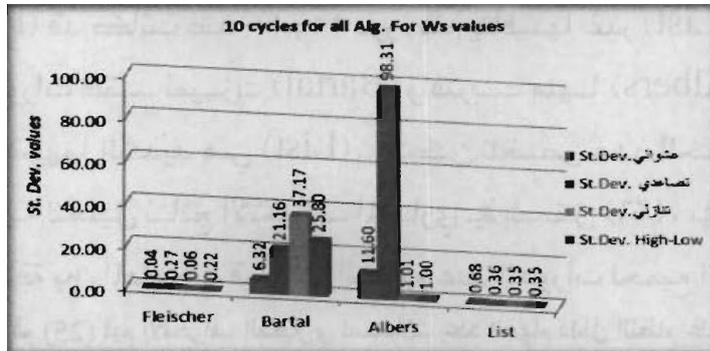
بالنسبة لقيمة الانحراف المعياري لمعاملات عدد المهام عند صفوف الانتظار لجميع الخوارزميات، فإن (Fleischer) هي الأفضل نسبيا في 8 حالات دراسية من أصل 12 عند جميع قيم التوزيعات العشوائية لعدد الدورات 10 و 15، وتميزت (List) في 4 حالات عند الـ 5 دورات، وتقاربت (List) من (Fleischer) في الحالات الثمان أعلاه، شكل رقم (23) مثال لقيم الانحراف المعياري لمعاملات عدد المهام عند صفوف الانتظار عند 10 دورات.

وقت انتظار المهام داخل النظام (Ws1, Ws2, Ws) :-

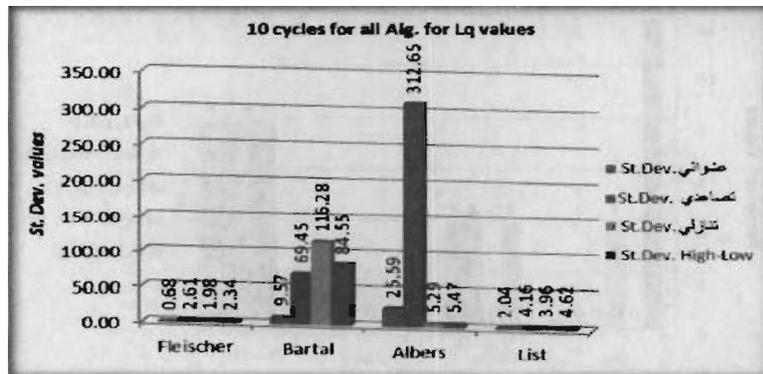
بالنسبة لقيمة الانحراف المعياري لمعاملات وقت انتظار المهام داخل النظام لجميع الخوارزميات، فإن (Fleischer) هي الأفضل نسبيا في 8 حالات دراسية عند جميع قيم التوزيعات العشوائية لعدد الدورات 10 و 15،

وتتميزت (Bartal) في 4 حالات عند الـ 5 دورات، وتقاربت (List) من الجميع في الحالات 13 أعلاه، شكل رقم (24) مثال لقيم الانحراف المعياري لمعاملات وقت الانتظار عند 10 دورة.

شكل رقم (23) قيم الانحراف المعياري لمعاملات عدد المهام عند 10 دورات



شكل رقم (24) قيم الانحراف المعياري لمعاملات وقت الانتظار داخل النظام عند 10 دورات



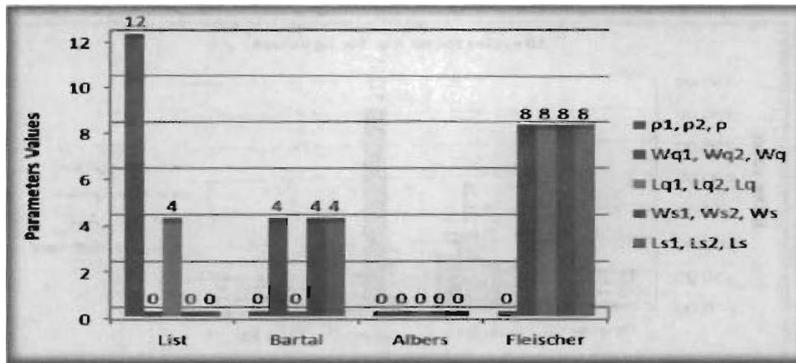
طول صف الانتظار داخل النظام (Ls1, Ls2, Ls) :-

بالنسبة لقيمة الانحراف المعياري لمعاملات عدد المهام في صف الانتظار داخل النظام لجميع الخوارزميات، فإن (Fleischer) هي الأفضل نسبيا في 8 حالات دراسية عند جميع قيم التوزيعات العشوائية لعدد الدورات 10 و 15، وتميزت (Bartal) في 4 حالات عند الـ 5 دورات، وتقاربت (List) من الجميع في الحالات 12 أعلاه، شكل رقم (25) مثال لقيم الانحراف داخل النظام عند 15 دورة.

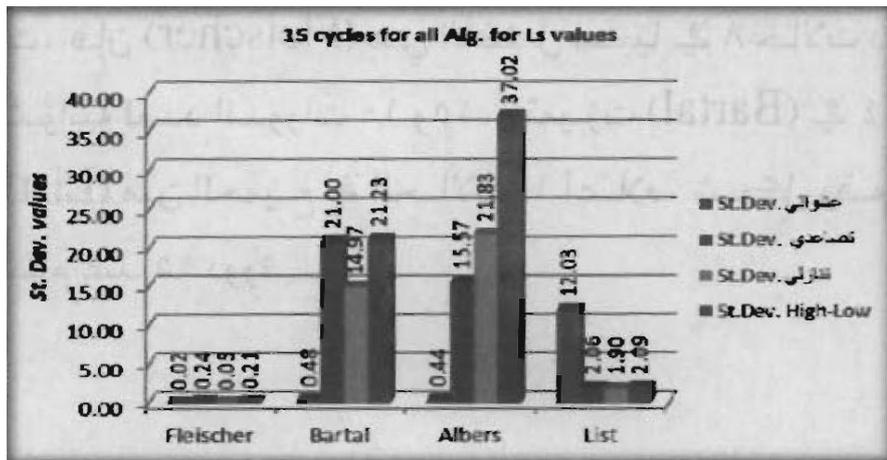
الاستنتاجات Conclusions

- تعزى النتائج الجيدة إلى الدور الفعال لأساليب بحوث العمليات وما قدمته نظرية صفوف الانتظار من تحسين لعمل خوارزميات جدولة المعالج.
- إن تقنيات المحاكاة تزودنا بطريقة منهجية مقبولة من أجل دراسة الأنظمة المعقدة، ويتضمن ذلك دراسة أزمنة انتظار المهام لأجل تنفيذها في أنظمة تعدد المعالجات، تحت افتراض استخدام استراتيجيات جدولة مختلفة، بينما يكون من المكلف بناء أنظمة تستخدم للدراسات التجريبية والاختبار، لهذا يتم اعتماد المحاكاة لحل القضايا المعقدة بأقل جهد وتكلفة.
- بالنظر إلى جميع حالات عدد الدورات نجد 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) واقتربت منها (Albers) بفارق الكسور العشرية ومنافسهما الشديد هي (List)، يمكن تلخيص عدد الحالات التي تميزت بها كل خوارزمية لتحليل نتائج الانحراف المعياري في شكل (26)، في حين يلخص جدول رقم (3) العلاقة بين المعاملات قيد الدراسة مع عدد الدورات لجميع الخوارزميات.

شكل رقم (25) قيم الانحراف المعياري لمعاملات عدد المهام داخل النظام عند 15 دورة



شكل رقم (26) عدد الحالات التي تميزت بها كل خوارزمية لتحليل نتائج الانحراف المعياري



جدول رقم (3) العلاقة بين المعاملات قيد الدراسة مع عدد الدورات لجميع الخوارزميات

المعامل	List	Bartal	Albers	Fleischer
P_1 P_2 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
Wq_1 Wq_2 Wq	شاذة مقارنة لـ Bartal طردي معتدل طردي معتدل	أعلى قيمة شاذة أعلى قيمة شاذة أعلى قيمة شاذة	علاقة عكسية مقارنة لـ Bartal مقارنة لـ Bartal	علاقة عكسية عكسي معتدل عكسي معتدل
Lq_1 Lq_2 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
Ws_1 Ws_2 Ws	أعلى قيمة شاذة علاقة طردية طردي معتدل	أعلى قيمة شاذة أعلى قيمة شاذة أعلى قيمة شاذة	علاقة عكسية أعلى قيمة شاذة أعلى قيمة شاذة	علاقة عكسية علاقة عكسية علاقة عكسية
Ls_1 Ls_2 Ls	أعلى قيمة شاذة أعلى قيمة شاذة علاقة متقلبة	أعلى قيمة شاذة أعلى قيمة شاذة أعلى قيمة شاذة	عكسي حاد شاذة مقارنة لـ Bartal مقارنة لـ Fleischer	عكسي حاد عكسي حاد عكسي حاد

التوصيات Recommendations:

- اعتماد بيانات حقيقية للوصول المهام مأخوذة من شركات عملاقة كشركة Intel و AMD لتقييم سلوك الخوارزميات.
- بسبب التحسين الذي قدمته بحوث العمليات لخوارزميات الجدولة، نوصي باستخدامها خاصة في الحالات التي تحاول إيجاد الأسلوب الأمثل لمواردها.

الدراسات المستقبلية Future studies :

- يقترح الباحث عمل خوارزمية تمزج بين جميع الخوارزميات الأربعة، أو كل اثنين أو ثلاث معاً وذلك بعد دراسة سلوك معاملات نظرية صفوف الانتظار، ومن ثم تطوير شروط عمل الخوارزمية الجديدة.
- 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.

المراجع

1. زياد القاضي، المفاهيم الأساسية في أنظمة التشغيل، الطبعة الأولى، مكتبة المجتمع العربي للنشر والتوزيع، صفحة 29، 2004 م.
2. قصي حميد السلامي، 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، 2012
3. محمد إبراهيم العدوي، المعالجات الدقيقة Microprocessors البرمجة والمواجهة والتطبيق (-pentium-) intel18085-z80-intel18086-intel180186-intel180286-intel180386-intel180486، الدار الدولية للاستثمارات الثقافية، مصر، الطبعة الأولى، 2000.

4. Albert Y.Z., Matthew C., and Stephan O. Iariu, A Framework for Reinforcement- Based Scheduling in Parallel Processor Systems, IEEE Transactions on Parallel and distributed systems, Vol.9, No.3, MARCH 1998, pp.249.
5. Graham R.L., Bounds for certain multi-processing anomalies, Bell System Technical Journal '45:1563-1581, 1966.
6. Raihan Al-Ekram, Online Multiprocessor Scheduling, 2004. Available for download at <http://ametminer.org/dev.do?m=downloadpdf&url=http://ametminer.org/pdf/PDFFiles2>.

7. Susanne Albers, Bianca Schroder, An Experimental Study of Online Scheduling Algorithms, 2000, available for download at <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.8580&rep=repl&type=pdf>