المجلة العراقية للعلوم الاحصائية (20) 2011
عدد خاص بوقائع المؤتمر العلمي الرابع كلية علوم الحاسوب والرياضيات
ص ص [612– 625]

خوارزمية مقترحة تعتمد أسلوب البرمجة الخطية لجدولة ثلاثة أعمال في بيئة الورشة الانسيابية ذات الثلاث مراحل بمعيار تقليل وقت التنفيذ

مناف حازم احمد* منال عبد الكريم العباجي**
MANAF HAZIM AHMED MANAL ABDULKAREEM

الملخص

في هذا البحث تم اقتراح خوارزمية تقوم بإيجاد أفضل جدولة لثلاثة أعمال (jobs) في بيئة الورشة الانسيابية بثلاث مراحـل (flow shop environment with three stage) بحيث يكون وقت التنفيـذ (makespan) اقـل مـا يمكن . تعتمد هذه الخوارزمية في حساب وقت التنفيذ على أسلوب البرمجة الخطية، إذ تمت صياغة نموذج دالة الهدف والقيود بحيث يطابق هذا النموذج شروط الأسبقيات بين العمليات (Operations) والتي تحكمها بيئة الورشة الانسيابية. و بعد تطبيق هذه الخوارزمية على عدة مسائل تم توليدها عشوائيا من التوزيـع المنـتظم، أظهـرت النتائج ان أوقات التنفيذ التي أوجدتها هذه الخوارزمية مكافئة لخوارزميات الجدولة الأخرى.

A proposed algorithm based on linear programming approach for three-stage three jobs flow shop scheduling with criteria : minimize makespan

Abstract

This research has proposed an algorithm to find the optimal schedule for three jobs in flow –shop environment with three stages , so that the makespan is less than what can be.

* مدرس مساعد L كلية علوم الحاسوب والرياضيات/جامعة الموصل manaf_matrood@yahoo.com

** مدرس مساعد /كلية علوم الحاسوب والرياضيات/ جامعة الموصل

تاريخ التسلم :2011/10/1 ـــــــــــ تاريخ القبول : 21 /12 /2011

This algorithm is based on the linear programming approach in computing the makespan ,the objective function and constraints have been formulated under the conditions of precedence between operations in flow shop environment .

After applying this algorithm on several problems which are generated randomly by uniform distribution ,the results showed that the makespan which is computed by the proposed algorithm is equivalent to other scheduling algorithms .

1- المقدمة:

من الأمور المهمة في مسائل الجدولة هو إيجاد أفضل ترتيب لمجموعة من الإعمال (jobs) عند المعالجة بحيث يعطي هذا الترتيب اقل وقت تنفيذ ممكن (makespan) , ففي مسائل الورشة الانسيابية (flow – shop) يوجد هناك (m) من الماكنات المختلفة و (n) من الأعمال , كل عمل من هذه الأعمال يحتوي على (m) من العمليات (operation) , كل عملية من هذه العمليات تعالج بماكنة مختلفة عن الأخرى بحيث انه لا يمكن معالجة أية عملية إلا بعد الانتهاء من معالجة العملية التي قبلها وعند نفس العمل وهكذا مع باقي الأعمال.

لدراسة الورشة الانسيابية (flow – shop) يؤخذ بنظر الاعتبار الفرضيات الآتية (Etiler and Wilson, 2004)

- جميع الأعمال جاهزة للمعالجة عند الوقت $t = 0$.
- جميع الماكنات تكون متوافرة و بصورة مستمرة.
- جميع أوقات المعالجة للعمليات اكبر أو تساوي صفراً.
- جميع الماكنات لا يمكن أن تعالج أكثر من عملية واحدة في الوقت نفسه.

في هذا البحث تم تحويل لوحة كانت Gant chart و التي تمثل الورشة الانسيابية إلى بيان حلقي مباشر (DAG) إذ إن العقد في هذا البيان تمثل أوقات البدء و الانتهاء للعمليات و الحافات تمثل أوقات المعالجة و بعدها تمت صياغة نموذج خطي بدالة هدف و قيود إذ أن دالة الهدف تمثل وقت التنفيذ للمسألة و القيود هي

فروض المسألة وكذلك قيود الأسبقية بين العمليات , و لما كانت المسألة هي تقليـل وقت التنفيذ وان هناك (6) احتمالات من المتتابعات للأعمال الثلاث و الهدف هـو إيجاد أفضل متتابعة من بين المتتابعات الست بحيث تعطي اقل وقت تنفيذ ,تم أنشاء خوارزمية تأخذ الاحتمالات الست وكل احتمال من هـذه الاحتمـالات يـتم حلـه بالنموذج الخطي الذي تتم صياغته ومن ثم يتم تحديد المتتابعة التي تعطي اقل وقت تنفيذ.

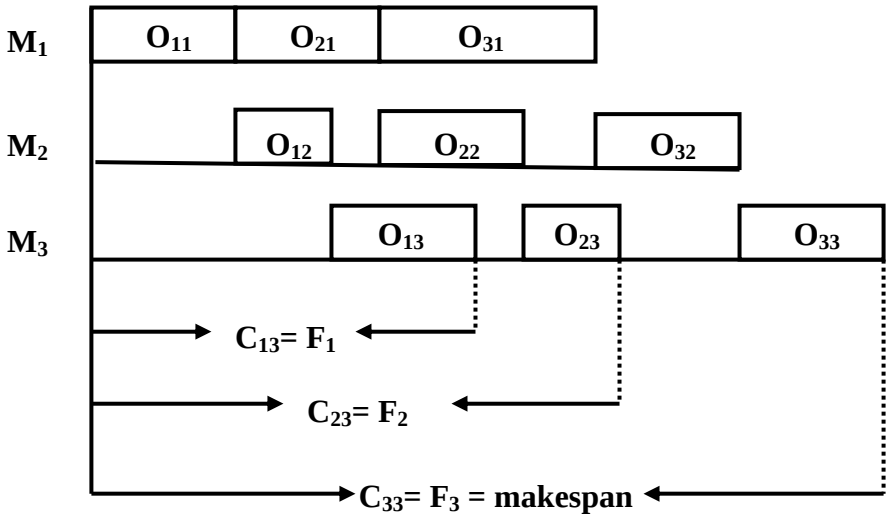
2- التعريف بالمسألة:

على افتراض أن هنـاك مـسألة جدولـة الورشـة الانـسيابية (flow – shop scheduling problem) بثلاث مكائن (machine) (m=3) j=1,2,3 و ثلاثة أعمال (jobs) (n = 3) i=1,2,3 . كل عمل يحتـوي علـى ثـلاث عمليـات operation بحيث أن $J_i = \{o_{i1},o_{i2},o_{i3}\}$ وان كل عملية من هذه العمليات تعالج بماكنة مختلفة عن الأخرى ولا يمكن أن تبدأ أية عملية إلا بعد الانتهاء من معالجة العملية التي قبلها عند نفس العمل , وان جميع الأعمال تعالج بنفس الترتيب عنـد جميع الماكنات (Aggoune and Portmann , 2006).

ليكن (p_{ij}) هو وقت المعالجة للعملية التابعة للعمل i و عند الماكنة (j) و أن (c_{ij}) هو وقت الانتهاء من معالجة العملية التابعة للعمل (i) و عند الماكنة (j) و لهذا فأن (c_{i3}) يعرف على انه وقت الانتهاء من معالجة العمل (i) بصورة كاملة بحيث يغادر النظام و يسمى أيضا بالوقت الانسيابي flow–time للعمل I , و يعرف وقت الانتهاء لأخر عمل بأنه وقت التنفيذ C_{max} (makespan) بحيث أن:

$$C_{\max} = \max_i \{C_{i3}\}$$

و الشكل (1) يوضح مسألة الورشة الانسيابية بثلاثة أعمال و ثـلاث مكـائن مـع ملاحظة أن جميع العمليات للأعمال عولجت بنفس الترتيـب مـن M_1 إلـى M_3 (Deb, 2001).

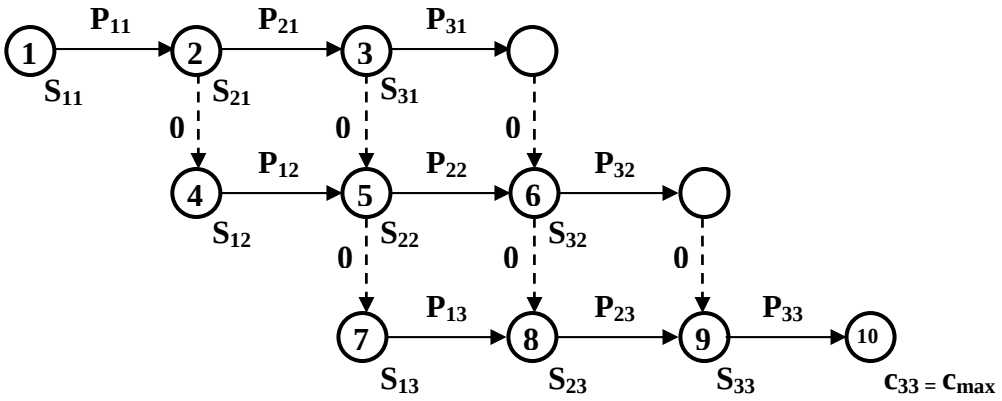


الشكل (1)

يوضح مسالة الورشة الانسيابية بثلاثة اعمال وثلاث مكائن

3- أسلوب البرمجة الخطية:

لصياغة نموذج البرمجة الخطية لمسألة الورشة الانسيابية , تم تحويل المسألة من لوحة گانت إلى بيان حلقي مباشر إذ إن العقد في البيان تمثل وقت البدء و الانتهاء من معالجة العمليات للأعمال و الحافات تمثل أوقات المعالجة لهذه العمليات و البيان أدناه يوضح مسألة الورشة الانسيابية بثلاث مكائن و ثلاثة أعمال(Schniederjans and Hong, 1996):



الشكل (2)

يوضح البيان لورشة انسيابية بثلاث أعمال وثلاث مكائن

وبالاعتماد على البيان أعلاه يمكن صياغة نموذج البرمجة الخطية على النحو الآتي:

اذ ان (⟶) يمثل وقت المعالجة وان (⇢) وقت المعالجة الوهمي و(S_{ij}) هو وقت البدء المعالجة للعملية التابعة للعمل (i) عند الماكنة (j).

1- دالة الهدف :

إن الهدف من الجدولة هو ترتيب الأعمال بحيث إن هذا الترتيب يعطي اقل وقت تنفيذ ممكن و بما إن وقت التنفيذ C_{max} هو حاصل الفرق بين وقت الانتهاء من المعالجة لأخر عملية في الأعمال و وقت البدء بالمعالجة لأول عملية S_{11} , عليه فأن دالة الهدف لهذه المسألة هو تقليل (C_{33}) في الشبكة أعلاه و بهذا تصبح دالة الهدف كالأتي :

$$\min Z = C_{33} - S_{11}$$

2- القيود على دالة الهدف :

كما هو معلوم أن بيئة الورشة الانسيابية لها نظام معين خاص تتبعه في معالجة الأعمال, هو أن العمليات في العمل الواحد تحكمه قيود الأسبقية , أي انه لا يمكن معالجة أية عملية ما لم تتم معالجة العملية التي قبلها, و بما انه لا توجد قيود أسبقية بين العمليات للأعمال المختلفة (Brucker,2007) , عليه يمكن أن تصاغ القيود بالاعتماد على الماكنات الثلاث :-

A – قيود المعالجة عند الماكنة الأولى

بما ان جميع الأعمال متوافرة عند الماكنة الأولى و جاهزة للمعالجة عند الوقت (t=0), وانه لا يوجد قيود الأسبقية بين العمليات الأولى للأعمال الثلاث (o_{31},o_{21},o_{11}), فانه بالإمكان معالجة هذه العمليات الثلاثة من دون وقت شاغر و لضمان عدم وجود وقت شاغر عند الماكنة الأولى تمت صياغة القيدين أدناه:

$$s_{21} - s_{11} = p_{11} \dots (1)$$
$$s_{31} - s_{21} = p_{21} \dots (2)$$

يوضح القيد (1) أن حاصل الفرق بين وقت البدء لمعالجة العملية o_{21} و وقت البدء لمعالجة العملية o_{11} هو p_{11} , هذا يعني أن العملية o_{21} تبدأ بالمعالجة حال انتهـاء o_{11} , وهكذا ضمنا عدم توافر وقت شاغر بين العمليتين o_{11} و o_{21}.وهكذا مع القيد (2) تمت صياغته لضمان عدم توافر وقت شاغر بين العمليتين o_{21} وo_{31} .

B – القيود عند الماكنة الثانية و الثالثة

كما أسلفنا انه توجد قيود أسبقية بين العمليات في العمل الواحد , و لهذا فان الماكنة الثانية من الورشة الانسيابية تحكمها قيود الأسبقية بين العمليتين الأولى والثانية من كل عمل $(o_{i1} \rightarrow o_{i2})$أي أن العملية الثانية لا يمكن معالجتها إلا بعد الانتهاء مـن معالجة الأولى (Brucker,2007) و تكون أوقات المعالجة مختلفة بـين العمليـات جميعاً و لغرض صياغة القيود يمكن الاستفادة من علاقة البدء – البدء بين العمليات والواضحة في العلاقة رقم (3) (Brucker and Knust,2006):

$$s_{ij} + p_{ij} \leq s_{ij+1} \dots (3)$$

اذ تعني هذه العلاقة بأن العملية (j+1) وقت البدء بمعالجتها يجب أن يكون اقل من وقت البدء بمعالجة العملية j مع وقت معالجتها p_j وهذا ينطبق مع شروط الورشـة الانسيابية و يمكن كتابة العلاقة (3) بالشكل الآتي :

$$s_{ij+1} - s_{ij} \geq p_{ij}$$

و بتطبيق هذه العلاقة على العملين الثاني و الثالث تنتج لنا القيود الآتية :

عند العمل الثاني

$$s_{22} - s_{21} \geq p_{21}$$
$$s_{23} - s_{22} \geq p_{22}$$

وعند العمل الثالث

$$s_{32} - s_{31} \geq p_{31}$$
$$s_{33} - s_{32} \geq p_{32}$$

أما فيما يخص العمل الأول فأن قيود الأسبقية تكون بالصيغة الآتية:

$$s_{12} - s_{11} = p_{11}$$

$$s_{13} - s_{12} = p_{12}$$

السبب في ذلك يعود إلى أن جميع الأعمال تعالج بنفس الترتيب و بما انه هـو أول عمل سوف يعالج و عليه فان جميع العمليات التابعة لهذا العمل سوف تعـالج فـي البداية أيضا , أي أن جميع الماكنات سوف تكون متوافرة عند الوقت صفر وبهـذا يكون الفرق مساوياً لوقت المعالجة للعملية التي قبلها.

C – القيود على الماكنات

بما انه لا توجد قيود أسبقية بين العمليات للأعمال المختلفة و هذا ما طبق عند الماكنة الأولى ، لكن هناك فرضية يجب ان تتحقق وهي ان الماكنة لايمكن ان تعالج في نفس الوقت اكثر من عملية واحدة , لذلك يجب وضع قيود بين العمليات للأعمال المختلفة عند الماكنة الثانية و الثالثة لضمان تحقيق هذه الفرضية ، بهذا تكون القيود كالآتي :-

$$s_{22} - s_{12} \geq p_{12}$$
$$s_{32} - s_{22} \geq p_{22}$$
$$s_{23} - s_{13} \geq p_{13}$$
$$s_{33} - s_{23} \geq p_{23}$$

و أخيرا يجب أن يكون الفرق بين وقت الانتهاء لأخر عملية و الذي يمثل وقت الانتهاء و وقت بدء نفس العملية مساوياً إلى وقت المعالجة للعملية نفسها و عليه فان القيد سيكون :

$$c_{33} - s_{33} = p_{33}$$

و بهذا يصبح النموذج الخطي كالأتي, إذ تمت صياغة هذا النموذج لإيجاد وقت التنفيذ لكل احتمال من الاحتمالات الموجودة في ترتيب الأعمال.

$$\min Z = c_{33} - s_{11}$$

s.to

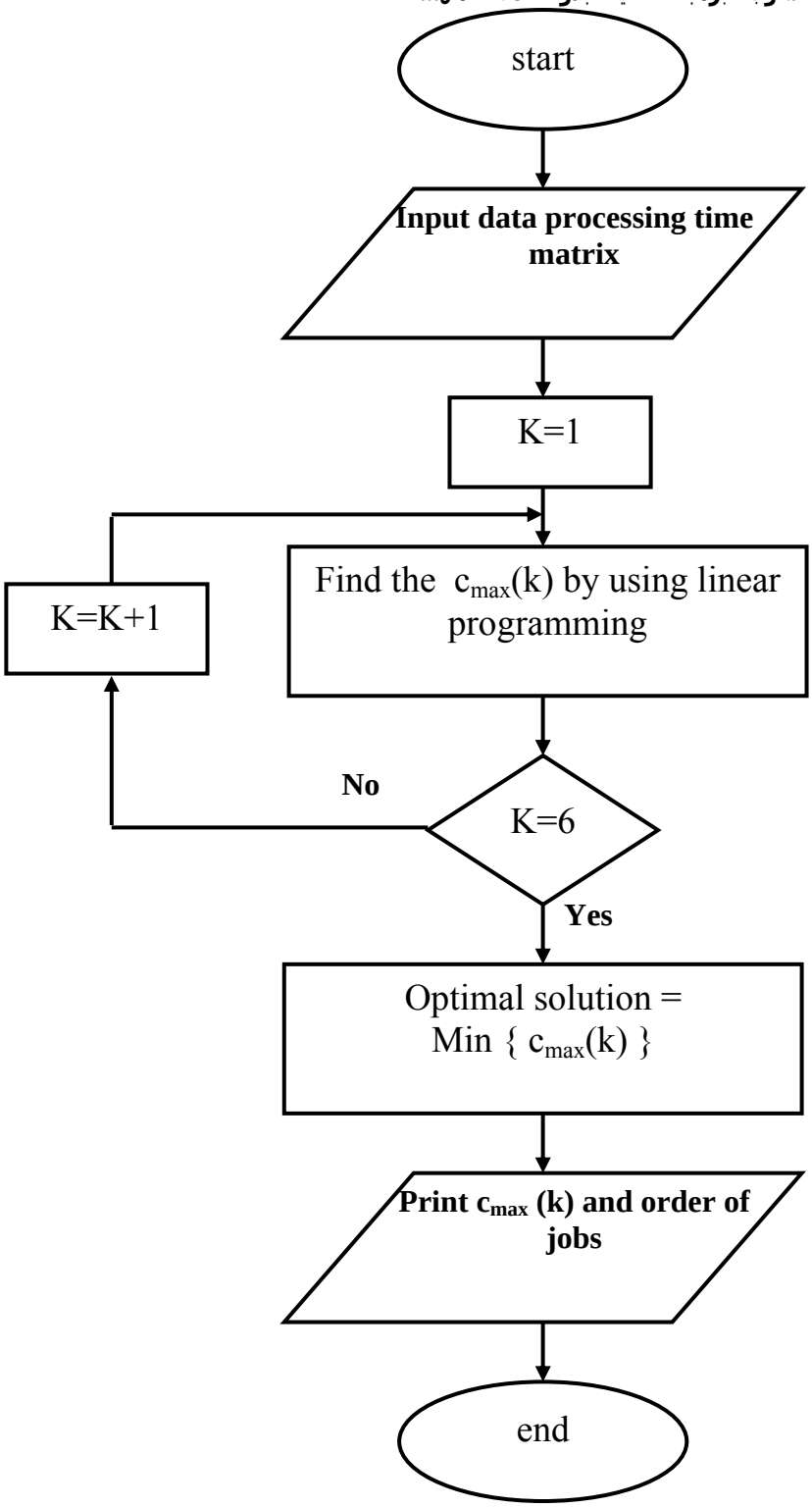
$$s_{21} - s_{11} = p_{11}$$
$$s_{31} - s_{21} = p_{21}$$
$$s_{22} - s_{21} \geq p_{21}$$
$$s_{23} - s_{22} \geq p_{22}$$
$$s_{32} - s_{31} \geq p_{31}$$
$$s_{33} - s_{32} \geq p_{32}$$
$$s_{12} - s_{11} = p_{11}$$
$$s_{13} - s_{12} = p_{12}$$
$$s_{22} - s_{12} \geq p_{12}$$
$$s_{32} - s_{22} \geq p_{22}$$
$$s_{23} - s_{13} \geq p_{13}$$
$$s_{33} - s_{23} \geq p_{23}$$
$$c_{33} - s_{11} = p_{33}$$
$$s_{ij}, c_{33} \geq 0$$

4- الخوارزمية المقترحة:

إن عدد الاحتمالات التي يمكن أن ترتب بها الأعمال الثلاثة هـي (!3) أي (6) احتمالات يمكن أن ترتب بها الأعمال, و لإيجاد أوقات التنفيذ لكل احتمال يجب أن يطبق أسلوب البرمجة الخطية على كل احتمال من الاحتمـالات الـستة و اختيار اقل وقت تنفيذ (makespan) من أوقات التنفيذ الناتجة من كل احتمال. هذا الإجراء يتطلب وقتاً و جهداً كبيراً ولذلك تم اقتراح الخوارزمية الآتية والتي تقوم بحساب أوقات التنفيذ وفق أسلوب البرمجة الخطية المقترح لكل احتمـال من الاحتمالات الست , وبعد ذلك يتم اعتماد الترتيب الذي يعطي اقل وقت تنفيذ , إذ تتلخص هذه الخوارزمية بالخطوات الآتية:

1- البداية
2- إدخال أوقات المعالجة على شكل مصفوفة بحيث أن كل صف يمثل عملاً و كل عمود يمثل عملية.
3- اختيار احد الاحتمالات من احتمالات الترتيب الستة وبدون تكرار.
4- أيجاد وقت التنفيذ $\{c_{max}\}$ بتطبيق أسلوب البرمجة الخطية المقترح وفـق الترتيب المختار في الخطوة (3).
5- إذا كان عدد الاختيارات اقل من (6) يتم الرجوع إلى الخطوة (3) .
6- بعد إيجاد أوقات التنفيذ الست لكل احتمال من الاحتمالات الست يتم اختيار الترتيب الذي نتج منه اقل وقت تنفيذ و بهذا يكون هو الترتيب الأمثل.
7- النهاية.

وفيما يأتي الشكل (3) الذي يمثل المخطط الانسيابي لهذه الخوارزمية



الشكل (3)

يوضح المخطط الانسيابي للخوارزمية المقترحة

5- التجربة :

لغرض معرفة كفاءة الخوارزمية المقترحة تمت مقارنة النتائج مع النتائج التـي أخرجتها خوارزميات (Gupta,1975) و(Johnson ,1954) , إذ تم توليـد (30) مسألة من التوزيع المنتظم , المسائل العشرة الأولى تم توليـدها ضـمن الفترة [1,30] بحيث أن p(i,j) ~ U (1,30) و العشرة الثانية تم توليدها ضمن الفترة [1,60] بحيث أن p(i,j) ~ U (1,60) و العشرة الأخيرة تـم توليـدها ضمن الفترة [1,80] بحيث أن p(i,j) ~ U (1,80) , وان جميع الأعداد التـي تم توليدها أعداد صحيحة و موجبة(Oğuz, Zinder and Janiak , 2004).

تم تطبيق الخوارزمية المقترحة و خـوارزميتي Gupta و Johnson علـى هذه المسائل المولدة و كانت النتائج كما في الجداول الآتية:

الجدول (1)

يوضح نتائج تطبيق الخوارزمية المقترحة مقارنة مع خوارزمياتي Gupta و Johnson ضمن الفترة [1,30]

	Gupta		Johonson		propoused algorithm	
	schedule	makespan	schedule	makespan	schedule	makespan
1	3-2-1	85	3-2-1	85	3-2-1	85
2	1-2-3	94	2-3-1	99	1-2-3	94
3	1-3-2	103	1-3-2	103	3-1-2	102
4	2-1-3	57	1-3-2	64	1-2-3	56
5	2-1-3	84	2-1-3	84	2-1-3	84
6	3-2-1	74	3-1-2	71	2-3-1	70
7	2-3-1	77	2-3-1	77	2-3-1	77
8	3-1-2	77	3-1-2	77	3-1-2	77
9	3-1-2	110	3-1-2	110	3-1-2	110
10	2-1-3	71	1-2-3	66	1-2-3	66

الجدول (2)

يوضح نتائج تطبيق الخوارزمية المقترحة مقارنة مع خوارزميتي Gupta و Johnson ضمن الفترة [1,60]

	Gupta		Johonson		propoused algorithm	
	schedule	makespan	schedule	makespan	schedule	makespan
1	1-3-2	153	3-1-2	153	1-3-2	153
2	1-3-2	193	1-3-2	193	1-3-2	193
3	2-3-1	143	2-3-1	143	3-2-1	138
4	1-2-3	122	1-2-3	122	1-2-3	122
5	2-3-1	151	2-3-1	151	2-3-1	151
6	1-2-3	191	1-2-3	191	1-2-3	191
7	3-2-1	147	3-2-1	147	3-2-1	147
8	1-2-3	190	2-3-1	185	1-3-2	183
9	1-2-3	125	1-2-3	125	1-2-3	125
10	1-2-3	178	1-2-3	178	1-2-3	178

الجدول (3)

يوضح نتائج تطبيق الخوارزمية المقترحة مقارنة مع خوارزميتي Gupta و Johnson ضمن الفترة [1,80]

	Gupta		Johonson		propoused algorithm	
	schedule	makespan	schedule	makespan	schedule	makespan
1	1-2-3	506	1-2-3	517	2-1-3	506
2	1-2-3	389	1-2-3	389	1-2-3	389
3	1-2-3	532	1-2-3	532	1-2-3	532
4	2-1-3	337	2-1-3	378	1-3-2	327
5	2-1-3	561	1-2-3	570	2-1-3	561
6	2-3-1	463	2-3-1	463	2-3-1	463
7	1-3-2	436	1-3-2	436	1-3-2	436
8	3-2-1	432	3-2-1	432	3-2-1	432
9	2-3-1	377	2-3-1	377	2-3-1	377
10	2-3-1	477	2-3-1	477	2-1-3	464

نلاحظ في الجدول (1) أن نسبة التحسين في وقت التنفيذ قياساً مع عدد المسائل المولدة كانت 40% و لكلا الخوارزميتين , إذ قامت الخوارزميـة المقترحـة بإيجاد المتتابعة التي أعطت اقل وقت تنفيذ مقارنةً مع خوارزمية Gupta فـي المسائل رقم (10,6,4,3) , و قياساً مع خوارزمية Johnson كانـت نـسبة التحسين كذلك 40% إذ تم اختيار المتتابعات التي أعطت اقل وقـت تنفيـذ و كانت هذه المسائل هي المسائل (5,4,3,2) .

وهكذا مع باقي الجداول, في الجدول (2) كانت نسبة التحـسين 20% بالنـسبة لخوارزمية Gupta و 20% بالنسبة لخورزمية Johnson وعنـد المـسائل (8,3) , أما في الجدول (3) فكانت نسبة التحسين 20% بالنـسبة لخوارزميـة Gupta وعند المسائل (10,4) و 40% بالنسبة لخوارزمية Johnson و عند المسائل (10,5,4,1).

أما عند باقي المسائل في الجداول الثلاث ,كانت نتائج الخوارزميـة المقترحـة مساوية لنتائج الخوارزميتين الاخرتين.

6- الاستنتاجات

1- الخوارزمية المقترحة كانت أفضل في اختيار المتتابعة المثلى والتي ينتج منهـا اقل وقت تنفيذ makespan مقارنة مع خوارزميتي Gupta و Johnson.
2- كفاءة أسلوب البرمجة الخطية يتناسب تناسباً عكسياً مع حجم المـسالة (عـدد الأعمال) ، أي أنه كلما زاد عدد الأعمال قلت كفاءة أسلوب البرمجة الخطيـة لزيادة عدد القيود مما يؤدي إلى الصعوبة في الحل ، ففي هذا البحـث تمـت صياغة (13) قيداً في حساب وقت المعالجة لثلاثة أعمال.

المصادر

1. Aggoune,Riad and Portmann, M.Claude,(2006),"flow shop scheduling problem with limited machine availability : A heuristic approach ", ***Journal of production economics,*** Vol.99,PP.4-15
2. Bruker, P. (2007), ***"Scheduling algorithms"***, Springer, Germany.
3. Bruker, P. (2006), ***"Complex scheduling"***, Springer, Germany.
4. Deb, Kalyanmoy, (2001***)," Multi-objective optimization using Evolutionary Algorithms "***, John Wiley & sons,Ltd.
5. Etiler, O. Toklu, B. and Wilson, J. (2004), "A genetic algorithm for flow-shop scheduling problems", ***Journal of the Operational Research Society, Vol.55, P.P. 830-835.***
6. Gupta,J.N.D.,(1975),"optimal schedule for special structure flow shops",***Nav.Res.Log.Q. 26, pp.255-269***.
7. Johnson, S.M.,(1954), " Optimal two and three stage production schedule with step time included ," ***Nav.Res. Log.Q.1,PP. 61-68.***
8. Oğuz, C., Zinder, Y., Do, V.H. and Janiak, A. (2004), "Hybrid flow-shop scheduling problems with multi-processor task systems", ***European Journal of Operational Research***, Vol. 152, PP. 115-131.
9. Schniederjans,Marc.J and Hong, Sukki , (1996), "Multi-objective concurrent Engineering :A goal programming Approach", ***IEEE TRANSACTION ON ENGIN-EERING MANAGEMENT ,*** Vol,43,No.2, PP. 202-20***9***.