دراسة حالة اسلوب الامثلية الديناميكي لمشاكل الافق المحدود و غير المحدود لحل مشكلة تخصيص الموارد

أ.د. محمد صادق عبد الرزاق محمد حسن أحمد

كلية الأدارة و الأقتصاد/ جامعة بغداد/ العراق

Case Study Dynamic Optimization Technique for Finite and Infinite Horizon Problems for Solving Resource Allocation Problem

Prof. Dr. Mohammed Sadiq A-AL R. **Mohammed Hasan A.**

College of Administration and Economics

University of Baghdad/Iraq

تاريخ استلام البحث 2021/5/9 تاريخ قبول النشر 2021/6/7

المستخلص:

في هذا البحث تم دراسة مشكلة تخصيص الموارد في شركة الراجي للمشروبات الغازية و العصائر. تهدف الشركة بتخصيص عدد من المكائن لإنجاز عدد من المهام لإنجاز العملية الأنتاجية حيث تنتج عدة أنواع من العصائر و المشروبات الغازية. إذ ترغب بتخصيص 6 مكائن على 4 مهام مختلفة لإنجاز هذه المهام. تكون المكائن المخصصة لكل مهمة معرضة للفشل، و يتم تصليح هذه المكائن لتعود الى العملية الانتاجية مرة أخرى. و من خلال السجلات الخاصة بالشركة فقد سجلت أوقات فشل المكائن و ذلك ليتم حساب احتمالية فشل كل ماكنة و من ثم حساب احتمالية فشل المهام، و أيضا تم تسجيل الوقت الذي تتطلبه كل ماكنة لانجاز كل مهمة. يهدف هذا البحث الى تحديد الحد الأدنى من الوقت المتوقع لإنجاز جميع المكائن المخصصة لأنجاز مهامهم في الشركة و ذلك باستعمال أسلوب الامثلية الديناميكية و عبر آفاق محدودة وغير محدودة. ومن خلال النتائج التي تم الحصول عليها وجد أن مقدار إنجاز المكائن للمهام أصبحت متقاربة فيما بينها، و لكن بالنسبة للمهمة الأولى تستغرق وقتاً أقل من باقي المهام مما يجعلها افضل سياسة يمكن للشركة أن تعتمدها ثم بعد ذلك تكون المهمة الثانية و هكذا بالنسبة للمهمتين الثالثة و الرابعة على التوالي. و أيضا كشف التحليل الدقيق للوضع أنه تم تخفيض المدة التي تستغرقها كل ماكنة لإنجاز مهامها و أن التخطيط المناسب والصيانة السريعة والفعالة يمكن أن تعزز قدرة المكائن وبالتالي تقلل الوقت والجهد و أستمرار عمل المكائن اطول مدة ممكنة، مما يساهم في تقليل تكاليف الشركة وبالتالي تعظيم القدرة الإنتاجية لزيادة أرباح الشركة.

الكلمة المفتاحية: مشكلة تخصيص الموارد، الامثلية الديناميكية، خوارزمية تكرار القيمة، تخصيص المكائن لانجاز المهام.

Abstract:

In this research the problem of resource allocation in Al-Raji Company for soft drinks and juices is studied. The company wants to allocate a number of machines to accomplish a number of tasks to complement the systemic process as it produces several types of juices and soft drinks. If you want to allocate 6 machines, you want 4 different tasks to accomplish these tasks. The machines assigned to each task are subject to failure, and these machines are repaired to return to the production process again. And through the records of the company, the times of failure of the machines were recorded in order to calculate the probability of failure of each machine and then calculate the probability of failure of the tasks, and also the time required for each machine to accomplish each task was recorded. This research aims to determine the minimum expected time to complete all the machines assigned to accomplish their tasks in the company by using the dynamic optimization method and over finite and infinite horizons. And through the results obtained, it was found that the amount of the machines' completion of the tasks became close to each other, but for the first task it takes less time than the rest of the tasks, which makes it the best policy that the company can adopt, then after that the second task and so on for the third and two tasks Fourth in a row. A careful analysis of the situation also revealed that the time it takes for each machine to complete its tasks has been reduced, and that appropriate planning and quick and effective maintenance can enhance the capacity of the machines and thus reduce the time and effort and continue the work of the machines as long as possible, which contributes to reducing the company's costs and thus maximizing the production capacity To increase company profits.

Keywords: Resource Allocation Problem, Dynamic Optimization, Finite and Infinite Horizon, Value Iteration Algorithm, Machines Allocation to Accomplish Tasks.

المقدمة:

هناك العديد من الطرق لحل النماذج الرياضية، والتي تم تصميمها بشكل أساسي لحل المشكلات والوصول إلى الحل الأمثل لأنموذج رياضي معين ويتم استخدامها في العديد من المجالات المختلفة، أحد هذه المجالات هو بحوث العمليات. في هذا القسم، يتم تحديد النهج لبناء أنموذج تخصيص الموارد وحلها باستخدام اسلوب الامثلية الديناميكية، وشرح مشكلتنا. هناك العديد من النماذج في بحوث العمليات مثل (تخصيص الموارد، نظرية الألعاب، إلخ ...)، سيتم في هذا البحث دراسة مشكلة تخصيص الموارد. إذ تعتبر مشكلة التخصيص احد الحالات الخاصة لمشاكل النقل و التي تساعد على أتخاذ القرار. و ايضا تعتبر مشكلة التخصيص أحد أوجه تطبيقات البرمجة الخطية حيث تساعد على أتخاذ القرارات و تكون مشكلة التخصيص أما تعظيم max (للأرباح أو الكفاءة،...) أو تقليل min (للكلف أو الوقت،...)، و تتلخص بوجود عدد معين من الأعمال، الوظائف أو المهام يراد تخصيص عدد من الأشخاص او المكائن لتنفيذ هذه المهام أو الأعمال. و تختلف كل مشكلة عن الأخرى في ما اذا كانت تهدف في تقليل الكلف او الأوقات أو تعظيم الأرباح و زيادة الكفاءة الأنتاجية و لذلك فهي تعتبر حالة خاصة من حالات النقل بين مصادر التجهيز و مناطق الأستعمال والتي يتم حلها باستخدام العديد من التقنيات وأحد هذه التقنيات هو البرمجة الديناميكية. فالبرمجة الديناميكية هي تقنية رياضية لحل المشاكل التي تبحث عن الحل الأمثل للمشاكل متعددة المراحل. قد يعني الحل الأمثل لتعظيم الأرباح وتقليل التكاليف أو الوقت لإكمال المهمة، و التي تقوم بإيجاد القرارات المثلى. تستخدم البرمجة الديناميكية لإنشاء سلسلة من القرارات المترابطة، من خلال تقسيم المشكلة الى عدة مراحل، وفي كل مرحلة يوجد العديد من القرارات ويتم اختيار أفضلها. من أجل حل أي مشكلة تحتاج إلى قرار لاتخاذها، حيث يلعب عامل الوقت الدور الأساسي، ومن ثم يتم اتخاذ القرار على مراحل متعددة، حيث يتم تقسيم المشكلة إلى أجزاء متعددة ويمثل كل جزء مرحلة، إذ يتم اتخاذ القرار في نهاية كل مرحلة. كل قرار يتم اتخاذه يعتمد على المرحلة السابقة ويؤثر على المرحلة التالية التي يتعلق بها. تضمنت هذه الدراسة حل مشكلة تخصيص الموارد باستخدام تقنية التحسين الديناميكي و ذلك باستخدام خوارزمية تكرار القيمة لإيجاد الحل الأمثل. يتم صياغة النموذج ونأخذ في الاعتبار تخصيص

مكائن مختلفة لعدد من المهام المختلفة في شركة إنتاجية. كان لدى شركة الإنتاج عدد محدود من المكائن يراد تخصيصها للمهام في بداية الدراسة. حيث قامت الشركة بتقدير أوقات العمل لكل ماكنة وتوزيعها لإنجاز كل مهمة يتم استلامها في المرحلة الأولى من العملية. كما تم حساب احتمال فشل كل ماكنة مخصصة لكل مهمة. يتم إصلاح المكائن التي تعرضت للعطل للمشاركة في العملية الإنتاجية، حيث سيتم تخصيصها في بداية المدة القادمة. تم صياغة المشكلة على أنها مشكلة أمثلية ديناميكية.

الدراسات السابقة:

يعود الفضل في استخدام البرمجة الديناميكية الى العالم الأميركي (Richard Bellman) و ذلك في عام 1952م. ثم توالت بعدها دراسات كثيرة لحل المشاكل المتعلقة بإتخاذ القرار من خلال استخدام اسلوب البرمجة الديناميكية. في عام 1969م قام الباحثان (G. L. Nemhauser and Z. Ullmann) بدراسة البرمجة الديناميكية المنفصلة و تخصيص رأس المال، و تم من خلالها تطوير بعض الخوارزميات الخاصة بالبرمجة الديناميكية لتخصيص رأس المال الأمثل وفقًا لقيود الميزانية *(Nemhauser & Ullmann, 1969)*. عام 1992م قام الباحثان (Hercules Vladimirou and John M. Mulvy) بدراسة برمجة الشبكة العشوائية لمشاكل التخطيط المالي، و تم استخدام تقنية البرمجة العشوائية في مشاكل تخصيص بعض المدخرات المالية لتصميم المحافظ منخفضة المخاطر*(Valdimirou & Mulvy, 1992)*. في عام 1997م قام كل من (Benjamin Van Roy, Dimitri P. Bersekas, Yuchun Lee and John N. Tsitsiklis) بدراسة البرمجة العصبية الديناميكية لأدارة مخازن بائع التجزئة، ناقشا تطبيق أساليب ألبرمجة الديناميكية لتحسين نظام مخزون المتاجر. إذ تمت مقارنة اداء الحلول الناتجة عن خوارزميات البرمجة الديناميكية العصبية و كانا قادرين على توليد استراتيجيات تحكم متفوقة الى حد كبير و الحد من تكاليف المخزون *(Van Roy & et al., 1997)*. في عام 2002م قام الباحثان (Gregory A. Godfrey and Warren B. Powell) بنشر بحث حول دراسة خوارزمية البرمجة الديناميكية التلاؤمية لإدارة الوقت، قامو بحل المشكلة بأستخدام خوارزمية برمجة ديناميكية تكيفية تستخدم تقريبات دالة غير خطية. و وجدوا أن التقريبات توفر حلولا شبه مثالية للمشاكل و الحلول الطبيعية التي تتفوق بشكل كبير على طرق الأفق المتداول في المشاكل العشوائية *(Godfrey & Powell, 2002)*. في عام 2003م قام الباحثان (D. P. de Farias and B. Van Roy) بدراسة نهج البرمجة الخطية لتقريب البرمجة الديناميكية، قاما بدراسة طريقة فعالة تعتمد على الرمجة الخطية لتقريب الحلول لهذه المشاكل. طوروا من خلال هذه الدراسة حدود الخطأ التي توفر ضمانات الأداء و التي تؤثر على جودة التقريب*(Farias & Van Roy, 2003)*. في عام 2003م قدم الباحثون (Carlos Guestrin, Ronald Parr, Daphne Kollerand and Shobha Venkataraman) بدراسة خوارزميات حل فعالة لعامل MDPs، و من خلال هذه الدراسة تناولو مشكلة التخطيط في ظل

عدم اليقين في عمليات قرار ماركوف الكبيرة (MDPs). قدم الباحثون خوارزميتين من الحلول التقريبية التي تستغل البنية في MDPs المعتمدة. و تتمثل المساهمة الرئيسية لهذه الدراسة في أنها توضح كيف يمكن إجراء العمليات الأساسية لكل من الخوارزميات في عمليات قرار ماركوف MDP *(Guestrin & et al., 2003)*. في عام 2004م درس الباحثان (Michael Z. Spivey and Warren B. Powell) مشكلة التخصيص الديناميكي، و تناولا في هذه الدراسة مشكلة التخصيص الديناميكي الأبسط، و قاما بأقتراح فئة عامة من نماذج التخصيص الديناميكية*(Spivey & Powell, 2004)*. قام الباحون (Warren B. Powell, Abraham George, Belgacem Bouzaiene and Hugo P. Simao) بدراسة البرمجة الديناميكية التقريبية لمشاكل التخصيص عالية الأبعاد و ذلك في عام 2005م، درسوا مشاكل متعددة الأبعاد لحل مشاكل التخصيص المنفصلة و التي يجب حلها بمرور الوقت في ظل عدم اليقين. و تم صياغتها على أنها برمجة ديناميكية *(Powell & et al., 2006)*. عام 2006م أقترح الباحثان (Huseyin Topaloglu and Sumit Kunnumkal) دراسة لتقريب البرمجة الديناميكية لمشكلة تخصيص المخازن في ظل عدم اليقين، تم من خلال هذه الدراسة اقتراح طريقتين ديناميكيتين تقريبيتين للبرمجة الديناميكية لتحسين عمليات التوزيع لشركة تصنيع منتج معين *(Topaloglu & Kunnumkal, 2006)*. في عام 2006م نشر الباحثين (Samuel Venner, Claude Bel−Venner, Alain Pasquet, Franc−ois Iadine Chade` s, Marie− Charpillet and Raymond Leborgne) دراسة تخص مشاكل الأمثلية الديناميكية لأفق زمني غير محدود في طرق بناء شبكة العنكبوت لنسج مدار الشبكة كدراسة حالة، و تم إعطاء وصف شامل للخوارزمية التي تم تنفيذها، من خلال هذه الدراسة تم التوصل الى تنبؤين رئيسين و هما: أولا، تقلل العناكب من حجم ألشبكة ألخاص بها و ذلك لأنها تكسب وزنا بسبب الكلفة المعتمدة على كتلة الجسم لسلوك بناء الويب. ثانيا، هذا الانخفاض في حجم الشبكة يبدأ بوزن أقل في ظل افتراس أعلى مخاطرة *(Venner & et al. 2006)*. في عام 2006م قدم الباحثان (Huseyin Topaloglu and Warren B. Powell) دراسة تقريب البرمجة الديناميكية للمشاكل العشوائية ذات المرحلة−الزمن الصحيحة لتدفق السلع المتعددة، إذ اقترحوا منهجا تكراريا قائما على البرمجة الديناميكية التكرارية يستفيد من التقريبات الخطية او غير الخطية لدالة القيمة *(Topaloglu & Powell, 2006)*. في عام 2012م قام الباحثان (A. M. Ajofoyinbo and Kehinde O. Orolu) بأختبار التخصيص الأمثل لمصدر الراديو في الوصلة الهابطة LTE الخلوية بناءاً على البرمجة الديناميكية المبتورة في ظل عدم اليقين: و أقترحوا تقنية مبنية على البرمجة الديناميكية المبتورة للتخصيص الفعال و الأمثل لمورد الراديو في ظل عدم اليقين و الناتج عن تنقل المستخدمين داخل منطقة التغطية بهدف تقليص الوقت الحسابي و تحديد نظام تعديل

مطبق ديناميكيا إستجابة لحركة المستخدمين *(Ajofoyinbo & Orolu, 2012)*. في عام 2012م قدم الباحثان (Charles I. Nkeki and Chuma R. Nwozo) دراسة مشكلة التحسين الديناميكي لمشاكل تخصيص الموارد في شركات إنتاجية، و تضمنت الدراسة تخصيص المكائن لعدد من المهام بأستعمال التحسين الديناميكي و الهدف من هذه الدراسة هو تعظيم الأرباح التي ستعود للشركة خلال فترة الدراسة*(Nkeki & Nwozo, 2012)*. قام الباحثان في عام 2012م (Milos Hauskrecht and Tomas Singliar) بدراسة حول تحسينات مونت كارلو لمشاكل تخصيص الموارد في أنظمة الشبكة العشوائية، و أظهرت النتائج إمكانيات جيدة لتوسيع نطاق الأساليب المتعلقة بالمشاكل في الشبكات العشوائية الكبيرة *(Hauskrecht & Singliar, 2012)*. في عام 2013م نشر الباحث (Charles I. Nkeki) بحث يخص دراسة تقنية التحسين الديناميكي لتوزيع البضائع مع عجز عشوائي، من المفترض أنه عند شحن هذه البضائع الى متاجر مختلفة، فإن بعضا من هذه البضائع سوف يتضرر مما يؤدي الى عجز في البضائع. من أجل تحديد الحد الأدنى من تكاليف النقل للعملية إذ تم اعتماد مبدأ التحسين الديناميكية *(Nkeki, 2013)*. في عام 2017م قدم الباحثون (Harriso O. Amuji, Geoffrey U. Ugwuanyim, Chukuudi I. Ogbanna, Bridget N. Okechukwu and Hycinth C. Iwu) دراسة حول فائدة البرمجة الديناميكية في تخصيص دورات في الجامعات النيجيرية، حيث تم تحديد افضل سياسة و ذلك بإستعمال أنموذج برمجة ديناميكية لتخصيص دورات المحاضرين في الجامعات النيجيرية *(Amuji & et al., 2017)*. تم ترتيب البحث و تم تفصيله بالاقسام الآتية. القسم (2)، تضمن الدراسات السابقة. و القسم (3)، تضمن وصف المشكلة. القسم (4)، تضمن اعطاء هدف البحث. في حين تم اعطاء مقدمة و تعريف عن البرمجة الديناميكية الاحتمالية في القسم (5). في حين تم بناء الأنموذج في القسم (6). أما القسم (7)، فقد شمل الجانب التطبيقي و أستعراض ما تم التوصل اليه من نتائج. و أخيرا تضمن القسم (8) و القسم (9)، الأستنتاجات و التوصيات على التوالي.

وصف المشكلة:

هناك العديد من الأساليب المستخدمة في مشاكل إتخاذ القرار الخاصة بمشكلة التخصيص. لقد أخذنا أسلوباً واحداً من هذه الأساليب و التي سوف يتم وصفها بأعلى مستوى لتحديد المرحلة. إن الشركة تنتج عدد من المنتجات التي تتطلب إنجاز بعض المهام وتريد تخصيص مكائن لإنجازها في كل مهمة. إن المكائن المخصصة تكون معرضه للعطل. تكون حالات المكائن عشوائية و ذلك لعدم التأكد من عدد المكائن التي سوف تفشل. إن الشركة لابد أن تعرف عدد المكائن التي ستتوفر في المرحلة القادمة وقبل أن تتخذ اي قرار بشأن كيفية تخصيص المكائن التي تعمل لتنجز كل مهمة مخصصة لها. حيث أن عدد المكائن التي سوف تشترك في العملية الانتاجية يعتمد للمرحلة المقبلة على عدد المكائن التي تعطلت في الفترة الحالية.

هدف البحث:

يهدف هذا البحث الى تحديد أدنى حد من الوقت المتوقع لإكمال جميع المكائن المخصصة لإنجاز مهامها في الشركة و ذلك من خلال إستخدام أسلوب الأمثلية الديناميكي و التي تبحث عن الحل الأمثل في كل مرحلة و من ثم تربط جميع الحلول للوصول إلى الحل النهائي للمشكلة عبر H من الآفاق خلال مدة الدراسة.

البرمجة الديناميكة الاحتمالية:

تم إعطاء تعريف مبسط للبرمجة الديناميكية في القسم السابق، و تعرف بأنها أسلوب رياضي يهدف إلى إيجاد أمثل حل لمشكلة معينة من خلال عمل سلسلة مترابطة من القرارات بعد أن تقسم المشكلة إلى أقسام متعددة، يمثل كل جزء كمرحلة وربط هذه الاجزاء مع بعضها من خلال علاقات رياضية، تتضمن كل مرحلة من مراحل المشكلة مجموعة من الحالات. تستخدم البرمجة الديناميكية لحل الكثير من المشاكل سواء كانت هذه محددة أو إحتمالية. و هناك اختلاف بين البرمجة الديناميكية الاحتمالية و البرمجة الديناميكية المحددة في كون أن الحالات في المرحلة التالية تكون غير محددة و غير معروفة عن الحالات في المرحلة الحالية،إذ أن الحالات تكون عشوائية. لذلك فإن هذا النوع من البرمجة الديناميكية أكثر صعوبة وتعقيدًا من النوع المحدد (*Hillier, 2001*).

و يمكن اعتبار أنموذج البرمجة الديناميكية أنموذج احتمالي إذا ما توفر فيها الشروط الآتية:

- إذا كانت العوائد التي لها علاقة بالحالات غير مؤكدة و مجهولة.
- إذا ما كانت الحالات الناتجة عن كل مرحلة من المراحل عشوائية وغير مؤكدة.

وفي حالة توفر هذه الشروط فإن هذا يعني بأنه يمكن تعريف النظام بأنه نظام في ظل عدم اليقين، في هذه الحالة يتم أخذ التوقع لإيجاد الحل الأمثل للأنموذج، إذ أن المعادلة (1) تمثل معادلة الحل للبرمجة الديناميكية و تسمى أيضا بالمعادلة التكرارية أو Bellman Equation.

$$F_h(s_h) = \underset{x_h}{\operatorname{opt}} E\{\Omega_h(S_h, x_h) + F_{h+1}(s_{h+1})\} \qquad (1)$$

حيث أنه:

$\Omega_h(S_h, a_h)$: يمثل العوائد (المساهمات) التي يتم الحصول عليها من خلال الحالة s و الحدث x في المرحلة h.

$F_h(s_h)$: قيمة الدالة للحالة s في المرحلة h.

صياغة معادلة الأمثلية و مشكلة التخصيص:

تتطلب مشكلة التخصيص وجود مزود و مستلم، نقصد بالمزود الجزء الذي يراد تخصيصه الى مستلم أو مصب و و المقصود هنا تخصيص المكائن أو الموظفين أو...الخ، أما المستلم تعني المهام أو الوظائف التي تتطلب تخصيص المصدر أو المزود لها. وقد تم دراسة هذه المشكلة و المتمثلة بتخصيص عدد من

المكائن لإنجاز عدد من المهام. حيث أن المكائن التي سوف تخصص تكون معرضة للفشل. بهذا نتوقع فشل بعض المكائن، مما يؤدي الى عدم التأكد من عدد المكائن التي سوف تفشل.و لذلك فإننا نفترض بأن حالات المكائن عشوائية. الشركة الإنتاجية يجب أن تعرف عدد المكائن التي ستتوفر في المرحلة المقبلة قبل ان تتخذ قرار بشأن تخصيص هذه المكائن للمهام. عدد المكائن المشاركة في العملية الأنتاجية بالمدة المقبلة تعتمد على عدد المكائن التي فشلت في نهاية المدة الحالية. و الهدف من هذه الدراسة هو تحديد أقل الأوقات المتوقعة لأنجاز المكائن للمهام المخصصة لها و ذلك خلال حِقب زمنية معينة. اذا كان الوقت إنجاز المكائن و الناتج من تخصيص m من المهام الى n من المكائن خلال مدة h هو Ω_h^k، و ان حالة المكائن هي S_h، فإن المكائن المخصصة في لعملية الإنتاجية لكل مهمة في كل فترة بموجب السياسة π هو $x_h^{\pi^k}$ و أن المكائن الالتي تعرضت للعطل هو b، أما بالنسبة للعوائد التي سوف تحقق للشركة عبر H من الأفاق هو [*Nwozo & Nkeki*, *2012*].

$$\sum_{h=0}^{H}\sum_{k=1}^{m}\Omega_h^k\left(x_h^{\pi^k}\left(s_{h-1}(b)\right)\right) \qquad (2)$$

و ما يمثل ادنى قيمة متوقعة من الوقت المنجز لمكائن الشركة في السياسة π كما في المعادلة الأتية:–

$$\mathrm{Y}_{\mathrm{h}}^{\pi}(\mathrm{s_h}) = E\left[\min_{x_h\in X(s_h)}\sum_{h=0}^{H}\sum_{k=1}^{m}\beta^h\Omega_h^k\left(x_h^{\pi^k}\left(s_{h-1}(b)\right)\right)\right] \qquad (3)$$

subject to

$$\sum_{k=1}^{m}x_h^{\pi^k}\left(s_{h-1}(b)\right)\geq s_h, h=0,1,2,\dots,H$$

$$x_h^{\pi^k}\left(s_{h-1}(b)\right)\geq 0, h=0,1,\dots,H; k=1,2,\dots,m$$

حيث أن:

β: عامل التخفيض، $0<\beta<1$.

S: مجموعة حالات جميع المكائن.

H: مجموعة الآفاق و التي تمثل مجموعة المراحل في مدة الدراسة.

π: السياسة التي سيتم اختيارها، و تعتمد على الحالة الحالية للنظام.

$x_h^{\pi^k}$: عدد المكائن التي ستخصص للمهمة k في المرحلة h في ظل السياسة π،$h\in[0,H]$.

s_h: عدد المكائن التي تعمل في المرحلة h، $h\in[0,H], s_h\in S$.

b: عدد المكائن العاطلة ضمن H من الآفاق، $h\in[0,H]$.

$x_h^k\left(s_{h-1}(b)\right)$: عدد المكائن المتاحة التي يجب تخصيصها في القادمة.

$X(s_h)$:- يمثل مجموعة الحلول الممكنة للمشكلة (3). بشرط أنه $s_h\in S$.

Ω_h^k: العوائد (المساهمات) المتوقعة من المهمة k في المرحلة h، $h\in[0,H]$.

Π: مجموعة جميع السياسات الممكنة; $\pi\in\Pi$.

s_0: عدد الماكنات في بداية الدراسة.

$Y_h(s_h)$: دالة الهدف، $h \in [0, H]$.

$$\mathrm{Y}_{\mathrm{h}}^{\pi}(\mathrm{s_h}) = \mathrm{E}\left\{\min_{x^{\pi}\epsilon X(s)} \sum_{h'=h}^{H} \sum_{k=1}^{m} \beta^{h} \Omega_{h'}^{k}\left(x_{h'}^{k}\left(s_{h'-1}(b)\right)\right) | s_h \in S_h\right\} \qquad (4)$$

Subject to

$$\sum_{k=1}^{m} x_h^{\pi^k}\left(s_{h-1}(b)\right) \geq s_h, h = 1,2, \dots, H$$

المعادلة (4) تقلل الأوقات المتوقعة على مدى $(X(s_h))$. اذا و من خلال مراكمة العوائد في المرحلة الاولى فسوف يتم الحصول على:

$$\Omega_H(s_H) = \sum_{k=1}^{m} \Omega_H^k(s_H)$$

عندها تصبح المعادلة (4) بالشكل الاتي:

$$Y_h^{\pi}(s_h) = E\left[\left\{\min_{x^{\pi}\epsilon X(s)} \sum_{h'=h}^{H-1} \sum_{k=1}^{m} \beta^{h} \Omega_{h'}^{k}\left(x_{h'}^{k}\left(s_{h'-1}(b)\right)\right)\right\} + \beta^{H} \Omega_H(s_h) | s_h \in S_h\right] \qquad (5)$$

Subject to

$$\sum_{k=1}^{m} x_h^{\pi^k}\left(s_{h-1}(b)\right) \geq s_h, h = 1,2, \dots, H$$
$$x_h^{\pi^k}\left(s_{h-1}(b)\right) \geq 0, h = 0,1, \dots, H; k = 1,2, \dots, m$$

بفرض أن الحالات تكون عشوائية، إذ تمثل s_h كمتغير الحالة في المرحلة h. فأن S تعتبر مساحة الحالة. سوف يتم صياغة هذه المشكلة كمشكلة برمجة ديناميكية، و أن عدد المكائن التي تعرضت للعطل في المهمة k للمرحلة h هو $R_k x_h^k$. حيث R_k يمثل إحتمالية عطل المكائن في المهمة k. و أن عدد المكائن الكلية التي تعطلت و لكل المهام تكون كما في الصيغة الآتية:

$$b = \sum_{k=1}^{m} R_k x_h^k \ , \ h = 1,2, \dots, H$$

إذا فرضنا s_h تمثل عدد المكائن التي سوف تخصص في المرحلة h و أن α تمثل النسبة المؤية للمكائن التي تعرضت للفشل و من المتوقع بأن تشترك في العملية الانتاجية في المرحلة h. يتم حساب s_h من خلال الصيغة الأتية:

$$s_h = s_{h-1} - \left((1-\alpha) * \sum_{k=1}^{m} R_k x_h^k \quad , h = 1,2, \dots, H\right) \qquad (6)$$

α: النسبة المؤية للعطل و لكن تم اصلاحها.

R_k: احتمالية عطل المكائن للمهمة k.

نقوم بتعويض β بدلا من (1-α) فتكون المعادلة بالصيغة:

$$s_h = s_{h-1} - \left(\beta * \sum_{k=1}^{m} R_k x_h^k\right) \ , \ h = 1,2, \dots, H \qquad (7)$$

- اما بالنسبة لحساب احتمالية العطل فسوف نستخدم العلاقة الآتية لحساب إحتمالية فشل كل ماكنة و من ثم نقوم بحساب إحتمالية فشل المهمة، و كما يأتي:

1. من خلال إستخدام العلاقة (8) نقوم بإيجاد إحتمالية الفشل لكل ماكنة (مركبة) .[*Ebeling, 2003*]

$$\hat{R}(i) = 1 - \frac{i}{n+1} = \frac{n+1-i}{n+1} \qquad (8)$$

2. بعد حساب إحتمالية فشل كل ماكنة نقوم بإيجاد أحتمالية الفشل و لكل مهمة [*Ebeling, 2003*] من خلال الصيغة (9).

$$R(s) = \prod_{i=1}^{n} R(i) \qquad (9)$$

- في هذه الحالة، السياسة المثلى يمكن ايجادها وذلك من خلال إيجاد قيمة الدالة عن طريق مشكلة الامثلية (مشكلة التحسين):

$$F_h^{\pi}(s_h) = \min_{x^{\pi} \in X(s_h)} \{ \textstyle\sum_{k=1}^{m} \Omega_h^k \left(x_h^{\pi^k}\left(s_{h-1}(b)\right)\right) + \left(E\{F_h(s_h)\}|S_h = s_h\right) \} \quad (10)$$

Subject to

$$\textstyle\sum_{k=1}^{m} x_h^{\pi^k}\left(s_{h-1}(b)\right) \geq s_h, \quad h = 1,2,\dots,H$$

و بالتساوي

$$\textstyle\sum_{k=1}^{m} x_h^{\pi^k}\left(s_{h-1}(b)\right) = s_h + \xi_h \quad ,h = 1,2,\dots,H$$

$$x_h^{\pi^k}\left(s_{h-1}(b)\right) \geq 0, \quad h = 0,1,\dots,H; \; k = 1,2,\dots,m$$

أن قيمة $\xi_h = 0$ و ذلك لأن جميع المكائن التي تعمل و المتوفرة و يجب أن تخصص في المرحلة المقبلة أقل من عدد المكائن الواجب تخصيصها و ذلك للتقليل من مقدار الوقت المتطلب لإنجاز كل ماكنة و لكل مهمة مخصصة لها. و لجميع $h \in H$.

<u>نظرية 1</u>: بفرض أن s_h متغير حالة و بفرض أن $Y_h(s_{h+1})$ قيمة دالة محسوبة في مرحلة معينة[*Powell, 2004*]. عندها،

$$E\{E(Y_{h'}|S_{h+1})|S_h\} = E\{Y_{h'}|S_h\}$$

بالتالي يصبح لدينا

$$Y_h^{\pi}(s_h) = F_h^{\pi}(s_h)$$

و الذي يثبت صحة الفرضية خلال (h).

لأي دالة هدف معطاة يجب إيجاد السياسات الممكنة π، و التي تحسن هذه الدالة.

$$F_h^*(s_h) = \min_{\pi \in \Pi} F_h^{\pi}(s_h)$$

يتم الحصول على المعادلة أعلاه و ذلك عن طريق حل النموذج (معادلة الأمثلية):

$$F_h(s_h) = \min_{x_h \in X(s_h)} \left[\sum_{k=1}^{m} \Omega_h^k * \left(x_h^k\left(s_{h+1}(b)\right)\right) + E\{F_{h+1}(s_h)\} \right] \qquad (11)$$

و بالتعويض عن المعادلة (7) في المعادلة (11) تكون الصيغة النهائية هي كالأتي:-

$$F_h(s_h) = \min_{x_h \in X(s_h)} \left[\sum_{k=1}^{m} \Omega_h^k * \left(x_h^k\left(s_{h+1}(b)\right)\right) + F_{h+1}\left(s_{h-1} - \beta \sum_{k=1}^{m} R_k x_h^k \right) \right] (12)$$

و يتم حل الأنموذج و ذلك بإستخدام خوارزمية تكرار القيمة و التي سوف يتم توضيحها بالخطوات الآتية[*Venner & et al., 2006*]:

<u>Step 1</u>: Initialization

$h = 0, \forall s \in S, F_0(S) = 0, \quad 0 < \beta < 1, \quad \varepsilon$

<u>Step 2</u>: Iterative process

$h = h + 1,$

For each $s \in S,\ f = F_h(s)$

For each $x \in X$, calculate:

$$F_{h+1}(s_h) = \min_{x_h \in X(s_h)} \left[\sum_{k=1}^{m} \Omega_h^k \left(x_h^k\left(s_h(b)\right)\right) + F_h\left(s_{h-1} - \beta \sum_{k=1}^{m} R_k x_h^k \right) \right]$$

$\pi_h(s) = arg\ min\ F_h(s, x)$

$F_h(s) = F_h\left(s, \pi_h(s)\right)$

<u>Step 3</u>: if $max\ (|F_h(s) - f|) \leq \varepsilon$

$\pi_h(s) = \pi^*$

و تعد خوارزمية تكرار القيمة (Value Iteration Algorithm) إحدى الطرق التي تستخدم لتحديد الإستراتيجية المثلى على أفق زمني محدود وغير محدود. في التكرار الأول لدينا h = 0، إذ تتم تهيئة جميع قيم الحالة إلى 0. بعد ذلك يتم تشغيل التكرارات المتتالية و ذلك عن طريق زيادة التكرار بمقدار خطوة واحدة. في كل حالة (s) ولكل حدث (x) ، يتم حساب قيمة F (s)، (x) كمجموع لجميع القيم المرتبطة بأي حالة لاحقة في المستقبل و المحسوبة في التكرار السابق.

الجانب التطبيقي:

تم تطبيق هذه الدراسة في مجموعة شركات الراجي لإنتاج المشروبات الغازية و العصائر وهي واحدة من الشركات المحلية و المعروفة في العراق تقوم بإنتاج المشروبات الغازية والعصائر ومشروبات الطاقة والمياه الصحية. و قد كان تركيزنا على عملية الإنتاج و كذلك مراقبة المكائن. كان لدى الشركة (6) عدد من المكائن وتريد تخصيص هذه المكائن لإنجاز (4) مهام، و إن هذه المكائن تكون معرضة للفشل. و قدرة الشركة أن 0.95 من آلمكائن العاطلة ستعود الى العمل و تنضم إلى الأجهزة العاملة و في كل مرحلة من المراحل. الهدف هنا بانه و من خلال هذا البحث هو تحديد الحد الأدنى من الوقت المتوقع لإكمال جميع المكائن المخصصة لكل مهمة في الشركة باستخدام أسلوب الامثلية الديناميكي خلال مدة الدراسة H. بعد حساب احتمال الفشل لجميع المهام من المعادلة (8) والمعادلة (9) والتي تدل على R(s) أي بعد تسجيل أوقات التشغيل حتى حصول الفشل نحصل على النتائج في الجدول (1).

جدول رقم (1)

يمثل إحتمالية فشل كل مهمة

R(i) \ Task	1	2	3	4
0	1	1	1	1
1	0.8	0.8	0.8333	0.8
2	0.6	0.6	0.6667	0.6
3	0.4	0.4	0.5	0.4
4	0.2	0.2	0.3333	0.2
5			0.1667	
6				
R(S)	**0.0384**	**0.0384**	**0.0154**	**0.0384**

والجدول رقم (2) يعطي معلومات مفصلة عن أوقات الإنجاز الأولية المتوقعة (بالساعات) للمكائن لإكمال المهام مع احتمال الفشل لكل مهمة. الهدف هو تحديد الحد الأدنى من الوقت المتوقع لإكمال جميع المكائن المخصصة لأداء مهامها لكل منتج في الشركة باستخدام اسلوب الامثلية الديناميكي.

جدول رقم (2)
أوقات الإنجاز الأولية المتوقعة للمكائن بـ(الساعة) وكذلك احتمال تعطل المكائن

Machines \ Tasks	x^1	x^2	x^3	x^4
1	1.0667	1.1429	1.2	1.3714
2	1.0667	1.1429	1.2	1.3714
3	1	1	1	1
4	0.8772	0.8772	0.9524	1.1905
5	1.0033	1.0033	1.0033	1.0033
6	1.2	1.2973	1.2	1.3714
Probability of failure (R_k)	0.0384	0.0384	0.0154	0.0384

إذا كان s_0 هو عدد المكائن في بداية الدراسة، فإن عدد المكائن الكلية المتعطلة لكل مهمة هي $(\sum_{k=1}^{4} R_k x_h^k)$ و بالنسبة للمكائن التي من المتوقع بأنها سوف تشارك في العملية الانتاجية مع المكائن العاملة $(0.95 * \sum_{k=1}^{4} R_k x_h^k)$. و أن عدد المكائن المتوقع بإنها ستخصص في المرحلة القادمة

$$s_h = s_{h-1} - \left((0.05) * \sum_{k=1}^{4} R_k x_h^k \quad , h = 1,2, \dots , H\right) \qquad (13)$$

و التي تمثل المعادلة الحركية للمشكلة. إن الإنموذج الخاص بالمشكلة يمكن كتابته بالصيغة الآتية:

$$Y_h^{\pi}(s_h) = E\left[\min_{x_h \in X(s_h)} \sum_{h=0}^{H} \sum_{k=1}^{4} \beta^h \Omega_h^k \left(x_h^{\pi^k}\left(s_{h-1}(b)\right)\right)\right] \qquad (14)$$

Subject to

$$\sum_{k=1}^{4} x_h^{\pi^k}\left(s_{h-1}(b)\right) = s_{h-1}, \quad h = 0,1,2, \dots , H$$

$$x_h^{\pi^k}\left(s_{h-1}(b)\right) \geq 0, h = 0,1, \dots , H; k = 1,2,3,4$$

القيد الاخير يعني بأنه لا يوجد تخصيص مكائن سالبة، بما أنه المعادلة (14) هي نفسها المعادلة :

$$F_H(s_{h-1}) = \min_{x_h \in X(s_h)} \left[\sum_{k=1}^{4} \Omega_h^k \left(x_h^k\left(s_{h-1}(b)\right)\right) + E\{F_{h-1}(s_h)\}\right] \Rightarrow$$

$$F_H(s_{h-1}) = \min_{x_h \in X(s_{h-1})} \left[\sum_{k=1}^{4} \Omega_h^k \left(x_h^k\left(s_{h-1}(b)\right)\right) + F_{H-1}\{ s_{h-1} - (0.05 \sum_{k=1}^{4} R_k x_h^k)\}\right]$$

Subject to (15)

$$\sum_{k=1}^{4} x_h^k\left(s_{h-1}(b)\right) = s_{h-1}$$

$$x_h^{\pi^k}\left(s_{h-1}(b)\right) \geq 0, \qquad h = 0,1, \dots , H , k = 1,2,3,4$$

لدينا m=4 من المهام و n=6 من المكائن و ان R هي إحتمالية فشل المهمة، و أن عدد المكائن في بداية الدراسة $s_0 = 6$، و كذلك $\beta = 0.05$ هو عامل التخفيض و الذي يمثل نسبة فشل المكائن و

عدم اشتراكها مرة اخرى في العملية الأنتاجية و كذلك $\varepsilon = 0.001$ الذي يمثل حد التوقف أو حد الخطأ، بالإضافة الى H الذي يمثل عدد الآفاق أو المراحل (التكرارات). تم إستخدام برنامج MATLAB لحل الخوارزمية الخاصة بإيجاد الحل الأمثل لمشكلة التخصيص (15)، و من خلال البرنامج فقد تم التوصل الى النتائج المذكورة في الجدول (3). أدناه،

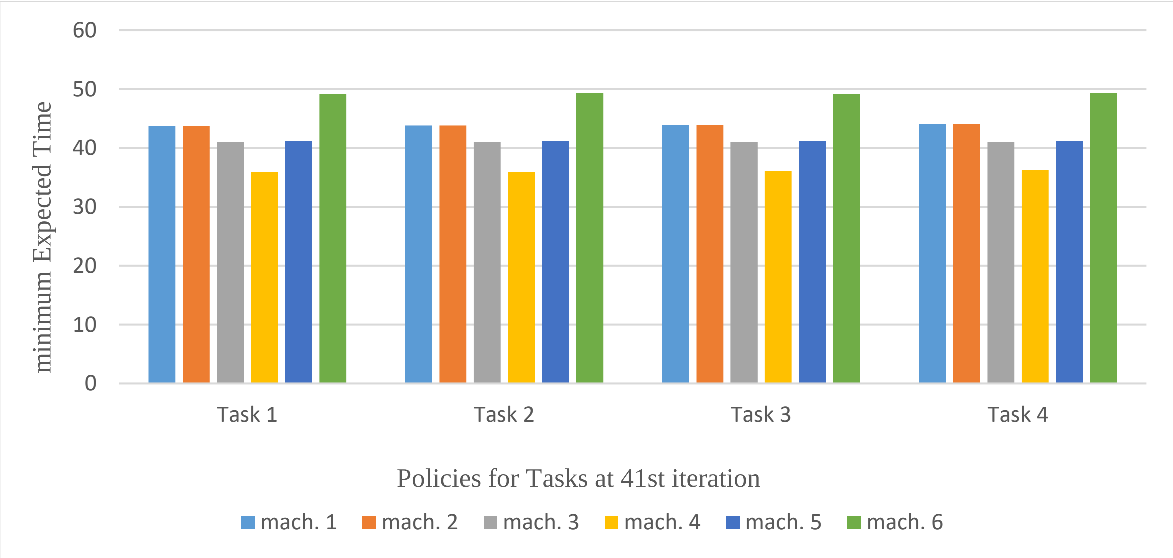
جدول رقم (3)

يمثل النتائج بـ(الساعات) التي تم الحصول عليها و لعدة تكرارات بأستخدام خوارزمية تكرار القيمة و حلها بواسطة برنامج MATLAB.

		$F_h^*(s_h)$			
n.o. iter.	**n.o. mach.**	**Task 1**	**Task 2**	**Task 3**	**Task 4**
41	1	43.7347	43.8109	43.8680	44.0394
	2	43.7347	43.8109	43.8680	44.0394
	3	41	41	41	41
	4	35.9652	35.9652	36.0404	36.2785
	5	41.1353	41.1353	41.1353	41.1353
	6	49.2	49.2973	49.2	49.3714
		$F_h^*(s_h)$			
n.o. iter.	**n.o. mach.**	**Task 1**	**Task 2**	**Task 3**	**Task 4**
135	1	144.0045	144.0807	144.1378	144.3092
	2	144.0045	144.0807	144.1378	144.3092
	3	135	135	135	135
	4	118.4220	118.4220	118.4972	118.7353
	5	135.4455	135.4455	135.4455	135.4455
		$F_h^*(s_h)$			
n.o. iter.	**n.o. mach.**	**Task 1**	**Task 2**	**Task 3**	**Task 4**
300	1	320.0100	320.0862	320.1433	320.3147
	2	320.0100	320.0862	320.1433	320.3147
	3	300	300	300	300
		$F_h^*(s_h)$			
n.o. iter.	**n.o. mach.**	**Task 1**	**Task 2**	**Task 3**	**Task 4**
1150	1	1226.7	1226.8	1226.8	1227
		$F_h^*(s_h)$			
n.o. iter.	**n.o. mach.**	**Task 1**	**Task 2**	**Task 3**	**Task 4**
∞	1	1379.2	1379.3	1379.4	1379.5

لإجل أن يتم حل الأنموذج و الحصول على السياسة المثلى و من خلال التكرارات المتتالية. في بداية الدراسة، فإن جميع قيم الدالة تعطى كقيم أولية لكل الحالات و هذه القيم تكون مساوية للصفر $F_0(S) = 0$. و بعد حل الأنموذج بأستخدام طريقة التقدم الى الأمام و من خلال إعادة العمليات التكرارية

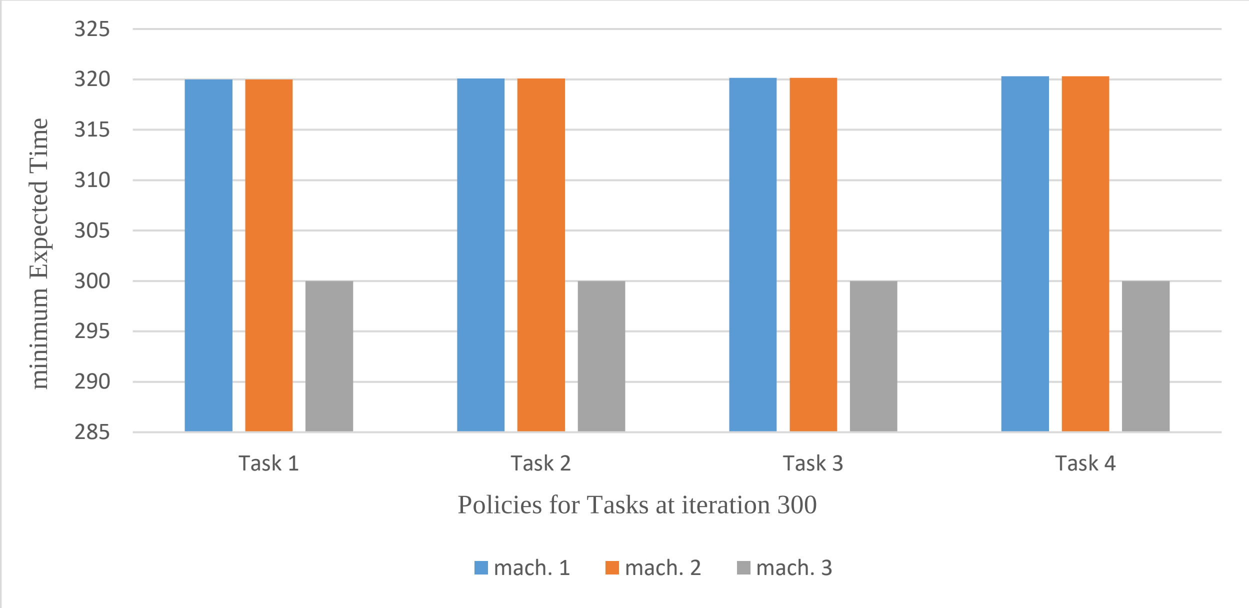
المتتالية فإن الدالة F(S) تكون دالة ذات قيم متزايدة عبر المراحل إذ كلما تقدمنا نحو الأمام بإعادة العمليات التكرارية يتم مراكمة القيم الاولية في كل مرحلة من المراحل. نلاحظ و من خلال النتائج التي تم الحصول عليها بأنه في التكرار 41 فإن الماكنتين الاولى و الثانية للمهمة الاولى أخذت مقدار 43.7 ساعة تقريبا في حين للمهمة الثانية أخذت 43.8 ساعة تقريبا أما بالنسبة للمهمة الثالثة فقد اخذت 43.9 ساعة تقريبا وأخيرا للمهمة الرابعة فقد أخذت ما يقارب 44 ساعة. الماكنة الثالثة و الخامسة فقد كانت أوقات إنجاز المكائن متساوية لجميع المهام، بالنسبة للماكنة الثالثة فقد استغرقت 41 ساعة و لجميع المهامK أما بالنسبة للماكنة الخامسة فقد استغرقت 41.1 ساعة تقريبا لجميع المهام. تساوت مدة إنجاز الماكنة الرابعة للمهام الاولى و الثانية و الثالثة بمقدار 36 ساعة تقريبا في حين أخذت ما يقارب 36.3 ساعة تقريبا للمهمة الرابعة. وأخيرا للماكنة السادسة تساوت مدة إنجاز الماكنة للمهمتين الاولى و الثالثة بمقدار 49 ساعة تقريبا، و أخذت مدة 49.3 ساعة للمهمة الثانية أما بالنسة للمهمة الرابعة فقد أخذت 49.4 ساعة تقريبا.



شكل رقم (1)

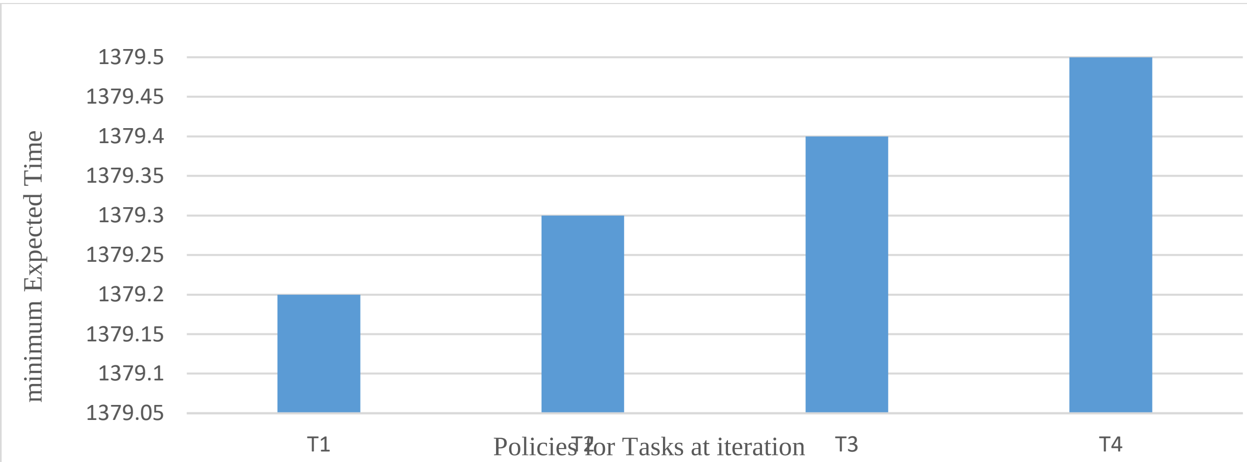
قيم الاوقات المتوقعة و السياسة المثلى التي تم الحصول عليها في اول 41 تكرار

بعد التحرك الى الأمام و بأعادة العمليات التكرارية و في التكرار 135 من الجدول (3) نلاحظ بأن عدد المكائن أصبحت 5 مكائن و في التكرار 300 أصبحت عدد المكائن 3، ذلك لأنه و من خلال المعادلة (4−3) فإن s_h تكون متناقصة بمقدار نسبة الفشل $\beta = 0.05$ و احتمالية الفشل لكل مهمة R(s). بالنسبة للمهمة الاولى من التكرار 300 فإن المكانتين الاولى و الثانية أخذت 320 ساعة تقريبا و الماكنة الثالثة استغرقت 300 ساعة، بالنسبة للمهمة الثانية و الثالثة فإن الماكنتين الاولى و الثانية قد استغرقتا 320.1 ساعة تقريبا و الماكنة الثالثة استغرقت 300 ساعة، في حين أخذت الماكنتين الاولى و الثانية للمهمة الرابعة 320.3 ساعة تقريبا والماكنة الثالثة أخذت 300 ساعة تقريبا.



شكل رقم (2)

قيم الاوقات المتوقعة و السياسة المثلى التي تم الحصول عليها في التكرار 300



شكل رقم (3)

قيم الاوقات المتوقعة والسياسة المثلى التي تم الحصول عليها في حالة مالانهاية من التكرارات

في التكرار 1150 نلاحظ بأن الماكنة الاولى أخذت للمهمة الاولى فقد أخذت 1226.7 ساعة تقريبا، و 1226.8 ساعة تقريبا للمهمتين الثانية و الثالثة، و 1227 ساعة تقريبا بالنسبة للمهمة الرابعة. و في حالة ما لانهاية من التكرارات فقد تم الحصول على الحل الأمثل.إذ أخذت المهمة الاولى 1379.2 ساعة و أستغرقت الثانية مدة إنجاز 1379.3 ساعة تقريبا، بالنسبة للمهمة الثالثة فقد أستغرقت 1379.4 ساعة تقريبا و المهمة الرابعة 1379.5 ساعة تقريبا.

الأستنتاجات:

تناولنا من خلال هذه الدراسة مشكلة تخصيص المكائن لمهام مختلفة ومقارنتها مع بعضها البعض، وذلك لتقليل مدة إنجاز المكائن للمهام الموكلة إليها وعلى مدى أفاق محدودة وغير محدودة. من خلال هذه النتائج التي وجدناها في القسم السابق، نلاحظ بان المهمة الأولى هي أفضل من ناحية سرعة إنجاز مهمتها و أخذت مقدار 1379.2 ساعة تقريبا و لكن بإحتمالية فشل أعلى من المهمة الثالثة و التي تكون أفضل. بالنسبة للماكنة الثانية فقد أخذت 1379.3 ساعة تقريبا و بأحتمالية فشل 0.0384 و المهمة الثالثة أخذت 1379.4 ساعة تقريبا بإحتمالية فشل 0.0154، في حين أن المهمة الرابعة قد أخذت 1379.5 ساعة تقريبا بإحتمالية فشل 0.0384، في الشكل (3) يظهر لنا بأن النتائج التي تم التوصل لها في التكرار المالانهاية و من خلال هذا التوضيح يمكن إختيار أفضل سياسة. نلاحظ بأن أفضل سياسة هي السياسة الخاصة بالمهمة الأولى و التي تستغرق 1379.2 ساعة عبر آفاق غير محدودة و هي الأقل من بين باقي المهام و كما موضحة في الشكل (3).

التوصيات:

محاولة تقليل الأعطال التي تحصل في المكائن من خلال مراقبة المكائن بصورة مستمرة و دورية و كذلك إجراء الصيانة السريعة إذا ما وجد خلل قبل حصول الفشل و إتخاذ اللازم. حيث أنه كلما قلت إحتمالية فشل المكائن كلما زادت مدة عملها وعدم تعرض الماكنة و كذلك المهمة الى التلف. بما أن أفضل سياسة هي السياسة الخاصة بإنجاز المهمة الأولى، فإننا نوصي بتطبيق هذه السياسة لإنجاز جميع المهام الأخرى، لإجل تقليل الوقت و الجهد المبذول من قبل الماكنة لإنجاز مهمتها و في هذه الحالة فإنه يساهم في تقليل التكاليف التي ستتسبب بها للشركة، و من هذا يمكن زيادة القدرة الأنتاجية للماكنة مما يساعد في زيادة ارباح الشركة.

المصادر

أولا: الاجنبية

1. Ajofoyinbo, A. M., & Orolu, K. O. (2012), “Optimal Allocation of Radio Resource in Cellular LTE Downlink Based on Truncated Dynamic Programming under Uncertainty”, Int. J. Communications, Network and System Sciences, 5, p.p. 111-120.
2. AL-Nasser, A. H. (2009), “Reliability of System”, ITHRAA Publishing and distribution, An introduction to Statistical Reliability, Amman, Jordan, p.p. 49-67.
3. Amuji, H. O., Ugwuanyim, G. U., Ogbonna, C. J., Iwu, H. C., & Okechukwu, B. N. (2017), “The Usefulness of Dynamic Programming in Course Allocation

in the Nigerian Universities", Open Journal of Optimization, 6(4), p.p. 176-186.

4. Ebeling, C. E. (2003)," Data Collection and Empirical Methods", University of Dayton, McGraw-Hill, An Introduction to Reliability and Maintainability Engineering, p.p. 283-307.
5. Ebeling, C. E. (2003)," Reliability of Systems", University of Dayton, McGraw-Hill, An Introduction to Reliability and Maintainability Engineering, p.p. 83-107.
6. D. P. de Farias & B. Van Roy. (2003), "The Linear Programming Approach to Approximate Dynamic Programming", Operations Research, 51(6), p.p. 850-865.
7. G. L. Nemhauser & Z. Ullmann. (1969), "Discrete Dynamic Programming and Capital Allocation", Management Science, 15(9), p.p. 494-505.
8. Godfrey, G. A., & Powell, W. B. (2002), "An adaptive dynamic programming algorithm for dynamic fleet management, I: Single period travel times", Transportation Science, 36(1), p.p. 21-39.
9. Guestrin, C., Koller, D., Parr, R., & Venkataraman, S. (2003), "Efficient solution algorithms for factored MDPs", Journal of Artificial Intelligence Research, 19, p.p. 399-468.
10. Hauskrecht, M., & Singliar, T. (2012), "Monte-Carlo optimizations for resource allocation problems in stochastic network systems", In Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence, p.p. 305-312.
11. Hillier, F. S. (2001), "Dynamic Programming", Seventh Edition, McGraw-Hill, Introduction to Operations Research, p.p. 533-575.
12. Mulvey, J. M., & Vladimirou, H. (1992), "Stochastic network programming for financial planning problems", Management Science, 38(11), p.p. 1642-1664.
13. Nwozo, C. R., & Nkeki, C. I. (2012), "On a dynamic optimization technique for resource allocation problems in a production company", American Journal of Operations Research, 2, p.p. 357-363.
14. Nkeki, C. I. (2013), "Dynamic Optimization Technique for Distribution of Goods with Stochastic Shortages. Journal of Optimization", Hindawi Publishing Corporation, Journal of Optimization, Volume 2013, p.p. 1-12.
15. Powell, W. B., George, A., Bouzaiene-Ayari, B., & Simao, H. P. (2005), "Approximate dynamic programming for high dimensional resource allocation problems", In Proceedings In Proceedings, 2005 IEEE International Joint Conference on Neural Networks, Vol. 5, p.p. 2989-2994.

16. Powell, W. B. (2011), Approximate Dynamic Programming: Solving the curses of dimensionality, Princeton University, John Wiley & Sons, Vol. 842.
17. Powell, W. B. (2004), Approximate Dynamic Programming for Asset Management, Princeton University, Princeton.
18. Spivey, M. Z., & Powell, W. B. (2004), "The dynamic assignment problem", Transportation Science, 38(4), p.p. 399-419.
19. Topaloglu, H., & Kunnumkal, S. (2006), "Approximate dynamic programming methods for an inventory allocation problem under uncertainty", Naval Research Logistics (NRL), 53(8), 822-841.
20. Topaloglu, H., & Powell, W. B. (2006), "Dynamic-programming approximations for stochastic time-staged integer multi-commodity flow problems", Informs Journal on Computing, 18(1), p.p. 31-42.
21. Van Roy, B., Bertsekas, D. P., Lee, Y., & Tsitsiklis, J. N. (1997), "A neuro-dynamic programming approach to retailer inventory management", In Proceedings of the 36th IEEE Conference on Decision and Control, Vol. 4, p.p. 4052-4057.
22. Venner, S., Chadès, I., Bel-Venner, M. C., Pasquet, A., Charpillet, F., & Leborgne, R., (2006),"Dynamic optimization over infinite-time horizon: web-building strategy in an orb-weaving spider as a case study", Journal of theoretical biology, 241(4), P.P. 725-733.