المجلة العربية للعلوم الإدارية
مجلد ٨، عدد ٢، مايو ٢٠٠١، ١٨٣ – ١٩٨

بحوث

خوارزمية لحل مسألة التخصيص والجدولة

ياسر عبدالله الموسى
جامعة الملك سعود
المملكة العربية السعودية

مصطلحات علمية

مشكلة النقل، البرمجة الخطية، المخططات الشبكية، الانفجار التراكيبي، البرمجة، غرضية التوجه.

الملخص

يهدف هذا البحث إلى صياغة مسألة التخصيص والجدولة وفق تقنيات تمثيل مسألة تلوين الخريطة الجغرافية. حيث سنمثّل عناصر جدول التخصيص على شكل مخطط شبكي، سنعتمد عليه في بناء شجرة تمكّننا من البحث عن مسار يمر بكافة عقد المخطط ويحقّق شروطاً موضوعة مسبقاً. وسيقدّم البحث برنامجاً يعمل على تمثيل بيانات المسألة وإيجاد الحل لها، وذلك بالاعتماد على التقنيات الأساسية المستخدمة في مجال الذكاء الاصطناعي. أُعِدَّ هذا البرنامج وفق منهجية غرضية التوجه، وتضمَّنت بنية البرنامج قوائم لتمثيل البيانات فضلاً عن أغراض برمجية للمعالجة (برمجة غرضية التوجه). إن الأسلوب المتبع في هذا البرنامج يمثل صيغة بديلة للأساليب الرقمية التقليدية المستخدمة في حل هذه المسألة المستنبطة من البرمجة الرياضية وبحوث العمليات.

اهتمَّ العديد من الباحثين وعبر عقود طويلة من الزمن بدراسة مسألة التخصيص والجدولة ووضع حلول لها، ذلك لما لها من أهمية في تطوير البحث، ولتطبيقاتها العملية المتعددة، واعتمدت أولى الطرق المستخدمة في حل هذه المسألة على نتائج البحوث في مجال بحوث العمليات، حيث نجد في أدبيات حل هذه المسألة طرقاً عديدة تنوعت بتنوع المسائل المطروحة على منهج بحوث العمليات، كما أنها مثلت بأساليب مختلفة تنوعت بحسب الأدوات والطرائق المتاحة لحل المسائل المشابهة، ومن هذه المسائل نذكر:

أ – مسألة النقل:

تعد خوارزميات إيجاد التدفق الأعظمي في مسائل النقل من الخوارزميات المهمّة في بحوث العمليات، وقد استخدمت هذه الخوارزميات في حل العديد من

تم تسلمُّ البحث في فبراير ٢٠٠٠، وأجيز للنشر في أغسطس ٢٠٠١.

المسائل ومنها مسألة الجدولة التي تصاغ بموجب هذا الأسلوب بتكوين عقد يمثل عليها المدرسون p_i وأخرى يمثل عليها الشعب c_i ويتم الوصل بين عقدتين مثل p_i وc_j بالقوس (p_i, c_j) إذا كان المدرس i يدرس الشعبة j. تضاف عقدة المنبع P بحيث تصل إلى كافة العقد p_i والأقواس (P, p_i) وكذلك عقدة المصب C والأقواس (c_j, C). ومن ثم يعرف معامل توجيه (Heuristic) للبحث عن مسار يمثل التدفق الأعظمي بأن يعطى كل قوس أحد القيم 0 و1، بحيث تمثل القيمة 0 عدم وضع المقرر في الجدول و1 وضع المقرر في الجدول. أما الهدف فهو البحث عن تدفق أعظمي ضمن شبكة النقل المكونة من أجل توقيت مقترح، وتكرر عملية الحساب من أجل كل توقيت بشرط عدم وضع قيمة لأقواس المقررات التي سبق تحديد موعدها (Lauriere, 1987). والنموذج التمثيلي لمسألة الجدولة يبدو بهذا الأسلوب سهلاً بعض الشيء؛ إلا أن الصعوبة الرئيسية في تطبيقه تكمن في اختيار تعريف معامل التوجيه خصوصاً فيما يتعلق بصياغة شروط الجدول.

ب – البرمجة الرياضية:

هناك العديد من خوارزميات البرمجة الرياضية التي يمكن استخدامها في حل نموذج تمثيلي لمسألة الجدولة. ويعتمد استخدام هذه الخوارزميات على الكيفية التي يصاغ فيها النموذج، وتبقى المسألة معقدة إلى حد ما نتيجة الشروط العديدة التي يمكن وضعها في النموذج والكيفية التي ستصاغ فيها هذه الشروط. وهذا ما يؤدي إلى تضخيم النموذج وجعل عملية الحل طويلة، فضلاً عن أنه قد لا يكون الحل الناتج محققاً لجميع شروط النموذج، وتحتاج هذه الطريقة أيضاً إلى إعادة صياغة الحل بما يناسب الواقع المدروس، وهذا ما يتطلب العودة إلى متخصص في مجال البرمجة الرياضية أو بناء برنامج يقوم بعملية إعادة صياغة النموذج، ويستدعي الاعتماد على تقنيات خاصة في البرمجة، وقد تصبح عملية صياغة البرنامج عملية أعقد من موضوع الجدولة ذاته.

ج – مسألة تلوين المخططات الشبكية:

تعد مسألة التلوين الأمثل لخريطة جغرافية من المسائل المعروفة حيث عرضت منذ زمن بعيد بوصفها إحدى المسائل المعقدة، وتتلخص هذه المسألة بإيجاد عدد الألوان الأقل التي يمكن استخدامها في تلوين خريطة جغرافية بشرط أن لا يتم تلوين دولتين متجاورتين باللون نفسه. وقد تناول هذه المسألة في الحل العديد من الرياضيين والمهتمين في مجال بحوث العمليات لفترات طويلة قبل أن توضع نظرية مفادها أن هذا العدد هو أربعة ألوان. وفي عام ١٩٧٦ تم البرهان على هذه النظرية بمساعدة الحاسب الآلي (Lauriere, 1987)، واستُخْدِم هذا البرهان في تمثيل مسألة الجدولة وتحديد عدد الأوقات الأقل في تنظيم جدول زمني.

تصاغ مسألة الجدولة بطريقة يمكن من خلالها الاستفادة من طريقة حل مسألة

التلوين بحيث يتم تكوين عقد تمثل عليها الثلاثية (مدرس، قاعة، شعبة) (P_i, H_j, C_l) ويتم وصل كل عقدتين من العقد الناتجة المشتركة فيما بينها إما بالشعبة أو بالقاعة أو بالمدرس، وتحل المسألة بالبحث عن عدد الأوقات الأقل التي يمكن من خلالها تنظيم الجدول ضمن شرط عدم إعطاء وقت واحد لعقدتين مرتبطتين بقوس واحد، والعقد ذات اللون الواحد تمثل المحاضرات التي تلقى في وقت واحد.

يبدو هذا الأسلوب في صياغة المسألة أبسط من الطرق التي سبق ذكرها، إلا أن الحلول المطروحة تعتمد على الأسلوب الخوارزمي في الحل، وتقتصر على تحديد عدد الساعات الأقل لتخصيص محتويات الجدول، ولا يسمح هذا الأسلوب بوضع الشروط التي يجب مراعاتها أثناء تكوين الجدول.

تتصف الخوارزميات المطورة في بحوث العمليات بخصوصيات محددة قد تحتاج إلى صياغة كل خوارزمية بحيث تتلاءم مع الواقع المدروس. كما أن إضافة شروط جديدة إلى المسألة قد يؤدي إلى إعادة النظر في صياغة المسألة وبالتالي إلى تطوير خوارزمية أخرى للحل، وقد تجعل من عملية متابعة تطوير النظام عملية أعقد من موضوع الجدولة ذاته (Boizumault, 1995).

إن مجال تطوير منهجية البرمجة المنطقية قد أخذ منحى خاصاً يميل إلى زيادة القوة في إمكانية تطوير لغات برمجة منطقية واستخدامها، وهناك عدة لغات برمجة منطقية تسمح بتضمين بعض القيود إلى البرنامج بما يسمح تقييد عملية البحث عن حل للمسألة ضمن فضاء خاص يتم تكوينه حسب معطيات المسألة، وأهم هذه اللغات هي "Prolog: Programming in Logic" وبعض اللهجات المشتقة منها. وعلى الرغم من أن هذه اللغات تبدو مناسبة أكثر من الطرق السابقة لإعداد برامج خاصة في تصميم الجداول الزمنية إلا أنها تعتمد على منهجيات قائمة على البرمجة المنطقية الأكثر استخداماً في صياغة برامج الذكاء الاصطناعي، وقد حد من انتشار هذه اللغات ظهور منهجية أخرى وهي البرمجة غرضية التوجه (Object Oriented)، وهذه المنهجية تتمتع بكفاءة وفاعلية أعلى وإمكانات أكبر في مجال صياغة حلول للمسائل التي كانت تعالج باستخدام لغات البرمجة المنطقية (Puget, 1992).

أهداف البحث ومنهجيته

يهدف البحث إلى صياغة مسألة الجدولة الزمنية على شكل مخطط شبكي بالاعتماد على تقنيات مسألة التلوين، ومن ثم تكوين برنامج لتحويل هذا المخطط إلى شجرة، ويتم البحث فيها عن جدول تخصيصات زمنية مناسب وفق شروط موضوعة مسبقاً، وتعميم هذا البرنامج يفيد في حل العديد من المسائل نذكر منها: جدولة عمل ورش الصيانة في أقسام المصانع، وجدولة محاضرات المؤتمرات، وتكوين جداول الدوام والامتحانات في كليات الجامعة والمدارس. سنعتمد على أسلوب القوائم في تمثيل بيانات المسألة وعلى منهجية البرمجة غرضية التوجه في إيجاد حل لها بالبحث عن الجدول الزمني المناسب.

تمثيل المخطط الشبكي

لتكن لدينا المجموعات المنتهية التالية:

أ – المجموعة $P = \{P1, P2, ..., Pn\}$. تمثل عاملاً أول في تكوين الجدول، كأن تكون مجموعة أسماء المدرسين في إحدى كليات الجامعة.

ب – المجموعة $G = \{G1, G2,...., Gm\}$. تمثل عاملاً ثانياً في تكوين الجدول، كأن تكون مجموعة شعب المقررات في إحدى كليات الجامعة.

ج – المجموعة $T = \{T1, T2,...., Tk\}$. تمثل مجموعة الأوقات المتاحة، وقد ترمز هذه المجموعة إلى عدد من الأوقات المتاحة للمدرسين والشعب.

د – مجموعة منتهية مكونة من عناصر الجداء السلمي $O = P \otimes G$، وقد تمثل جدول تعيين تدريس الشعب من قبل المدرسين بحيث تحقق الشروط التالية:

الشرط الأول: من أجل كل عنصر (مدرس) من عناصر المجموعة P هناك مجموعة جزئية من G (الشعب) بحيث تتكون لدينا مجموعة من العقد المشتركة بالعنصر P. أي:

$$\forall Pi \in P \ \exists Gpi \subset G | opi = Pi \otimes Gpi \subset O$$

نسمي مجموعة عقد المخطط الشبكي $O' = \bigcup_P Opi$. سنفترض وجود قوس يربط بين عقدتين تضمنتا إما العنصر P_i أو العنصر G_i.

الشرط الثاني: من أجل كل عنصر P_i من P هناك عنصر واحد على الأقل في هذه المجموعة مثل P_j بحيث يشتركان معاً في عنصر واحد على الأقل من عناصر G، أي:

$$\forall Pi \in P \ \ \exists Pj \subset G | opi \cap Opj \neq \phi$$

يُعَدُّ هذا الشرط ضرورياً لتكوين المخطط الشبكي، ويعني عدم تحققه وجود أكثر من شبكة مستقلة، ويمكن عندها معالجة كل منهما بشكل مستقل عن الأخرى.

هـ – من أجل كل عنصر P_i من P هناك مجموعة جزئية Tpi من T (الأوقات المختارة من قبل المدرس) عدد عناصرها أكبر أو يساوي عدد عقد الشبكة التي تحتوي Pi، أي:

$$\forall Pi \in P \ \ \exists \ Tpi \subset T$$

$$Card(Tpi) \ \geq \ Card(Opi)$$

تلخص مسألة التخصيص بتعيين عنصر من عناصر المجموعة T إلى كل عقدة من عقد المخطط الشبكي بحيث لا يتم تعيين عنصر من T إلى عقدتين مرتبطتين بقوس.

مثال:

إذا كانت لدينا مجموعة العوامل التالية:

مجموعة المدرسين P = {P1,P2,P3}

مجموعة الشعب G = {G1,G2,G3,G4,G5}

مجموعة الأوقات T = {T1,T2,T3,T5,T7,T8}

بحيث:

مجموعة الشعب التي يدرسها المدرس P1

Gp1 = {G1, G2, G3}

مجموعة الشعب التي يدرسها المدرس P2

Gp2 = {G1,G2,G4,G5}

مجموعة الشعب التي يدرسها المدرس P3

Gp3 = {G2, G3, G4}

مجموعة الأوقات التي يختارها المدرس P1

Tp1 = {T1, T2, T3}

مجموعة الأوقات التي يختارها المدرس P2

Tp2 = {T1, T3, T5, T7}

مجموعة الأوقات التي يختارها المدرس P3

Tp3 = {T3, T5, T8}

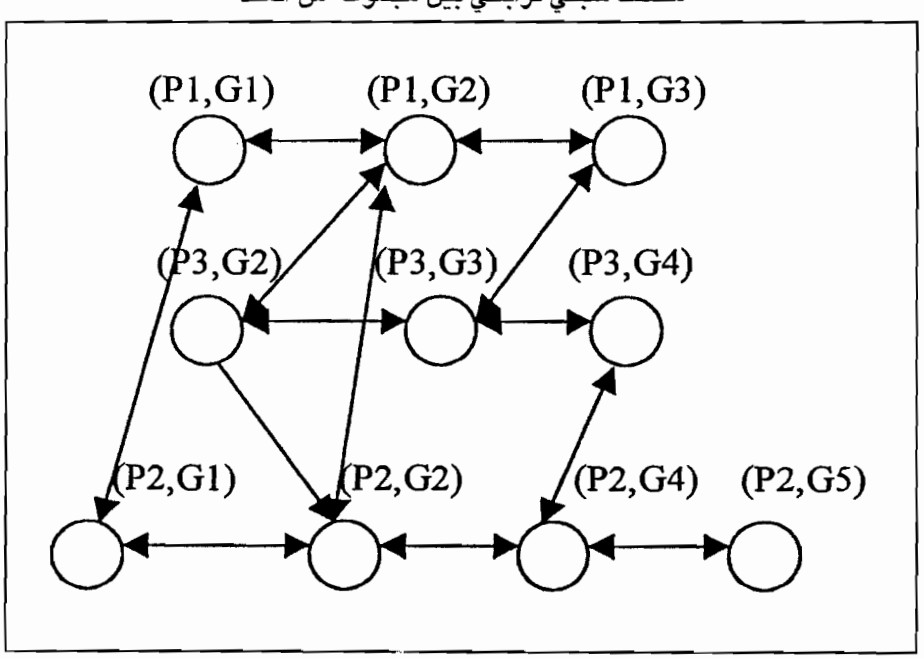
والعقد الممكنة هي:

Op1 = {(p_1,G_1), (P_1,G_2), (P_1,G_3), (P_2,G_1), (P_2,G_2), (P_2,G_4), (P_2,G_5) (P_3,G_2), (P_3,G_3), (P_3,G_4)}

ويصوَّر مخطط الترابط بين العقد على شكل شبكة كما هو موضح بالشكل (١) التالي:

شكل ١

مخطط شبكي ترابطي بين مجموعة من العقد



ويطلب في هذه الحالة تحديد توقيت لكل عقدة من عقد المخطط الشبكي بما يتلاءم مع مجموعة الأوقات الخاصة بكل عنصر من عناصر P وبحيث لا يعطى توقيت واحد لعقدتين مترابطتين من عقد المخطط.

تحويل المخطط الشبكي إلى شجرة

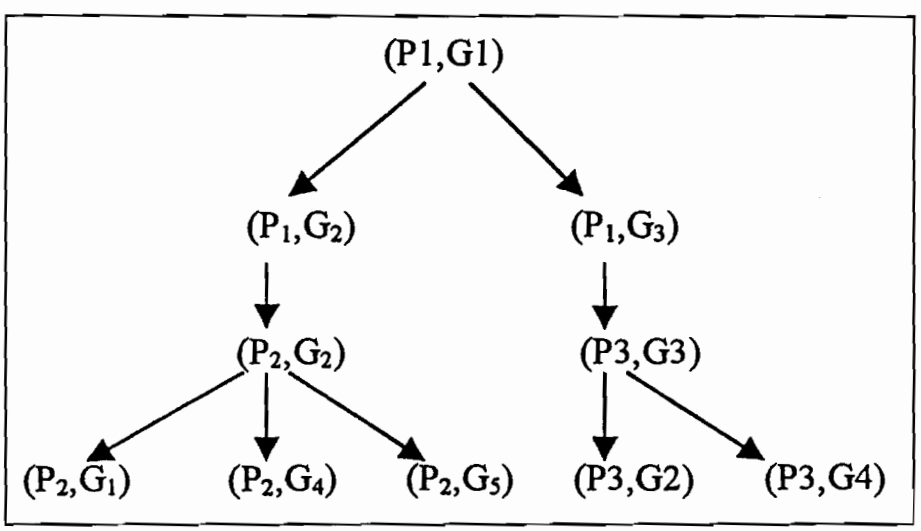
يمكننا تحويل هذا المخطط إلى شجرة تربط بين العقد كافة. وفي الواقع يمكننا تكوين الشجرة بأشكال مختلفة، وذلك بحسب اختيار عقدة البدء والمسارات التي تمر عبر عقد الشبكة كافة. وسنعتمد في تكوين الشجرة على الاستراتيجية التالية:

- البدء من عقدة ما من المخطط الشبكي.
- اختيار كافة عقد المخطط المشتركة مع هذه العقدة بقيمة العنصر P_i.
- اختيار عقد فرعية ترتبط بهذه العقد وتشترك معها بقيمة العنصر G_i.
- البدء باختيار العقد المشتركة معها بقيمة العنصر P_i كما في العقدة الأولى.

وهكذا حتى يتم الانتهاء من كافة عقد المخطط الشبكي. وباستخدامنا لهذه الاستراتيجية نحصل على شجرة كما هي مبينة في الشكل (٢).

شكل ٢

أحد أشكال الشجرة الناتجة عن مخطط شبكي لمجموعة من العقد



نلاحظ أنه مهما تغيرت استراتيجية تكوين الشجرة يظلُّ عدد عقدها مساوياً عدد العقد في المخطط. وبالعودة إلى المخطط الشبكي نلاحظ أن هناك عدة مسارات تؤدي إلى المرور بكافة العقد وبالتالي يمكننا الحصول على أشكال مختلفة من الشجرة بحسب المسار، وهذا يعتمد على عدد عناصر المجموعة G وعدد عناصر المجموعة P ومدى الترابط فيما بينها.

ولإضافة عنصر الوقت إلى عقد الشجرة سنفترض وجود عدة خيارات لتوقيت العنصر P_i أو العنصر G_i، وبالتالي توقيت عقد الشجرة. وإذا أخذنا في الحسبان عدد هذه الخيارات نلاحظ أن عدد طرق تعيين الأوقات يناسب عدد الأوقات المختارة للعنصر الواحد من P. ويمكننا تحديد هذا العدد لكل عنصر P_i بعدد تراتيب الأوقات لهذا العنصر $K = Card(T_{Pi})$ بالنسبة إلى عدد عناصر المجموعة G_{Pi} أي $N = Card(G_{Pi})$ وهو $R_{Pi} = K!/N! = K(K-1)...(K-N+1)$، وإذا أخذنا بعين الاعتبار عدد عناصر المجموعة P وليكن L، فعندها يمكننا تكوين شجرة يصل عدد عقدها إلى $\prod_{Pi} R_{Pi}$. وتتلخص عملية تكوين الجدول الزمني لهذه العناصر بالبحث عن المسار الذي يحقق الرغبات دون حصول أي تعارض.

لتكوين جدول زمني لعقد المخطط في الشكل (١) في حال أخذ خيارات الأوقات لعناصر P بعين النظر نلاحظ أن عقد الشجرة المشتركة ستترابط فيما بينها بقيمة P1 فقط كما في الشكل (٣)، حيث L عقدة وهمية تمثل بداية الشجرة. وسيتفرع عن كل واحدة من العقد عدد آخر من العقد يناسب عدد G و T من أجل كل قيمة P تتشابه فيها قيم G. أي أنه سيحصل ما يسمى بالانفجار التراكيبي (Combinatory Explosion) ويجعل زمن تكوين الشجرة والبحث فيها عن مسار مناسب طويلاً.

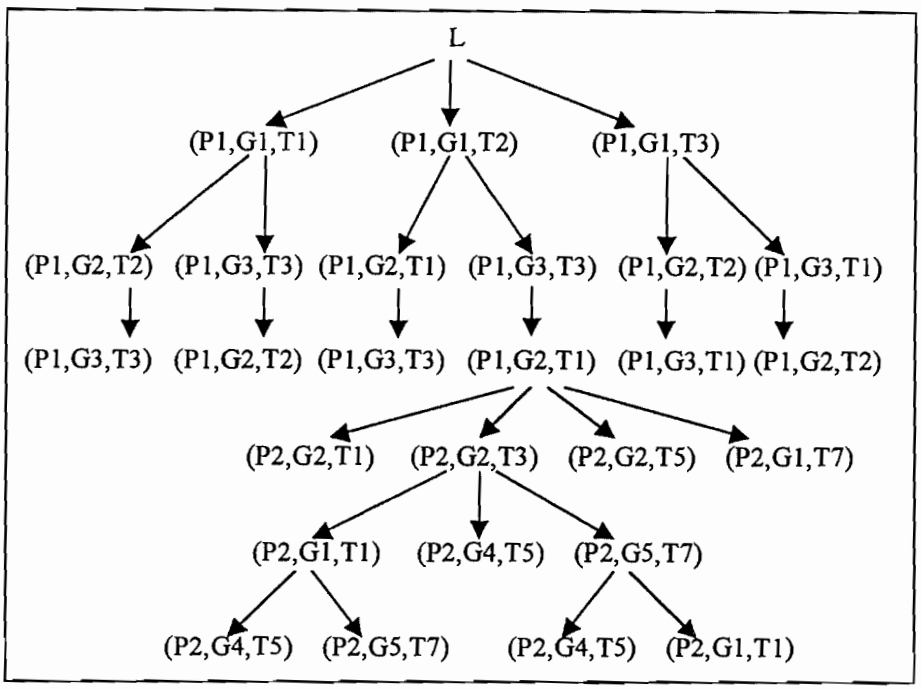
إلا أن عملية تكوين الشجرة بشكل تتابعي، وبتر الأفرع التي تؤدي إلى تعارض ستؤدي إلى اختصار كبير لعدد عقد الشجرة وبالتالي لحجم المسألة المدروسة. فلو استكملنا بناء الشجرة من العقدة (P1, G2, T1) مثلاً، نجد أن من بين العقد التي يمكن أن ترتبط بها العقدة (P2, G2)، ويمكننا بتر فرع الشجرة ابتداء من العقدة (P2,G2,T1) طبقاً لشرط عدم تخصيص عنصر واحد من G مشترك مع عنصرين من P بتوقيت واحد، بينما يمكن استكمال بناء الشجرة من العقد الأخرى.

مراحل تكوين الشجرة

لكي نتمكن من التوقف عند أول مسار يؤدي إلى المرور بكافة عقد المخطط الشبكي ويحقق شروط المسألة نلاحظ أنه من المناسب تكوين الشجرة بالأسلوب الرأسي أولاً، وفيما يلي المراحل الأساسية لبناء هذه الشجرة:

شكل ٣

شكل الشجرة الناتجة عن إدخال عامل الوقت في مخطط شبكي



المرحلة الأولى:

نبدأ ببناء الشجرة بتكوين عقدة وهمية مثل "L" تمثل عقدة الأصل لهذه الشجرة. ثم نقوم باختيار أحد عقد المخطط الشبكي (P_i,G_i) لتتفرع من عقدة الأصل، وبمتابعة تكوين العقد المشتركة مع هذه العقدة بأحد العناصر مثل P_i، وبإضافة عنصر الوقت إلى هذه العقد ضمن شرط عدم تخصيص عنصر واحد من P مشترك مع عنصرين من G بتوقيت واحد يتكون لدينا مسار أول للعنصر P_i. عدد المسارات التي يمكن تكوينها يساوي عدد تراتيب أوقات العنصر P_i بالنسبة إلى عدد عناصر المجموعة G_{Pi} وهو $R_{Pi} = K!/N! = K(K-1)...(K-N+1)$. عدد المسارات للعنصر P_1 التي يمكن تكوينها في مثالنا السابق هو ستة مسارات، والشكل (٣) يبينها.

المرحلة الثانية:

يستكمل بناء الشجرة من أحد المسارات، وذلك بالانتقال إلى تكوين العقد التي تشترك فيما بينها بعنصر آخر من P لم يسبق تكوين عقد تتضمنه. ويتم تحديد هذا العنصر بالعنصر من P للعقدة التي تشترك مع آخر عقدة في المسار المختار بأحد عناصر G. يرتبط وجود مثل هذا العنصر بالشرط الثاني من الشروط التي يفترض أن تحققها المجموعة $O = P \otimes G$.

المرحلة الثالثة:

تتم عملية متابعة تكوين عقد الشجرة ضمن شروط المسألة باتباع الخطوات السابقة مع تحقيق ما يلي:

- استبعاد تفريع العقد التي سبق تحديد وقت لها وذلك بالانتقال إلى تطوير مسار بديل.
- استبعاد تفريع العقد التي يحصل فيها تعارض مع الأوقات التي سبق تعيينها إلى عقد في المستوى الأعلى والانتقال إلى تطوير مسار بديل.
- العودة إلى عقدة أعلى في حال عدم وجود مسار بديل يرتبط بالعقدة قيد التفريع.

المرحلة الرابعة:

وهي مرحلة الحكم بالنجاح في تكوين الجدول إذا تم تعيين وقت لكل عنصر من عناصر العاملين الأول والثاني (المدرسين والشعب)، أو مرحلة الحكم بالفشل إذا تمت العودة إلى عقدة الأصل وعدم إيجاد مسار جديد بديل ومن ثم الكشف عن التعارض.

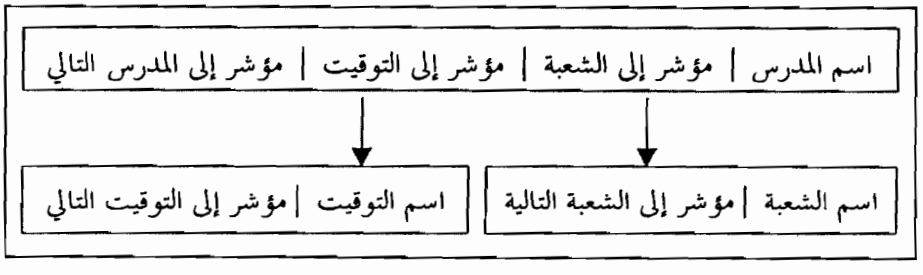
تمثيل البيانات

سنستخدم القوائم في تمثيل معطيات المسألة، وهي تتمثل بأسماء عناصر العوامل والأوقات المختارة وسنمثل هذه البيانات بلغة Visual C++ على الشكل التالي:

```
struct Assignement
{
    char Name[50];
    struct Assignement *NextAssignement;
};
struct Combinaison
{
    char NameGroupe[50];
    char NameTime[50];
    Assignement *Time;
    struct Combinaison *NextCombinaison;
};
struct Professor
{
    char Name[50];
    int Number_Groupes,Number_Times;
    Combinaison *Combinaisons;
    Assignement *Groupes, *Times;
    struct Professor *NextProfessor;
};
```

شكل ٤

الصيغة العامة لتمثيل معطيات مسألة التخصيص



تمثيل المخطط الشبكي النهائي

تُمثل كل عقدة من عقد المخطط الشبكي النهائي بالشكل التالي:

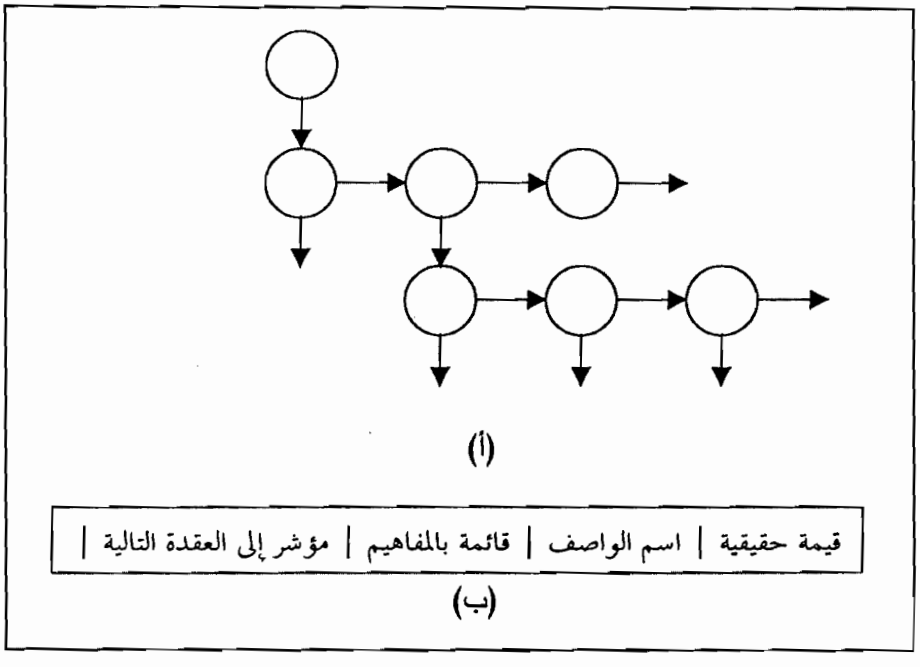
```
struct LPlan
{
    char Prof[50],Groupe[50],Time[50];
    Professor *LProf;
    struct LPlan *NextNode, *FrereNode;
};
```

حيث:

- Prof: اسم العنصر من العامل الأول (المدرس).
- Groupe: اسم العنصر من العامل الثاني (الشعب).
- Time: اسم التوقيت.
- LProf: قائمة عناصر العامل الأول الذين لم يتم تخصيص الأوقات لهم.
- NextNode: مؤشر للعقدة الفرعية.
- FrereNode: مؤشر للعقدة الموازية.

يمثل المخطط الشبكي داخلياً كما في الشكل (٥) حيث إن (أ) يبين الترابط الداخلي للعقد ويبين (ب) الشكل العام لبيانات كل عقدة.

الشكل ٥
التمثيل الداخلي لبيانات عقد الشبكة وترابطها الداخلي



آلية عمل البرنامج

يعمل البرنامج بمنهجية البرمجة غرضية التوجه. ويتضمن عدداً من الأغراض الأولية المترابطة والخاصة بقراءة بيانات المسألة وتنظيمها بالأسلوب المناسب، ومن بين هذه الأغراض: الغرض Professors الخاص بقراءة أسماء المدرسين وعددهم وخياراتهم وغير ذلك من البيانات، ويعتمد هذا الغرض في قراءة الخيارات من حيث الأوقات والشعب على غرض آخر هو Assignements.

الغرض الرئيسي المستخدم في تكوين الشجرة النهائية وإخراجها بالشكل المناسب هو:

```
class LPlans
{
    Professors List_Prof;
    Assignements List_Groupe;
Public:
    int NumbP,i;
    bool TestPlan,B,AddN;
    LPlan *List, *List_Corse, *Current_Corse;
    LPlans();
    FormePlan(LPlan *X);
    NewPlan(LPlan *X);
    bool Contraint1(LPlan *L,LPlan *X);
    EnregPlan(LPlan *X);
    CopyPlan(LPlan *X,LPlan *Y);
    LPlan *FormeNode(Professor *Prof);
    Professor *TestNode(LPlan *L);
    LPlan *Predessesor(LPlan *L,LPlan
        *X, LPlan *Pr);
    bool Add_NextNode(LPlan *Node);
    Add_FrereNode(LPlan *Node);
    int NumPlan(LPlan *X,int K);
    PrintPlan(LPlan *Node);
};
```

أهم الإجراءات التي يحتويها هذا الغرض هي:

- LPlans(): الإجراء الأولي "Const-ructor" ويقوم بقراءة البيانات التي سيعمل عليها البرنامج وتكوين العقدة الرئيسة في المخطط الشبكي.
- FormePlan(LPlan *X): وهو الإجراء الأساسي في تكوين عقد الشجرة، ويستدعي إجراء آخر لتكوين مخطط جديد بعد تغيير الأوقات وهو NewPlan.
- NewPlan(LPlan *X): إجراء يقوم بتكوين مخطط شبكي جديد بعد تغيير الأوقات المختارة، ويعمل بالتناوب مع الإجراء FormePlan بحيث يستدعي كل منهما الآخر حتى يتم التوصل إلى تكوين جدول التخصيصات أو التوصل إلى نتيجة تفيد بعدم إمكانية تكوين هذا الجدول.
- bool Contraint1(LPlan *L,LPlan *X) يختبر الإجراء حالة التعارض بين العقد.
- EnregPlan(LPlan *X): يُسَجِّل المخطط الشبكي المرحلي تمهيداً لاختيار المخطط الأكمل والتوقف عن توليد مخططات أخرى، ويتم نسخ عقد المخطط الأفضل بالإجراء CopyPlan(LPlan *X,LPlan *Y).
- LPlan *FormeNode(Professor *Prof) يقوم بتكوين مجموعة العقدة الجديدة الخاصة بمدرس جديد.

- Professor *TestNode(LPlan *L): اختبار الترابط بين العقدة الحالية والعقدة المؤهلة وتحديد العقدة اللاحقة تمهيداً لتكوين عقدة فرعية أخرى، وفي حال عدم وجود عقدة جديدة مؤهلة سيتم التوقف عن تطوير الشجرة من هذه العقدة ويستكمل البرنامج تطوير الشجرة من عقدة أخرى.
- LPlan *Predessesor (LPlan *L,LPlan *X, LPlan *Pr): تحديد العقدة الأب للعقدة الحالية تمهيداً لحذف فرع الشجرة من هذه العقدة إذا لم يكن صالحاً للتطوير وإعادة تغيير الخيارات والبدء بتطوير الشجرة من عقدة أخرى.
- bool Add_NextNode (LPlan *Node): تكوين عقدة تالية للعقدة الحالية، يستدعي هذا الإجراء الإجراء Add_FrereNode (LPlan *Node) الخاص بتكوين كافة العقد الفرعية من المستوى نفسه. ويعيد هذا الإجراء القيمة خطأ إذا أخفق في تكوين أي من هذه العقد.
- int NumPlan(LPlan *X,int K): يحسب هذا الإجراء عدد المدرسين الذين تم تخصيص الأوقات لهم.
- PrintPlan(LPlan *Node): يقوم هذا الإجراء بطباعة المخطط الناتج إذا كان كاملاً وخالياً من التعارض.

مراحل بناء المخطط الشبكي

يكون المخطط الشبكي بالأسلوب الرأسي أولاً مع العودة إلى الخلف. أي أنَّ تكوين العقد الفرعية لعقدة ما يتم وفقاً للشعب والأوقات التي سبق تحديدها للمدرس وتكون العقد الموازية للعقدة الفرعية وفقاً لأوامر الإجراء FormePlan التالي:

```
LPlans::FormePlan(Lplan *X)
{
    if (NumbP > = List_Prof.Number_
        Professor) return(0);
    if (X- > FrereNode! = NULL) Forme-
        Plan (X- > FrereNode);
    if Add_NesxtNode(X) FormePlan
        (X- > NextNode);
    EnregPlan(List_Corse);
    if (X! = List_Corse) NewPlan(X);
}
```

تتم عملية تكوين الشجرة وفقاً للمراحل التالية:

أ - قراءة معطيات مسألة التخصيص كأسماء المدرسين والشعب والأوقات المختارة.

ب - بناء العقدة الأولى في المخطط ويتم ربط المدرسين كلهم بهذه العقدة على شكل قائمة باسم List_Prof. وتتم هذه العملية من خلال الإجراء الأولي LPlan().

ج - باستدعاء الإجراء FormePlan (LPlan *X) من أجل العقدة الأولى List_Plan يتم تكوين باقي عقد المخطط الشبكي وفق الخطوات التالية:

ج١- اختبار إذا تم تخصيص الشعب والأوقات لكل المدرسين وبالتالي التوقف في حال اكتمال تكوين المخطط. أما إذا لم يكتمل المخطط ستتم متابعة تكوين العقد التالية.

ج٢- إذا كان للعقدة X عقدة موازية ستتابع عملية إعداد المخطط ابتداء من هذه العقدة بإعادة استدعاء إجراء تكوين المخطط FormePlan. وبالتالي سيعاد تنفيذ الإجراء حتى آخر عقدة موازية قبل أن يقوم بتنفيذ المرحلة اللاحقة.

ج٣- اختبار إمكانية تكوين عقدة لاحقة للعقدة الحالية X من خلال الإجراء Add_NextNode(X) حيث يقوم بتكوين العقدة اللاحقة وكافة العقد الموازية واختبار التعارضات مع عقد المخطط التي سبق تكوينها، سيعيد هذا الإجراء القيمة «خطأ» في حال حصول تعارض ويتوقف عن متابعة تكوين العقد التالية لهذه العقدة، أما إذا تم تكوين هذه العقد فسيعاد استدعاء إجراء استكمال تكوين عقد المخطط FormePlan(X) من أجل العقدة التالية.

ج٤- بالانتهاء من هذه المرحلة يستكمل تكوين مخطط بدون تعارضات، ويتم اختبار ما إذا كان المخطط كاملاً أم لا حيث يتم التوقف بموجب الخطوة ج١ في حال اكتمال تعيين الأوقات لكافة المدرسين والشعب.

ج٥- الاحتفاظ بصورة المخطط الناتج باستدعاء الإجراء EnregPlan (List_Corse).

ج٦- البدء بتكوين مخطط جديد باستدعاء الإجراء NewPlan(X) من أجل العقدة X إذا لم تكن X العقدة الأولى في المخطط.

تطبيق:

لتخصيص أوقات المحاضرات لستة مدرسين خلال ست ساعات موزعة على تسع شعب سنعمد إلى تسمية المدرسين والشعب والأوقات كما يلي:

P = {P1,P2,P3,P4,P5,P6}
G = {G1,G2,G3,G4,G5,G6,G7,G8,G9}
T = {T1,T2,T3,T4,T5,T6}

ولتكن الشعب التي سيدرسها كل مدرس:

$G_{p1} = \{G1,G2,G3,G7,G8\}$
$G_{p2} = \{G2,G4,G5\}$
$G_{p3} = \{G2,G3,G5,G6,G7,G8\}$
$G_{p4} = \{G3,G5,G6,G7\}$
$G_{p5} = \{G3,G6,G7,G8,G9\}$
$G_{p6} = \{G5,G6,G7,G8\}$

والأوقات المختارة من قبل كل مدرس هي:

$T_{p1} = \{T1,T2,T3,T4,T5\}$
$T_{p2} = \{T1,T2,T3\}$
$T_{p3} = \{T1,T2,T3,T4,T5,T6\}$
$T_{p4} = \{T1,T2,T3,T4,T5\}$
$T_{p5} = \{T2,T3,T4,T5,T6\}$
$T_{p6} = \{T3,T4,T5,T6\}$

نفذ البرنامج من أجل هذه البيانات على جهاز حاسب آلي من نوع PC Compatible IBM مزود بمعالج من نوع Pentum II Celeron سرعة 300Mhz وذاكرة داخلية RAM بحجم ٦٤ ميغابايت فأعطى الجدول الأول خلال زمن أقل من الثانيتين. وكان الجدول الأول كما يلي:

{P1,G1,T4},{P1,G2,T5},{P1,G3,T2},
{P1,G8,T1}
{P3,G2,T6},{P3,G3,T5},{P3,G5,T4},
{P3,G6,T3},{P3,G7,T1},{P3,G8,T2}
{P5,G3,T6},{P5,G6,T5},{P5,G7,T4},
{P5,G8,T3},{P5,G9,T2}
{P6,G5,T6},{P6,G6,T4},{P6,G7,T3},
{P6,G8,T5}
{P4,G3,T4},{P4,G5,T3},{P4,G6,T1},
{P4,G7,T5}
{P2,G2,T3},{P2,G4,T2},{P2,G5,T1}

خلاصة البحث

قمنا بصياغة مسألة تكوين جدول تخصيص لعاملين أساسيين فقط إضافة إلى عامل الوقت بصورة مخطط شبكي، وقد اعتمدنا في هذه الصياغة على تقنية حل مسألة تلوين خريطة جغرافية. وقمنا بتوصيف خطوات حل المسألة، وكتابة برنامج بلغة Visual C++ يعمل وفق منهجية غرضية التوجه لتمثيل بيانات المسألة وتنفيذ خطوات الحل. وتتلخص هذه الخطوات بإضافة عامل الوقت إلى عقد المخطط الشبكي وتحويله إلى شجرة. وتم البحث في هذه الشجرة عن مسار يسمح بالمرور بكافة عقد المخطط الشبكي محققاً الشروط الإضافية للمسألة.

يبدو هذا الأسلوب أكثر فاعلية مقارنة بالطرق الأخرى التي استخدمت في تكوين نموذج تمثيلي لهذه المسألة، حيث مكننا من تضمين العديد من شروط المسألة، كشرط عدم إعطاء توقيت واحد لعقدتين مشتركتين بعامل واحد وغيرها من الشروط التي قد تحتاج إلى صياغة أكثر تعقيداً وفق تقنيات حل مسائل النقل والبرمجة الرياضية. أما شروط المسألة الأخرى فقد صيغت على شكل إجراءات مستقلة عن الإجراء الأساسي المستخدم في تكوين الشجرة، وبذلك تتاح إمكانية تطبيق هذه الشروط أو عدم تطبيقها، كما يمكن إضافة شروط أخرى للمسألة دون التعرض إلى الإجراء الرئيسي.

قد يبدو الزمن اللازم لحل مسألة التخصيص بهذا الأسلوب طويلاً، وذلك لكثرة المسارات التي يجب اختبارها وما قد ينتج عن ما يسمى بالاستكشاف التركيبي، إلا أنه تم التغلب على هذه المشكلة باعتمادنا على أسلوب بتر فروع الشجرة التي تبدأ بعقد تتعارض مع الشروط الموضوعة مسبقاً. وقد أدى هذا الفعل إلى

إنقاص عدد المسارات الواجب اختبارها وبالتالي الاختصار في زمن الوصول إلى حل. وقد مكننا هذا الأسلوب من تجنب الطرق الرقمية المستخدمة في حل مثل هذه النماذج وما تمثله من صعوبات للوصول إلى حل يحقق شروط المثلوية، علماً بأن البحث عن حل أمثل لمثل هذه المسائل غير مطلوب.

ونشير هنا إلى إمكانية ملاءمة البرنامج وطريقة الحل هذه للعديد من مسائل الجدولة المماثلة التي كانت تحل باعتماد نماذج في بحوث العمليات، ونشير أيضاً إلى أن إضافة عامل جديد إلى الجدول يستدعي تعديل تمثيل البيانات فقط دون التطرق إلى تعديل أسلوب المعالجة. أما شروط المسألة الإضافية فقد تؤدي إلى الإقلال من عدد المسارات الواجب اختبارها، ويمكن إضافة مثل هذه الشروط عبر إجراءات مستقلة.

ولتعميم هذه المنهجية، واستخدامها في حل مسائل متشابهة في التخصيص والجدولة، يجب العمل على تطوير برنامج يتيح للمستخدم إمكانية التحكم بعوامل المسألة وتضمينها الشروط الملائمة بسهولة ويسر، وهو ما نحن بصدده حالياً.

المراجع

Kayser D, 1997. *La Représentation des Connaissances*. Editions_HERMES, France. 320p.

Lauriere, J.P. 1987. *Intelligence Artificielle: Rséolution de Problème par l'homme et la Machine*. EYROLLES, France.

Puget, J.F. 1992. *Programmation par contraintes Orientée Objet, Journées Systèmes Experts d'Avignon*, Avignon France.

Quinlan, J.R. 1996. *Induction of Decision Trees: Machine Learning1*. kluwer Academic Publishers, Boston.

Richalet, J. 1998. *Pratique de l'Identification*. Editions-HERMES 2ed., France.

Boizumault, P., Delon Y. and Guert CH. 1995. *Résolution de Problème en Programmation Logique avec Contraintes*. Revue d'Intelligence Artificielle, 9 (3): 383 - 406.

Burke, E., Elliman D. and Weare, R. 1994. *A Genetic Algorithm for University Time Tabling, in AISB Workshop on Evolutionary Computing*, University of Leeds, UK April 1994.

Carter, M. W. 1986. *A Survey of Practical Application of Examination Time Tabling Algorithms*, Operations Research, 24(2).

Caseau, Y. and Laburthe, F. 1994. *Improved CLP Scheduling with Task Intervals*. 11th Internatinal Conference on Logic Programming.

ABSTRACT

AN ALGORITHM FOR SOLVING THE ASSIGNMENT AND PLANNING PROBLEM

Yasser A. Al-Moussa
King Saoud University

The goal of this paper is to reformulat the problem of assignment and planning to utilize technique used to solve geographic map coloring problem. We use a network graph to represent the elements of the planning table. Therefor a tree is constructed to help searching for a path that goes through all graph nodes and satisfying some predetermined conditions. A Computer program is designed for the representation of the problem's data, and searches for a solution to the problem using the essential technique of the artificial intelligence. This program is written in Object Oriented language, and is consist of lists for the representation of data, as well as classes of objects to handle these data (programming Oriented object). The proposod methodology may be considerent as an alternative method to the classic numeric mathemtical programming method and its operation search.

ياسر عبدالله الموسى (دكتوراه في العلوم الاقتصادية – هندسة النظم الاقتصادية، المعلوماتية والإحصاء – جامعة إيكس مرسيليا ٣، فرنسا، ١٩٩١)، أستاذ مساعد، قسم الأساليب الكمية، كلية العلوم الإدارية، جامعة الملك سعود، له اهتمامات بحثية في مجال المعلوماتية، التنبؤ والنمذجة.