IJEIT
المجلة الدولية للهندسة وتقنية المعلومات
journal homepage:www.ijeit.misuratau.edu.ly
جامعة مصراتة MISURATA UNIVERSITY

اختيار المتغير ذو الكسر الأكبر أو الأصغر كأساس للتفريع في طريقة التفريع والتحديد

د. عبد الله محمد الشيخ
كلية الاقتصاد-جامعة مصراتة، قسم إدارة الأعمال، مصراتة ، ليبيا
a.elshaikh@eps.misuratau.edu.ly

أ. سمية معمر امسلم
كلية الاقتصاد-جامعة مصراتة ، قسم إدارة الأعمال، مصراتة ، ليبيا
s.amsalem@eps.misuratau.edu.ly

م. الصديق ميلاد أبعوه
الشركة العامة للكهرباء ، مصراتة، ليبيا
Sedek.milad@yahoo.com

الملخص— **إن معظم مشاكل البرمجة الخطية الواقعية تحتوي قيم حلها (متغيراتها القرارية Decision Variables) على كسور (Non-integer)، وفي العديد من الحالات الاقتصادية والفيزيائية يكون هذا الأمر غير منطقي، فمثلاً غير مقبول أن يكون الحل هو إنتاج 5.67 طائرة، وتُعد طريقة التفريع والتحديد (Branch and Bound Method) من أهم الطرق المستخدمة في تحديد الحل الأمثل الصحيح (Integer). وتُعنى هذه الورقة بدراسة هذه الطريقة (B&B) وتحديداً دراسة الاختلاف القائم بين البُحاث والمؤلفين في عملية تفريع (تجزئة) المشكلة الرئيسية إلى مشاكل فرعية، وكيفية تحديد المتغير الذي تتم على أساسه عملية التفريع بهدف الوصول للحل الأمثل الصحيح (Integer)، فيرى بعض البحاث أن عملية التفريع تتم على أساس المتغير الذي يحتوي على الكسر الأكبر، والبعض الآخر منهم لا يرى فرقاً في استخدام أي من المتغيرات. والجدير بالذكر هنا هو أن كلا الفريقين لم يذكروا التحليلات المنطقية والأدلة التي تؤيد نظريتهم، وتهدف هذه الورقة لدراسة هذه المشكلة والغوص في آلية عمل هذه الطريقة (B&B) باتباع المنهج التحليلي في تجزئة المشكلة الرئيسية إلى مشاكل فرعية وتتبع مسارات التفريع باستخدام الطريقة البيانية. تبين بأنه يمكن الوصول للحل الأمثل الصحيح باختيار أي من المتغيرين، إلا أنه ليس بالضرورة الوصول للحل الأمثل بنفس عدد عمليات التفريع (المشاكل الفرعية)، فقد يختلف عدد عمليات التفريع باختلاف المتغير المستخدم في عملية التفريع، وتمكنا أيضاً من تقديم التحليلات المنطقية والأدلة التي تؤيد صحة هذه التحليلات.**

الكلمات المفتاحية: **البرمجة، الخطية، التفريع، التحديد، Non-integer.**

1. المقدمة

إن عملية اتخاذ القرار ملازمة للإنسان منذ نشأته حيث كان عليه أن يقرر أين وكيف يعيش، وكانت قراراته في البداية تعتمد على الحدس والتخمين المعتمد على الخبرة والقدرات الشخصية، مع ظهور المنظمات الكبيرة تغيرت طبيعة وحجم ودرجة تعقيد هذه المشاكل، وأصبحت عملية اتخاذ القرارات هي الركن الأساسي في العملية الإدارية، وأضحت هناك ضرورة حتمية لظهور أساليب ونماذج كمية تتلاءم مع طبيعة المشاكل ودرجة تعقدها تعتمد على أسس علمية لاتخاذ هذه القرارات، وكان أول ظهور لهذه الأساليب في القرن الماضي، وعُرفت باسم بحوث العمليات (Operational Research (OR)).

ومن أهم هذه الأساليب وأكثرها شيوعاً هو أسلوب البرمجة الخطية (Linear Programming (LP))، وهو أسلوب يُعنى بصياغة المشاكل الواقعية التي تكون العلاقة بين متغيراتها علاقة خطية ووضعها في شكل نموذج رياضي (دالة هدف وقيود على شكل معادلات خطية)، وظهرت عدة طرق رياضية يمكن استخدامها في حل هذه النماذج، ومن أهم هذه

استلمت الورقة بالكامل في 13 اكتوبر 2021 وروجعت في 9 يناير 2022 وقبلت للنشر في 13 مارس 2022

ونشرت ومتاحة على الشبكة العنكبوتية في 8 يونيو 2022

الطرق طريقة (Simplex Method). والجدير بالذكر هنا هو أن نتائج الحل لهذه النماذج عادةً ما تحتوي على قيم غير صحيحة (كسر أو عدد صحيح وكسر)، بمعنى أن قيم المتغيرات القرارية للمشكلة (Decision Variables)تتضمن كسورًا، وفي العديد من الحالات الاقتصادية والفيزيائية يكون هذا الأمر غير منطقي، فمثلاً من غير المنطقي أن يكون المزيج الإنتاجي الأمثل لمصنع انتاج طائرات (Boeing) هو انتاج (4.7) طائرة من نوع 727 و(8.5) طائرة من نوع 747. وتجدر الإشارة هنا إلى أنه في هذه الحالة لا يمكن استخدام عملية التقريب الاعتيادية، فمثلاً إنتاج (5) طائرات من نوع 727 و (9) طائرات من نوع 747، قد يكون خارج الإمكانيات المتاحة بمعنى أنه حل غير ممكن (Infeasible Solution)، كما أن انتاج (8 ، 5) أو (9 ، 4) قد يكون خارج الإمكانيات المتاحة أيضاً، وإن كان غير ذلك فليس بالضرورة أن يمثل هذا الحل الاستغلال الأمثل للموارد المتاحة، بمعنى أخر أن هذه الحلول قد تكون داخل منطقة الحلول الممكنة(Feasible Region) ، إلا أنها ليس بالضرورة أن تكون الحل الأمثل (Optimal Solution) للمشكلة في حالة برمجة العدد الصحيح(Integer Programming) ، فالحل الأمثل في هذه قد يكون بعيد جداً عن عملية التقريب الاعتيادية، كأن يكون الحل الأمثل مثلاً (11، 1).

توجد أكثر من طريقة يمكن استخدامها لمعالجة هذه المشكلة (عملية التقريب)، ومن أهم هذه الطرق هي طريقة التفريع والتحديد (Branch and Bound Method)، والتي تعتمد على عملية تفريع (Branched) المشكلة الرئيسية إلى عدة مشاكل فرعية، وذلك عن طريق إضافة قيود معينة على المشكلة وتحديداً على أحد القيود الذي يتم احتياره كأساس لعملية التفريع، لتكون حلول المشاكل الفرعية هذه حلول صحيحة لا تحتوي على كسور (Integer)، ثم تتم المفاضلة بينها واختيار أفضلها، ليكون هو الحل الأمثل الصحيح للمشكلة الرئيسية.

ويوجد اختلاف في وجهات النظر بين المؤلفين من حيث تحديد أي من المتغيرات الواجب استخدامها كأساس في عملية التفريع، تهدف هذه الورقة وجهات النظر المختلفة وتحديد وجهة النظر الصحيحة، مع تقديم التحليلات المنطقية التي تؤيد الخيار المناسب في تفريع المشاكل، وتعزيزها بأدلة وبراهين عن طريق أتباع المنهج التحليلي الذي يتناسب مع طبيعة هذه الدراسة، وتوضيح آلية عملها مع تتبعها بالطريقة البيانية بهدف الوصول إلى نتائج جيدة ومرضية تحقق أهداف الدراسة.

2. مشكلة الدراسة

إن الفكرة الأساسية لطريقة الحل في طريقة التفريع والتحديد (B&B) تجزئة المشكلة الأصلية إلى عدة مشاكل فرعية، تبدأ هذه العملية بتجزئة المشكلة الرئيسية إلى مشكلتين فرعيتين عن طريق إضافة قيدين جديدين على أحد المتغيرات، والجدير بالذكر هو أن بعض المراجع المتاحة تشير إلى أن هذه القيود تضاف إلى المتغير الذي يحتوي على الكسر الأكبر، فيتم تكوين المشكلة الفرعية الأولى بإضافة قيد يحمل علامة أصغر أو يساوي (≤) للمتغير (ذو الكسر الأكبر)، وبناء المشكلة الفرعية الثانية بإضافة قيد يحمل علامة أكبر من أو يساوي (≥) لذات المتغير. فعلى سبيل المثال: إذا كان الحل لمشكلة ما ذات متغيرين هو (x_1= 4.29 , x_2= 2.14) فهنا

يتم إضافة قيدين على المتغير ذو الكسر الأكبر وهو المتغير (x_1)، وبالتالي تكون هذه القيود الإضافية كالتالي:

القيــــد الأول: يتم إنشاؤه عن طريق إضافة متباينة تحمل علامة أصغر من أو يســاوي (≤) بحيث يكون طرفها الأيســر المتغير الذي يحمل أكبر كسر(x_1) ، وطرفها الأيمن قيمة العدد الصحيح لذلك المتغير، وبهذا يكون القيد($x_1 \leq 4$) .

القيـــــد الثاني: بنفس الطريقة يتم إنشاء القيد الثاني، إلا أن إشارة هذا القيد (المتباينة) تكون علامة أكبر من أو يســـاوي (≥)، ويكون طرفها الأيسر المتغير (x_1) وطرفها الأيمن قيمة العدد الصحيح لهذا المتغير زائد واحد، وبهذا يكون هذا القيد($x_1 \geq 5$) .

وهنا يرى العبيدي والفضل، في كتابهم بحوث العمليات وتطبيقاتها في إدارة الأعمال، أنه يجب إضافة القيود للمتغير الذي يحتوي على أكبر كسر، دون أن يذكروا التحليل المنطقي والأدلة التي تؤيد ذلك [1]، كما أشار Hamdy Taha، في كتابه Operations Research: An Introduction، أنه ليس بالضروري أن تضاف القيود على المتغير الذي يحتوي على أكبر كسر، ودون أن يذكر أيضاً التحليل المنطقي والأدلة التي تؤيد ذلك [2]، كما أشارWayne L. Winston في كتابه Operations Research بـإن عمليــة الاختيار بين المتغيــرات يمكن أن تكون عشوائية [3]، في حين أن Ghaith Rabadi ,Camille Price , Michael Carter في كتابهم Operations Research: A Practical Introduction لم يشيروا إلى عملية الاختيار، إلا أنهم عند قيامهم باستخدام هذه الطريقة تم اختيار المتغير ذي الكسر الأصغر [4]. وعليه فإنه يمكن تلخيص مشكلة هذه الدراسة في التساؤلات الآتية:

- ما مدى صحة الاختلاف في وجهات نظر المؤلفين حول عملية تحديد المتغير الذي على أساسه تتم عملية تفريع المشكلة الرئيسية إلى مشاكل فرعية؟ وماهي الادلة والتحليلات المنطقية التي تفسر وتوضح عملية تحديد المتغير الذي سيستخدم في عملية التفريع؟
- هل كلا الحالتين (اختيار الكسر الأكبر أو اختيار الأصغر) تصل لذات الحل الأمثل(Integer) ؟ هل تختلف حلول المشاكل الفرعية وعددها باختلاف المتغير المستخدم في عملية التفريع؟
- إذا تساوت عدد المشاكل الفرعية في كلا الحالتين، هل يعني هذا حلول أن حلول المشاكل الفرعية متطابقة في الحالتين؟ ولماذا؟

3. أهمية الدراسة

تكمن أهمية الدراســـة في تســليط الضـــوء على طريقة التفريع والتحديد (B&B)، باعتبارها من أهم الطرق المســتخدمة في إيجاد الحل الأمثل الصــحيح (Integer)، وتوضـــيح آلية عملها عن طريق تتبع خطواتها باســتخدام الطريقة البيانية، وهذا ما يوفر معلومات مهمة ودقيقة حول هذه الطريقة، في ظل الندرة النسبية للمراجع التي تناولت هذا الموضوع .

4. طريقة التفريع والتحديد (B&B)

إن طريقة (B&B) تم اقتراحها عن طريق (Land and Doig) وهي خوارزمية رقمية بسـيطة لحل مشــاكل البرمجة بحيث يمكن لبعض أو كل المتغيرات أن تأخذ قيمًا منفصــلة (Discrete value) فقط [5]، وقد تم مراجعتها وتطبيقها على بعض التطبيقات الأخرى للبرمجة الخطية مثل مشـكلة البائع المتجول (TSP) [6]. وهي تسـتخدم بنجاح في (Optimization) لإيجاد الحلول المثلى (Exact Solutions) لمجموعة واسعة من مشاكل NP-hard problem، وتستخدم في ذلك اســتراتيجية البحث في الشــجرة (كافة الحلول الممكنة) بهدف الوصــول لأفضــل حل (Integer) [7]. الفكرة الأســاسـية لهذه الطريقة هو تجزئة المشكلة الرئيسية في البداية إلي مشكلتين فرعيتين عن طريق إضـافة قيود جديدة إلى النموذج الرياضـي للمشكلة، ومن تم حل كل مشكلة فرعية على حِدَة، ثم تكرار ذات الفكرة على المشــاكل الفرعيـة التي قيم حلها غير صحيحة (تحتوي على كسور)، القيام بعملية التفريع عدة مرات حتى نصل إلى مشاكل فرعيه تكون قيم حلهــا قيم صحيحة أو غيــر ممكنـــة الحل (Infeasible) أصـلاً. وكلا الحالتين هذا يعني عدم إمكانية التفريع، بعد ذلك يتم المفاصــلة بين هذه الحلول ذات القيم الصــحيحة (حلول المشــاكل الفرعية) واختيار الأفضــل، والذي يمثل الحل الأمثل للمشــكلة الأصــلية.

ونظراً لأن هذه الورقة تُعنى بدراســة وتحليل هذه الطريقة(B&B) ، سـيتم تسليط الضوء على هذه الطريقة بشيء من التفصيل.

تجدر الإشــــارة هنا إلى تصـــنيفات مشــــاكل البرمجة الخطية Linear programming problem (LPP) من حيث طبيعة حلولها كالتالي [8]:

- البرمجة الخطية غير الصــحيحة:
  Non-Integer programming problem
  وهي المسألة التي تكون فيها جميع قيم المتغيرات القرارية غير صحيحه.
- البرمجة الخطية الصحيحة المختلطة:
  Mixed Integer programming problem (MIPP)
  وهي المسألة التي تكون فيها بعض متغيرات القرار ذات قيمة صحيحة.
- البرمجة الخطية الصــــحيحة المطلقة:
  Pure Integer programming problem (PIPP)
  وهي المســـألـة التي تكون فيهـا جميع متغيرات القرار ذات قيمـة صحيحة.
- برمجة العدد الصحيح (البرمجة الرقمية المزدوجة):
  Binary Integer Programming
  وهي التي تكون قيم المتغيرات القرارية فيها إما صفر أو واحد صحيح .

إن مشـكلة الدراســة لهذه الورقة تُعنى بالنوع الثالث منها (PIPP) ويوجد عدة طرق تستخدم في حل هذا النوع من المشاكل، من أهمها طريقة التفريع والتحديد (B&B). إلا أن من سلبيات هذه الطريقة هو الاستمرار في عملية التفريع حتى بعد الوصـول للحل الأمثل، لأنه لا يمكن الجزم بصـحة أمثليه الحل إلا بعد حصر كافة الحلول لكل المشاكل الفرعية. يمكن أن يكون عدد التفريعات كبير جداً في حالة تعدد المتغيرات، وهذا يؤدي زيادة حجم الجهد اللازم للوصــــول إلى الحل الأمثل [9]. وتعتبر آلية عمل هذه الطريقة هي صلب موضوع هذه الدراسة .

أ. *الدراسات السابقة* (*Literature Review*).

تسـتخدم طريقة (B&B) ثلاثة عناصـر لضـبط سـلوك الخوارزمية، هذه المكونـات هي: اســتراتيجيــة البحـث (Search Strategy) وتُعنى بالترتيب الذي يتم من خلاله اسـتكشـاف المشـكلات الفرعية في الشـجرة، واستراتيجية التفرع (Branching Strategy) وهي كيفية تقسيم مساحة الحل لإنتاج مشكلات فرعية جديدة في الشجرة، وقواعد التقليم (Pruning Rules) وهي تعني القواعد التي تمنع من استكشاف مناطق دون مستوى الحل الأمثل في الشجرة [7]. والجدير بالذكر هنا هو أن مشكلة هذا البحث تنـدرج تحـت (Search Strategy)، والتي يمكن تصــــنيفهــا إلى: اســـتراتيجيات التفرّع الثنائية (Binary Branching)، وهي تقســـيم المشـكلة الفرعية (S) إلى مشـكلتين صـغيرتين متنافيتين، التفريع الواسـع (Wide Branching) وهو على النقيض من التفرع الثنائي ويركز على تحديد عنصـر واحد من مجموعة خيارات مختلفة، والتفريع الثالث والأخير هو برمجة العدد الصـــحيح (Branching in integer programs) [7]. إن مشـــكلة هذا البحث تنطوي تحت النوع الأخير من التفريع، ولهذا سيتم مرجعة أدبيات الموضوع من هذا الجانب وبشيء من التفصيل. تنقسم اسـتراتيجيات التفرع إلى مرحلتين تحديد متغير للتفريع، وتكوين مشـكلات فرعية بإضافة قيود لإبعادها عن القيم الكســرية [7]. يوجـــــد قاعدة شــائعة وسهــلة الاستخــدام تسمـــى (Most Fractional or Most Infeasible)، وهي تحدد متغير التفريع (y_i) الذي يكون جزئه الكســري هو الأقرب إلى 0.5، كما هو موضـــح من قبل .Achterberg et al [10]، فهذه القاعدة بشكل عام ليست أفضل من الاختيار العشوائي من حيث الوقت الحسـابي المطلوب وعدد المشـكلات الفرعية التي يتم اسـتكشـافها. لـذلـك تم اقتراح عدد من التقنيـات الأخرى كقـاعـدة التفريع المعـاكســـة (Opposite branching Rule)، وهي تختار (y_i) الذي يكون جزئه الكسري الأبعد عن 0.5 [11].

هناك اسـتراتيجية أخرى للاختيار اسـتخدمت لبرمجة الصـحيحة المختلطة (MIP) وهي التفرع الكاذب (Pseudocost Branching) [12]، وذلك بنـاءً على التنبؤ بـالتغير في دالـة الهدف لكل متغير مرشـــح للتفريع، ومن الصـــعوبات التي تواجه هذه الطريقة هو عدم توفر معلومات عن ســـلوك التفرع في بداية الخوارزمية. اقترح .Applegate et al [13] اسـتراتيجية التفرع القوي (Strong Branching) لاختيـار متغير التفريع، حيـث يتم اختيـار المتغير الـذي يؤدي إلى أكبر قـدر من التغيير في دالـة الهدف، وتم تطبيقها في حل مشكلة البائع المتجول (TSP)، وتمكن المؤلفين من حل 20

حالة لم يتم حلها سابقًا، إلا أن الوقت المطلوب لإجراء عملية الاختيار غالبًا ما يكون باهظًا. اقترح أيضاً Linderoth and Savelsbergh [14] جمع فيها بين التفريعين (القوي والكاذب) وسميت بالتفرع الهجين (Hybrid Strong/Pseudocost Branching). استكشف Pryor and Chinneck [15] استخدام قواعد تفريع على أساس تفريع المتغيرات التي تحدث تغيير في أكبر عدد من متغيرات المشكلة، وغالباً ما تؤدي إلى نتائج جيدة. طريقة تفريع حديثة أخرى تم تطويرها بواسطة Fischetti and Monaci [16] تسمى التفرع الخلفي (Backdoor Branching) تعمل هذه التقنية على تحديد مجموعة صغيرة من المتغيرات التي يجب أن تتفرع منها قبل غيرها وتنتج شجرة بحث صغيرة. استخدم Gilpin and Sandholm [17] نتائج المعلومات النظرية لتوجيه عملية البحث لإزالة عدم اليقين من المشكلات الفرعية في شجرة البحث، للقيام بذلك يتعاملون مع قيم المتغيرات الكسرية كاحتمالات، ويحسبون مقدار عدم اليقين ويكون المتغير أقلها يقيناً هو المرشح للاختيار. مؤخراً اقترح Munapo نهجًا جديدًا لتقليل التعقيدات في خوارزمية (B&B) لحل مشكلة (knapsack linear integer)، ففي طريقة (B&B) إذا كان الحل الأمثل للبرمجة الخطية (LP) عددًا صحيحًا، فسيكون الحل الأمثل لمشكلة العدد الصحيح متاحًا، وإذا لم يكن كذلك فيتم تحديد متغير ذو قيمة كسرية لإنشاء مشكلتين فرعيتين، بحيث يتم تجاهل هذا الجزء من المنطقة الممكنة (Feasible Region)، تتكرر العملية على جميع المتغيرات ذات القيم الكسرية حتى يتم إيجاد الحل الصحيح، وفي نهجه هذا يتم إنشاء مجموع متغير وقيود إضافية وإضافتها إلى المشكلة الأصلية قبل البدء في حلها أصلاً [18]. من أجل تحسين سرعة خوارزميات (B&B)، مؤخرًا تم إدخال تقنيات التعلم (Learning) [19]، وهي تعني إدخال الذكاء الاصطناعي في حل هذه الخوارزميات.
توجد العديد من حزم البرامج الفعالة (تجارية ومجانية) لحل برامج الأعداد الصحيحة باستخدام تقنيات B&B، مثل SYMPHONY CPLEX، Gurobi، LINDO، SCIP، Xpress-MP، CBC [7].

ب. آلية عمل طريقة التفريع والتحديد (B&B):

عندما تكون حلول مشاكل البرمجة الخطية غير صحيحة (Non-Integer)، وتتطلب طبيعة المشكلة إيجاد حلول صحيحة (Integer)، فإنه يمكن استخدام طريقة (B&B) لإيجاد هذه الحلول، وذلك بإضافة قيدين جديدين على أحد متغيرات المشكلة، بحيث يُكَون كل قيد مع قيود المشكلة الأساسية مشكلة فرعية، وبهذا ينبثق من المشكلة الرئيسية مشكلتان فرعيتان، ثم يتم حل كل مشكلة على حدَة بإحدى الأساليب المستخدمة في حل مشاكل البرمجة الخطية كطريقة (Simplex) مثلاً، والجدير بالذكر هنا هو أن الهدف الأساسي من إنشاء هذه القيود هو تكوين مشاكل فرعية تكون قيمة الحل لمتغيراتها قيمة صحيحة، أي أنه يتم إضافة هذه القيود هو للتحرك من الحل الاساسي (Non-integer) باتجاه الحلول الصحيحة بهدف الوصول إلى الحل الأمثل (Integer). والجدير بالذكر هنا أيضاً هو أن كل مشكلة فرعية يتم إنشاؤها ستكون قيمة الحل للمتغير المستخدم في عملية التفريع (المتغير الذي تم إضافة قيد جديد له) قيمة صحيحة، وعندما يتوافق هذا مع قيمة المتغير الآخر وتكون قيمته هو الآخر قيمة صحيحة يتم الاحتفاظ بهذا الحل، باعتباره أحد الحلول الصحيحة للمشكلة الرئيسة التي سيتم المفاضلة بينها، تستمر عملية إنشاء المشاكل الفرعية (التفريع) حتى تستنفد كل المشاكل الفرعية وذلك إما بالوصول إلى مشكلة فرعية تكون قيم الحل لمتغيراتها قيم صحيحة أو لمشكلة فرعية ليس لها حل ممكن(Infeasible) . وبالتالي لا يمكن القيام بعملية التفريع، ويمكن تلخيص خطوات الحل لهذه الطريقة في الشكل التخطيطي (1) وتتلخص في التالي:

- الخطوة الأولى: إيجاد الحل الأمثل للمشكلة الرئيسية:

في هذه الخطوة يتم حل مشكلة البرمجة الخطية باستخدام أي من الطرق الشائعة كطريقة (Simplex)، فإذا كانت قيم كافة المتغيرات قيماً صحيحة، نتوقف عن الحل ويعتبر هذا هو الحل الأمثل للمشكلة، وتعتبر هذه الحالة خاصة لأن قيم الحل لكافة متغيرات المشكلة الرئيسية قيم صحيحة (Integer)، أما إذا كان الحل غير ذلك ننتقل للخطوة التالية.

- الخطوة الثانية: تكوين المشاكل الفرعية :

س هذه النقطة بشيء من التفصيل، وعليه سنفرض المثال التالي (النموذج الرياضي الأول) وهو عبارة عن مشكلة خطية بمتغيرين وقيدين:

$$Max\ Z = 5x_1 + 6x_2 \qquad (1)$$

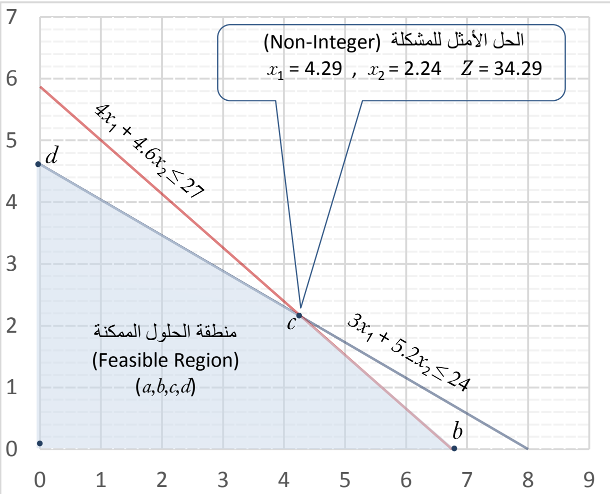
$$S.T:$$

$$4x_1 + 4.6x_2 \le 27 \qquad (2)$$
$$3x_1 + 5.2x_2 \le 24 \qquad (3)$$
$$x_1\ ,\ x_2\ \ge 0 \text{ and integer}$$

وقد تم حل هذه المشكلة باستخدام برنامج (QM for Windows V5) ويظهر الحل الأمثل لها المشكلة كالتالي:

$$Z = 34.29 \ ، \ x_1 = 4.29 \ ، \ x_2 = 2.14$$

كما يظهر ذلك في الشكل (2)، الذي يبين منطقة الحلول الممكنة (a,b,c,d) والحل الأمثل عند النقطة (c).



شكل (2) منطقة الحلول الممكنة والحل الأمثل لمشكلة خطية ذات متغيرين وقيدين

بناءً على الخطوات المتبعة في طريقة (B&B)، فإنه يتم تجزئة المشكلة الرئيسية إلى مشكلتين فرعيتين، يتم تكوين المشكلة الأولى عن طريق إضافة قيد يحمل علامة (≤) والثانية بإضافة قيد يحمل علامة (≥)، وسيتم هنا تفريع المشكلة على أساس المتغير الذي يحمل أكبر كسر وهو المتغير (x_1) والذي قيمته (4.29) وذلك كما يلي:

- **المشكلة الفرعية الأولى:** يتم تكوين هذه المشكلة الفرعية بإضافة القيد الأول على المتغير (x_1)، بحيث يكون هذا المتغير هو طرفها الأيسر ويكون طرفها الأيمن مساوية لقيمة العدد الصحيح لهذا المتغير وهي (4)، وبهذا يكون القيد الإضافي كالتالي:

$$x_1 \le 4 \qquad (4)$$

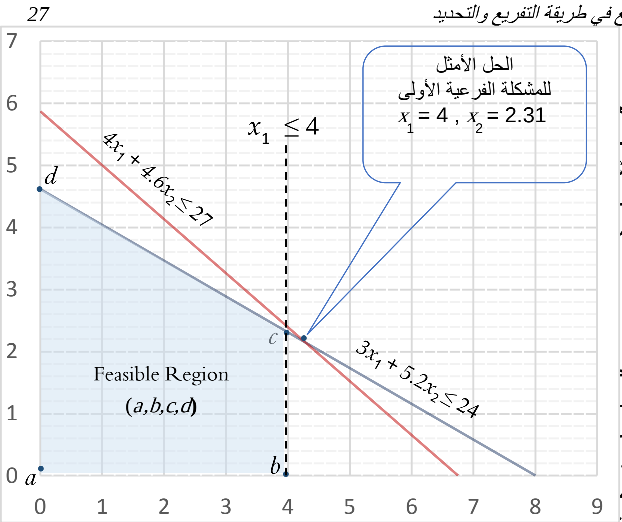
وبهذا يكون النموذج الرياضي للمشكلة الفرعية الأولى كالتالي:

$$Max\ Z = 5x_1 + 6x_2 \qquad (1)$$

$$S.T:$$

$$4x_1 + 4.6x_2 \le 27 \qquad (2)$$
$$3x_1 + 5.2x_2 \le 24 \qquad (3)$$
$$x_1 \le 4 \qquad (4)$$
$$x_1\ ,\ x_2\ \ge 0$$

وبهذا يكون الشكل البياني للمشكلة الفرعية الأولى كما يظهر في الشكل البياني رقم (3).
سس



شكل (3) المشكلة الفرعية الأولى بإضافة القيد ($x_1 \le 4$) على المتغير (x_1)

- ... طب ... صـ ... للمـ ... الصـ ... الأ ... لمنا ... الصـ ... إضا ... البد ... (الـ

للمتغير الأول وهكذا.. الهدف من ذلك هو تقسيم منطقة الحل إلى أجزاء ويستبعد منها المنطقة التي تحتوي على حلول غير صحيحة (كسرية أو مختلط)، فالقيود الإضافية هنا تستخدم كقواطع لاستبعاد المنطقة التي تحتوي على حلول غير صحيحة، بمعنى آخر تقسيم منطقة الحلول الممكنة إلى جزئيين (مشكلتين) بهدف حل كل مشكلة على حِدَة، والحل الأمثل لهذه المشكلة يمكن أن يكون حلاً صحيح (قيم كل المتغيرات قيم صحيحة) أو تكون مختلطة (بعضها صحيح والأخر كسري). كما يمكن ألا يحتوي أحد هذه الأجزاء على حل اصلاً، والجدير بالذكر هنا هو أن عملية الاستقطاع تتم في كل مرة بين عددين صحيحين فقط، أي إن عملية الاستقطاع هذه لا تتضمن هذين العددين، وتكون هذه الاعداد الصحيحة من ضمن الحلول التي ستدخل في عملية المفاضلة، وللوقوف على كل ما ذكر نقوم بتفريع النموذج الرياضي السابق الكائن في ص (3) (الممثل بالشكل البياني (2)) مرة باستخدام المتغير (x_1) وأخرى باستخدام المتغير (x_2) وتتبع هذه العمليات بيانياً.

- التفريع على أساس المتغير الأول (x_1):

سنقوم هنا بتفريع المشكلة على أساس المتغير ذو الكسر الأكبر وهو (x_1)، وذلك بإضافة قيدين على المشكلة الرئيسية كلاً على حدة، وبما إن قيمة المتغير (x_1) تساوي 4.29، و هذا يعني استبعاد هذا الحل (2.14 ، 4.29) وذلك باستبعاد المنطقة المحصورة بين العدد 4 والعدد 5 على المحور الأفقي وذلك كما في الشكل (7)، بحيث يشكل كل قيد إضافي مع قيود المشكلة الرئيسية مشكلة فرعية مستقلة، وهذه القيود كما يلي:

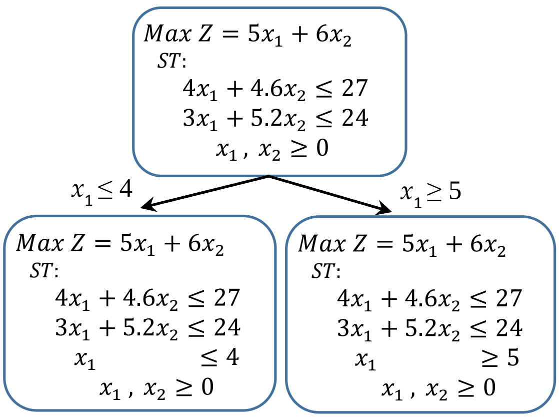
يضاف هذا القيد ليكون المشكلة الفرعية الأولى

$$x_1 \le 4 \qquad (4)$$

يضاف هذا القيد ليكون المشكلة الفرعية الثانية

$$x_1 \ge 5 \qquad (5)$$

وعليه يمكن تمثيل عملية التفريع على أساس المتغير الأول (x_1) كما في الشكل (5)



الشكل (5) مخطط لتفريع النموذج الرياضي الأول على أساس المتغير (x_1)

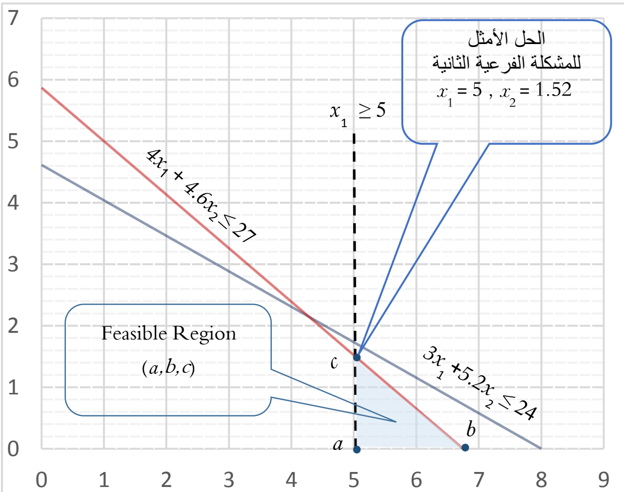
➢ **المشكلة الفرعية الثانية:** بنفس الفكرة يتم تكوين المشكلة الفرعية الثانية، إلا أنه هنا يتم إضافة القيد بعلامة أكبر أو يساوي (≥) ويكون الطرف الأيمن لهذا القيد مساوية لقيمة المتغير مضافاً إليه واحد ($4 + 1 = 5$)، وعليه يكون القيد الإضافي كالتالي:

$$x_1 \ge 5 \qquad (5)$$

وبهذا يكون النموذج الرياضي للمشكلة الفرعية الثانية كالتالي:

$$
\begin{aligned}
&Max\ Z = 5x_1 + 6 && (1)\\
&S.T:\\
&\quad 4x_1 + 4.6x_2 \le 27 && (2)\\
&\quad 3x_1 + 5.2x_2 \le 24 && (3)\\
&\quad x_1 \qquad\quad \ge 5 && (5)\\
&\quad x_1 , x_2 \ \ge 0
\end{aligned}
$$

سوبهذا يكون الشكل البياني للمشكلة الفرعية الثانية كما يظهر في الشكل (4).



- الخطوة الثالثة: إيجاد الحل الأمثل للمشاكل الفرعية:

في هذه الخطوة يتم حل المشاكل الفرعية كلاً على حدة، وذلك باستخدام أي من الطرق الشائعة كطريقة (Simplex Method) ، طبعاً هنا سيكون للقيود الجديدة تأثير على ناتج الحل للمتغير المستخدم في عملية التفريع فتكون س

حل المشكلة الفرعية الأولى:

$$Z = 33.85 \ ، \ x_1 = 4 \ ، \ x_2 = 2.31$$

حل المشكلة الفرعية الثانية:

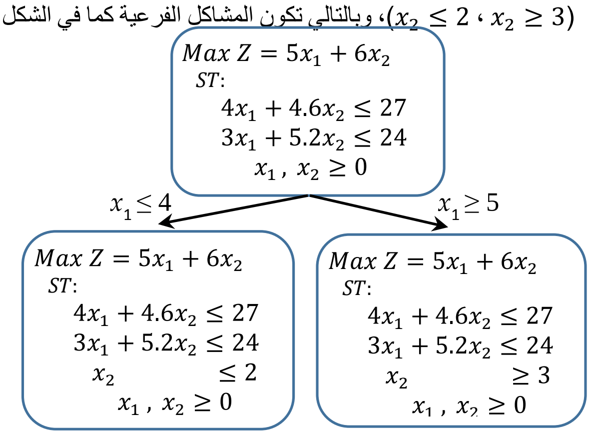
$$Z = 34.13 \ ، \ x_1 = 5 \ ، \ x_2 = 1.52$$

والجدير بالذكر هنا هو أن قيم المتغير (x_1) عند الحل الأمثل للمشكلتين الفرعيتين ستكون قيمه صحيحة، وكل منهما مساوٍ للطرف الأيمن للقيد المستخدم في عملية التفريع. فقيمة الحل للمتغير (x_1) في المشكلة الفرعية الأولى والتي تم تفريعها بالقيد $(x_1 \leq 4)$ تساوي (4)، وقيمة ذات المتغير في المشكلة الفرعية الثانية والتي تم تفريعها بالقيد $(x_1 \geq 5)$ تساوي (5) كما هو موضح في الشكل (6).

ويمكن توضيح عملية التفريع هذه بيانياً، وذلك عن طريق استخدام الشكل البياني للمشكلة الرئيسية بعد استقطاع الجزء الذي يعنى باستبعاد المنطقة المتضمنة للحل الأمثل (Non-integer) عن طريق القيدين الإضافيين. المنطقة المحصورة بين العددين الصحيحين (5 ، 4) على المحور الأفقي (x_1)، ويتم ذلك جبرياً عن طريق إضافة القيدين $(x_1 \leq 4)$ و$(x_1 \geq 5)$ للمشكلة الرئيسية، وبهذا تتجزأ المشكلة إلى مشكلتين فرعيتين، كما يظهر في الشكل التالي (7)، فتكون منطقة الحلول الممكنة للمشكلة الفرعية الأولى هي المنطقة المحصورة بين النقاط (a,b,c) ومنطقة الحلول الممكنة للمشكلة الثانية هي المنطقة المحصورة بين النقاط (d,e,f,g).

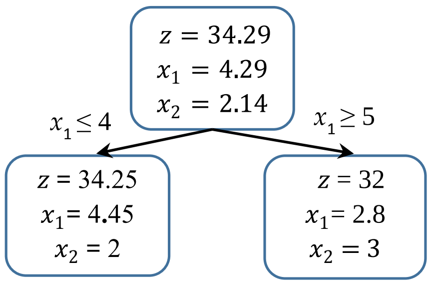
- التفريع على أساس المتغير الثاني (x_2):

ستتم هنا عملية التفريع على أساس المتغير (x_2)، وذلك بإضافة القيدين $(x_2 \leq 2$ ، $x_2 \geq 3)$، وبالتالي تكون المشاكل الفرعية كما في الشكل (8).



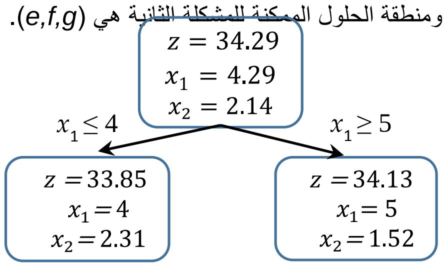
الشكل (8) مخطط لتفريع النموذج الرياضي الأول على أساس المتغير (x_2)

وبنفس المنطق الســـابق فـإن قيم المتغير (x_2) عند الحل الأمثل في كلا المشكلتين ستكون قيمه صحيحة، ففي المشكلة الفرعية الأولى تساوي (2)، وقيمته في المشــكلة الفرعية الثانية تكون (3) كما هو موضــح في الشــكل (9).

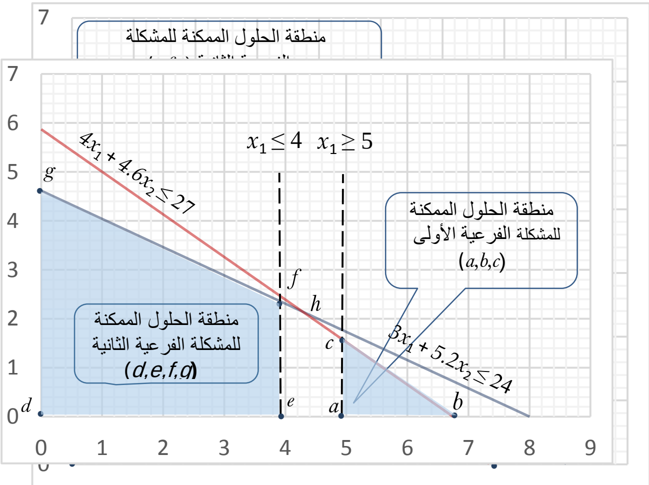


الشكل (9) مخطط لحلول النموذج الرياضي الأول على أساس (x_2)

يمكن توضـــيح عملية التفريع بيانياً عن طريق الشـــكل البياني للمشـــكلة الرئيسـية بعد اسـتبعاد المنطقة المتضـمنة للحل الأمثل (Non-integer)، وهي المنطقـة المحـددة بـالقيـدين $(x_2 \geq 3)$ و $(x_2 \leq 2)$، وبهذا تتجزأ المشـكلة إلى مشـكلتين فرعيتين. كما يظهر في الشـكل (10) فتكون منطقة الحلول الممكنة للمشكلة الفرعية الأولى هي المنطقة المحصورة بين النقاط (*a,b,c,d*) ومنطقة الحلول الممكنة للمشكلة الثانية هي (*e,f,g*).



الشكل (6) مخطط لحلول النموذج الرياضي الأول على أساس المتغير (x_1)



الشكل (7) المشكلتين الفرعيتين على أساس المتغير الأول (x_1)

الشكل (10) يوضح المشاكل الفرعية على أساس المتغير الثاني (x_2)

5. دراسة تحليلية للمقارنة بين التفريع على أساس (x_1, x_2):

بعد دراســـة وتحليل عملية التفريع على أســـاس المتغير الأول (x_1) والمتغير الثاني (x_2)، وتتبع عملية التفريع باســـتخدام الطريقة البيانية، تم استنتاج الملاحظات التالية:

- في عمليـة التفريع الأولى (على أســـاس المتغير (x_1))، تمـت باســـتخدام قيود إضـــافية على المتغير (x_1)، وهي عبارة عن عدد أثنين من القواطع تكون عمودية على المحور (x_1). الهدف منها اســـتبعاد المنطقة المحصـــورة بين هذه القواطع، وبهذه القواطع تتجزأ المشـــكلة الرئيســـية إلى مشـــكلتين فرعيتين كما في الشـــكل (7)، وتكون قيم هذا المتغير عند الحل الأمثل في كلا المشـــكلتين قيم صـــحيحة. أما في حالة التفريع على أســـاس المتغير الثاني (x_2) تكون القواطع على شكل أفقي (عمودية المحور العمودي (x_2)). وبهذا تتجزأ المشـــكلة الرئيســـية لتكون المنطقة المســـتبعدة موازية للمحور الأفقي كما في الشـكل (10)، وفي هذه الحالة أيضـاً تكون قيماً الحل الأمثل لهذا المتغير في كلا المشكلتين الفرعيتين قيماً صحيحة.
- إن عمليـة التفريع تكون تبـديليـة أي أنهـا تكون بـالتنـاوب بين المتغيرين، فمثلاً إذا تمت التجزئة على أساس المتغير الأول، فإن التجزئة في المسـتوى الثاني (في المرة الثانية) سـتكون على أسـاس المتغير الثاني، وفي المسـتوى الذي يليه سـتكون على أسـاس المتغير الأول وهكذا. أما إذا كانت بداية التفريع على أسـاس المتغير الثاني، فإن التجزئة التي تليها سـتكون على أسـاس المتغير الأول وهكذا، وهذا يعني بيانيا أنه إذا تمت التجزئة في البداية على شـكل قاطع رأسـي فإنه في التجزئة التي يليها سـتكون على شـكل قاطع أفقي، وفي الدورة أو المستوى الذي يليه تكون على شكل قاطع رأسي وهكذا.

من أجل توضـــيح كل ما ذكر وتبيان كيفية القيام بعملية التفريع وما هي الحلول الفرعية التي يتم الوصـول إليها وما مدى إمكانية تسـاوي المشـاكل الفرعية من حيث العدد والشـكل عند اختلاف المتغير المسـتخدم في عملية

التفريع، والتأكيد على أنه ســيتم الوصــول إلى الحل الأمثل (Integer) في حالة اختلاف المتغير المســتخدم في عملية التفريع وتقديم التحليلات المنطقية التي تؤيد ذلك. ســنقوم بحل النموذج الرياضــي الســابق (الأول) بالطريقتين (مرة باســتخدام (x_1) ومرة باســتخدام (x_2)) والذي يمثل حالة التســاوي في عدد المشاكل الفرعية وسنتتبع ذلك بيانياً، كما سنقوم بفرض نموذج رياضي أخر ليمثل حالة الاختلاف في عدد المشاكل الفرعية.

6. حل النماذج في حالتي التساوي والاختلاف في عدد المشاكل

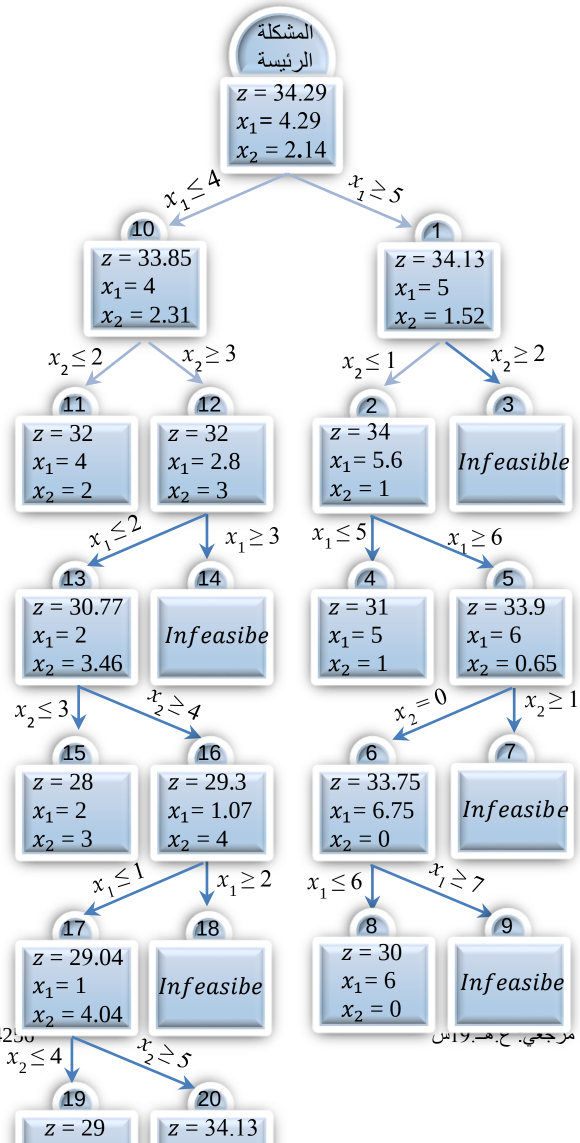
نؤكد هنا أن عدد المشاكل الفرعية يمكن أن تتساوى أو تختلف باختلاف المتغير المســتخدم في عملية التفريع، ففي هذه الفقرة ســيتم حل عدد 2 من الأمثلة التوضـيحية، بحيث يمثل النموذج الأول حالة التسـاوي، ويمثل الثاني حالة الاختلاف في عدد المشاكل الفرعية وذلك كما يلي:

أ. في حالة تساوي عدد المشاكل الفرعية:

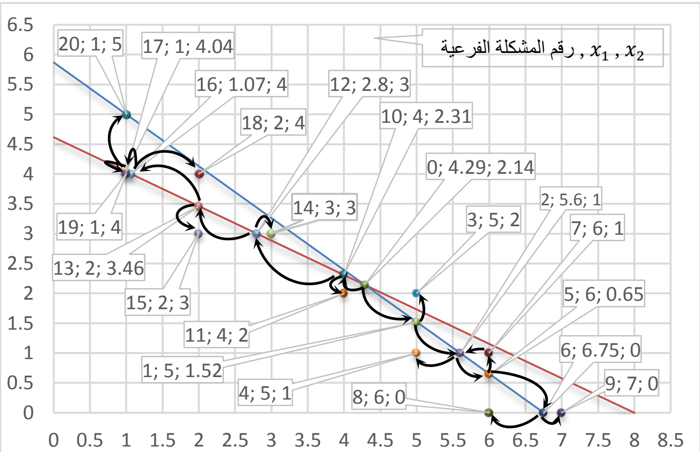
في هـذه الفقـرة سنقـوم بحـل المثـال السـابـق بكـلا الحالـتين، فـي الأولـى ســـــيتم استخدام المتغير ذي الكسر الأكبر(x_1)، وفي الثانية سيتم استخدام المتغير ذي الكسر الأصغر(x_2). وذلك للوقف على أن عدد المشاكل الفرعية في الحالتين قد يتســـاوى كما أنه هناك تناظر بين هذه المشـــاكل الفرعية، والنموذج الرياضي (الأول) يمثل حالة تساوي المشاكل الفرعية، لذلك سنقوم بحله مع توضيح المشاكل الفرعية في كل حالة (مرة باستخدام (x_1)، وأخرى باستخدام (x_2)). ثم نقوم بعملية المقارنة بين الحالتين من حيث عدد المشاكل الفرعية المتولدة عن كل حالة، ومدى وجود تناظر بين المشـاكل الفرعية في الحالتين.

- الحل باستخدام المتغير ذو الكسر الأكبر (x_1):

هنا قمنا بعملية التفريع باسـتخدام المتغير (x_1)، الشـكل (11) يوضـح حلول المشاكل الفرعية الناتجة من عملية التفريع والقيود المستخدمة في ذلك، حيث بلغ عدد المشـــاكل الفرعية 20 مشـــكلة فرعية من خلال تتبع المســـارات والتفرعـات في هذا المخطط نلاحظ بـأنـه تم الوصـــول الي الحل الأمثل الصحيح، بعد القيام بتفريع المشكلة الرئيسية إلى 20 مشكلة فرعية.



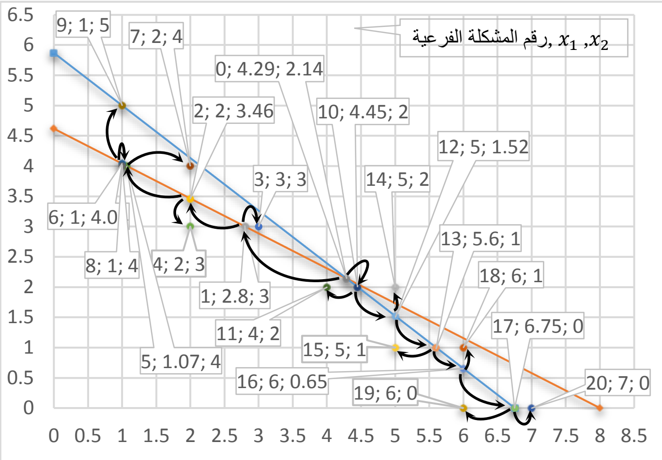
الشكل البياني (12) يوضـح المشكلة الرئيسية ونقاط الحل لتفرعاتها، كما يوضـــح عملية التحرك لـهذه النقاط والتي تبدأ من نقطة الحل للمشـــكلة الرئيسية [0, 4.29, 2.14]. ثم تنتقل أو تتفرع إلى نقطتين هما الحل (1) وهو الممثـــــل بالنقطـــــة [1, 5, 1.52] والحل رقم (10) وهو الممثل بالنقـطـــــة [10, 4, 2.31]، ومن الحل (1) يتم الانتقال للحـــــل (2) والممثــــل في النقطــة [2, 5.6, 1] والحـل (3) الممثـل في النقـطـة [3, 5, 2] والذي يظهر حل غير ممكن، وهكذا يمكن تتبع باقي الحلول عن طريق تتبع اتجاهات الأسهم كما في الشكل (12).



الشكل (12) حركة حلول المشاكل الفرعية عند التساوي والتفريع استخدام (x_1)

- الحل باستخدام المتغير ذي الكسر الأصغر (x_2):

هنا سـيتم تفريع المشـكلة الرئيسـية باسـتخدام المتغير ذي الكسـر الأصـغر (x_2 = 2.29)، والرسم التخطيطي (13) يوضح عمليات التفريع والحلول الخاصة بالمشاكل الفرعية والناتجة من عمليات التفريع.



الشكل (14) حركة حلول المشاكل الفرعية في حالة تساوي عددها وتفريعها عند استخدام المتغير (x_2)

والجدير بالذكر هنا هو أن عدد المشاكل الفرعية في هذه الحالة (التفريع على أساس المتغير ((x_2)) يساوي عدد المشاكل الفرعية الناتجة في الحالة الأولى (التفريع على أســـاس المتغير ((x_1)). وبالتدقيق في حلول هذه المشــاكل الفرعية في الحالتين نجدها متطابقة، باســتثناء المشــكلة الفرعية (10) في الحالتين فهي لم تتطابق.

ب. في حالة وجود اختلاف في عدد المشاكل الفرعية:

في هذه الفقرة ســنفرض نموذجاً رياضــياً أخر تختلف فيه عدد المشــاكل الفرعيـة عنـد اختلاف المتغير المســتخـدم في عمليـة التفريع، والنموذج الرياضي التالي (الثاني) يجسد هذه الحالة وهو:

$$Max\ Z = 5x_1 + 8x_2 \qquad (6)$$

$$S.T:$$

$$x_1 + x_2 \leq 6 \qquad (7)$$

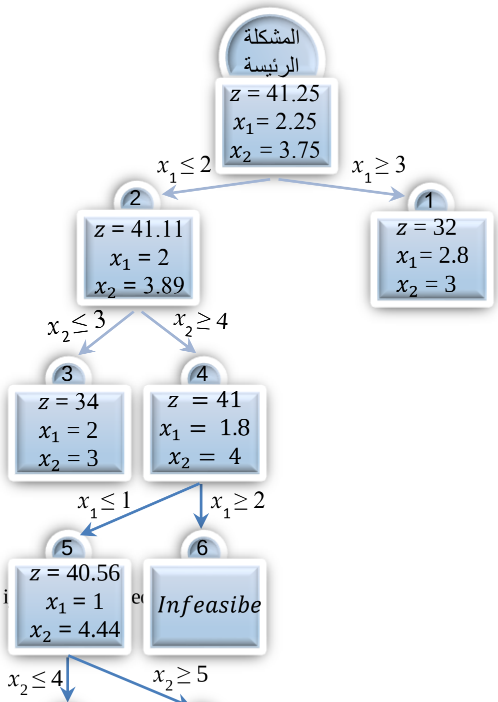
$$5x_1 + 9x_2 \leq 45 \qquad (8)$$

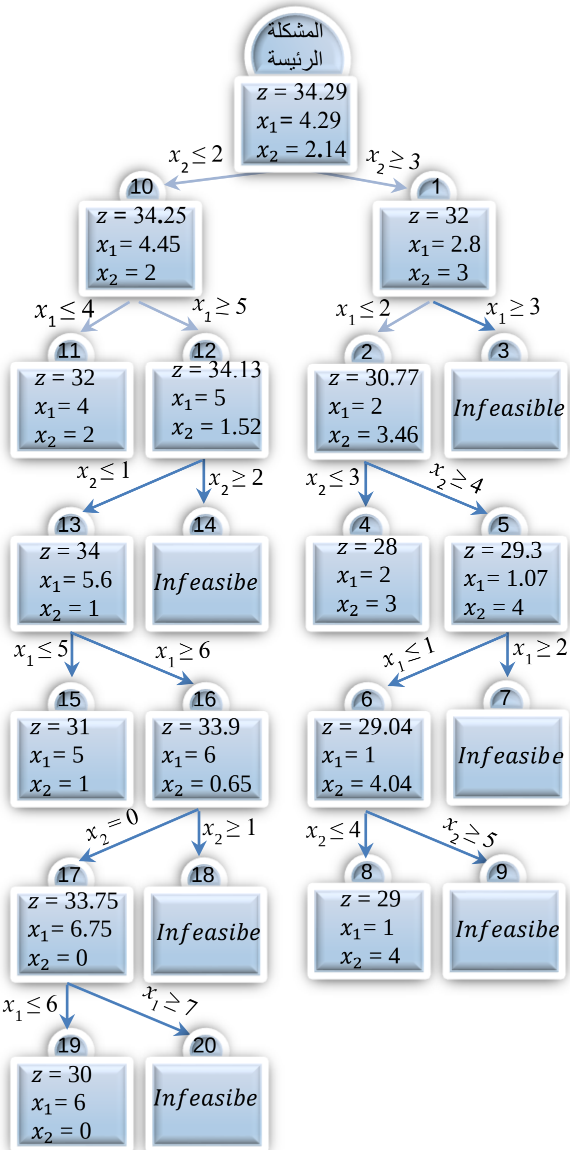
x_1 , x_2 ≥ 0 and integer

الحل الأمثل للنموذج ($Z = 41.25$ ، $x_1 = 2.25$ ، $x_2 = 3.75$)، ومن أجل الوقوف على مـا ذكر أعلاه نقوم بحـل هـذا النموذج في بكلا الحالتين كالتالي:

- الحل باستخدام المتغير الأول (x_1):

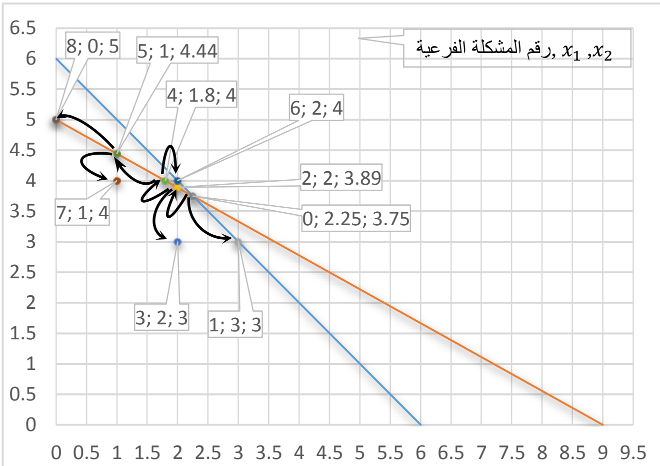
في هذه الفقرة ســيتم عملية التفريع باســتخدام المتغير الأول (x_1) والذي يحتوي على الكســر الأصــغر وقيمته (2.25). والرسم التخطيطي (15) يوضــح حلول المشـــاكل الفرعيـة المتولدة من عملية التفريع والقيود المستخدمة فيها، وقد بلغ عدد هذه المشاكل في هذه الحالة 8 مشاكل فرعية، كما يمكن تتبعها من خلال اتجاه الأسهم في الشكل البياني (16).





الشكل (13) تفرعات المشكلة الرئيسية في حالة تساوي عددها وتفريعها باستخدام المتغير (x_2)

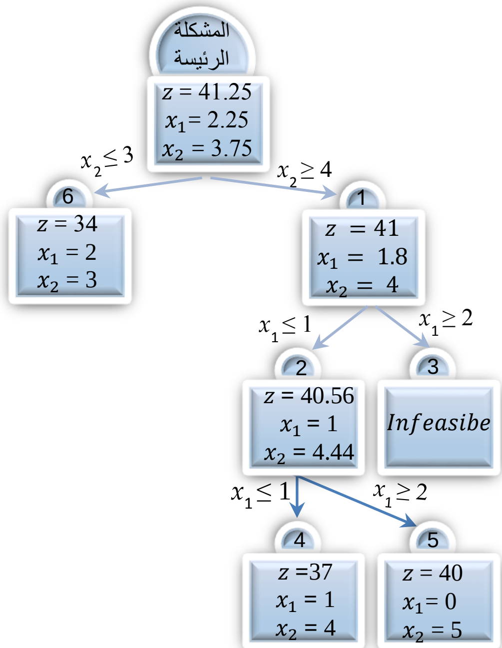
ولتوضــيح أكثر لعملية التفريع، يمكن تتبع الرســم التخطيطي (13) بيانياً، وذلك عن طريق الرسم البياني للمشـكلة الرئيسـية ومنطقة الحلول الممكنة لها، وذلك. كما هو موضــح في الرســم البياني (14)، الذي يوضــح حركة عملية التفريع من خلال تتبعها عن طريق اتجاهات الأسهم على الرسم.



الشكل (16) حركة حلول المشاكل الفرعية في حالة الاختلاف في عددها وتفريعها باستخدام المتغير (x_1)

- الحل باستخدام المتغير الثاني (x_2):

هنا ستتم تفريع المشكلة الرئيسية باستخدام المتغير الثاني وهو ذو الكسر الأكبر ($x_2 = 3.75$). الرسم التخطيطي (17) يوضح عمليات التجزئة وحلول المشاكل الفرعية الناتجة من عمليات التجزئة.



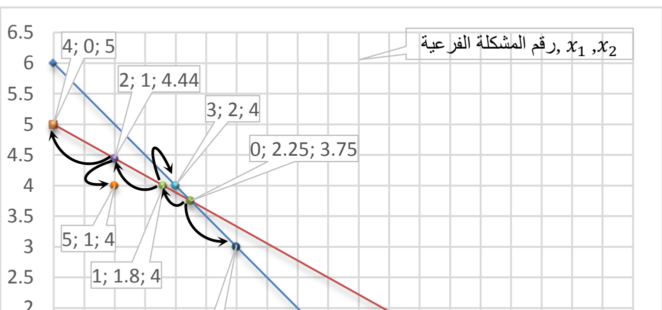
ز

الشكل (17) تفرعات المشكلة الرئيسية في حالة الاختلاف في عدد تفريعها على باستخدام المتغير (x_2)

جدول (1) حلول المشاكل الفرعية في حالتي التساوي والاختلاف في عددها عند تفريعهما على أساس المتغير (x_1) والمتغير (x_2)

نتائج الحالة الأولى: عند التساوي في عدد المشاكل الفرعية (20)					نتائج الحالة الثانية: عند الاختلاف في عدد المشاكل الفرعية (6 ، 8)				
ر.م	الحلول المتاحة من عملية التفريع		التناظر في المشاكل الفرعية		ر.م	الحلول المتاحة من عملية التفريع		التناظر في المشاكل الفرعية	
	z	(x_1, x_2)	على أساس (x_1)	على أساس (x_2)		z	(x_1, x_2)	على أساس (x_1)	على أساس (x_2)
1	34.13	(5 , 1.52)	1	12	1	41	(1.8 , 4)	1	4
2	34	(5.6 , 1)	2	13	2	40.56	(1 , 4.44)	2	5
3	-	Infeasible	3	14	3	-	Infeasible	3	6
4	31	(5 , 1)	4	15	4	37	(1 , 4)	4	7
5	33.9	(6 , 0.65)	5	16	5	40	(0 , 5)	5	8
6	33.75	(1.75 , 0)	6	17	6	39	(3 , 3)	6	1
7	-	Infeasible	7	18	7	41.11	(2 , 3.89)	-	2
8	30	(6 , 0)	8	19	8	34	(2 , 3)	-	3
9	-	Infeasible	9	20					
10	33.85	(4 , 2.31)	10	-					
11	32	(4 , 2)	11	11					
12	32	(2.8 , 3)	12	1					
13	30.77	(2 , 3.46)	13	2					
14	-	Infeasible	14	3					
15	28	(2 , 3)	15	4					
16	29.3	(1.07 , 4)	16	5					
17	29.04	(1 , 4.04)	17	6					
18	-	Infeasible	18	7					
19	29	(1 , 4)	19	8					
20	-	Infeasible	20	9					
21	34.25	(4.45 , 2)	-	10					

ولتوضيح عملية التفريع ونقاط الحل بيانياً، يمكن تتبع اتجاه الأسهم لعملية التفريع الموضحة في الشكل البياني (18).



7. المقارنة بين قيم الحل للمشاكل الفرعية في حالتي التساوي والاختلاف في عدد المشاكل الفرعية:

بعد حل النموذج الرياضـــي الخاص بكل حالة (حالة التســـاوي في المشـاكل الفرعية وحالة الاختلاف في عددها) بكلا الطريقتين (التفريع على أســـاس المتغير الأول (x_1) والتفريع على أســـاس المتغير الثاني (x_2))، ســـنقوم بمقارنة حلول هذه المشـــاكل الفرعية، بهدف الوقوف على مدى تطابقها مع بعضها البعض في حالة التساوي في عددها وفي حالة الاختلاف أيضـاً. وبهدف الخروج باستنتاجات تمكننا من الوصـول إلى أاهداف هذه الورقـة، تم تجميع هـذه النتـائج في الجـدول (1)، فيمثـل الجزء الأيمن من الجدول حالة التساوي، والتي تظهر فيها نتائج الحل لـ (20) مشكلة فرعية في كـلا الحالتين (عند استخدام المتغير (x_1) وعند استخدام المتغير (x_2))، أما الجزء الأيسر من الجدول (حالة عدم التساوي) فيظهر حلول لـــ (8) مشاكل فرعية، وعند استخدام المتغير (x_1) في عملية التفريع فيظهر حلـول لــ (6) مشاكل فرعية فقط عند استخدام المتغير (x_2) في عملية التفريع.

عند دراســة الجدول (1) دراســة مقارنة واســتخدام الرســوم البيانية والتحليل السـابق لآلية عمل طريقة (B&B) يمكن استخلاص بعض النقاط المهمة والتي يمكن الاستفادة منها عند استخدامها في حل مشاكل البرمجة الخطية ذات الأعداد الصـــحيحة (PIPP)، ويمكن تلخيص ذلك كما في الفقرة التالية:

8. استنتاجات الدراسة

1) يمكن اســـتخدام أي من المتغيرات القرارية (x_1 , x_2) للقيام بعملية التفريع، كلاهما يؤديان إلى ذات الحل الصـحيح، وإن الاختلاف القائم بين المؤلفين (استخدام المتغير ذو الكسر الأكبر أو الأصغر)، ليس له مبرر. ففكرة طريقـة (B&B) تعتمـد على تجزئـة منطقـة الحلول الممكنة إلى مشـــاكل فرعية وذلك على عدة مراحل، ومن تم حل كل مشـكلة فرعية على حِدة، والفرق في اسـتخدام أي من المتغيرات في عمليـة التفريع يكون في البدايـة فقط، فمثلاً إذا بـدأت عمليـة التفريع باستخدام المتغير الأول، ففي المرحلة الثانية (المسـتوى الثاني) سـيتم اسـتخدام المتغير الثاني في عملية التفريع، وفي المستوى الثالث سيتم العودة واسـتخدام المتغير الأول ثم نعود للثاني في المســتوى الثالث، وهكذا.. والعكس صـــحيح إذا كانت البداية بالمتغير الثاني، أي أن مسألة التفريع عملية تبديلية تتم بين المتغيرين كمـا تتضح في الأشكال التخطيطية (11 ، 13 ، 15 ، 17) عند مختلف مســـتويات عملية التفريع.

2) يمكن أن تتسـاوى أو تختلف عدد المشـاكل الفرعية المتولدة من عملية التفريع في كلا الحـالتين (عنـد اســـتخـدام المتغير الأول أو المتغير الثاني)، ففي الحالة الأول بالجدول (1) نلاحظ بأن عدد المشـــاكل الفرعية الناتجة من التفريع على أساس المتغير (x1) تساوت مع عدد المشاكل الفرعية المتولدة من التفريع على أسـاس المتغير(x2). حيث كان عدد المشــاكل الفرعية (20) مشـكلة فرعية في كلا الطريقتين، وفي الحالة الثانية (حالة الاختلاف) في ذات الجدول (1) لم يتســـاو عدد المشـاكل الفرعية في الحالتين، حيث كان عدد المشـاكل الفرعية عند التفريع على أسـاس المتغير (x1) هو (8) مشـاكل فرعية، وعند اســـتخدام المتغير الثاني (x2) في عملية التفريع كان عدد المشـــاكل الفرعية (6) مشاكل فرعية فقط.

3) يرجع سـبب الاختلاف في عدد المشـاكل الفرعية أحياناً كما ذكر في النقطة السابقة إلى ظهور حل صـحيح (Integer) أو عدم وجود حل أصـــلاً عند القيام بعملية التفريع في أول مرة. وهذا يعني طبعاً عدم إمكانية الاستمرار في عملية التفريع في هذا الجانب، نظراً لأن عملية التفريع عملية تبديلية (يتبدل المتغير المســتخدم في عملية التفريع عند كل مستوى) فإن المشـاكل الفرعية المتولدة في الشـق الأيمن في حالة استخدام المتغير الأول سيناظر المشـاكل الفرعية في الجانب الأيسر عند اســـتخدام المتغير الثاني. بمعنى المشـــاكل الفرعية المتولدة في المستوى الأول في الجانب الأيمن ستتناظر مع مشاكل الجانب الأيسر في المســـتوى الثاني، وبعد ذلك ســـتتناظر المشـــاكل المتولدة من كلٍ منهما، بمعنى عندما تتناظر مشكلتان فهذا يعني أنها متساوية في الحل وبالتالي طبعاً ستكون المشاكل المتفرعة من كلٍ منها متساوية، ويمكن ملاحظة هذا من الشـــكلين التخطيطيين (13 ، 11)، ويظهر الجدول (1) هذا أيضـاً، ففي حالة التسـاوي في عدد المشـاكل نجد المشـاكل الفـــرعيـــة (9.., 3, 2, 1) في حالة التفريع على أساس (x_1) تناظر المشـاكل (20..، 14 ، 13 ، 12) في حـالة التفـريـع على أسـاس المتغير (x_2)، وفي المقابل المشـــاكل (20..,12, 11, 10) تناظر المشـاكل الفرعية (11..,3, 2, 1) باستثناء المشكلة الفرعية (10) كما سبق الذكر، ويلاحظ حدوث ذلك في الحالة الأخرى (عدم تساوي عـدد المشـــاكـل الفرعيـة) حيث يتبين ظهور حـل أمثـل صـــحيح (Integer) في بداية أول تفريع كما يظهر في الشكل (15) والشكل (17).

4) يمكن العمل للاســـتفادة من النقطة الســـابقة (2) في تخفيض عدد العمليات الحسابية اللازمة للوصول للحل الأمثل الصحيح. بمعنى أنه في حالة الاختلاف في عدد التفريعات فهذا يعني أن اســـتخدام متغير دون غيره سـيولد عدداً أقل من التفريعات، وأن اسـتخدام هذا المتغير في عملية التفريع توصـــلك للحل الصـــحيح بأقل عدد من العمليات الحســابية (أقل جهد ووقت). يتضـح هذا في الحالة الثانية، فعند القيام بعملية التفريع على أســـاس (x_2) تم الوصـــول للحل بعد حل (6) مشـــاكل فرعية فقط، في حين أنه عند اســـتخدام المتغير (x_1) تم الوصول للحل الأمثل بعد حل عدد (8) مشاكل فرعية.

9. الخلاصة

تم إعداد هذه الدراســـة على طريقة التفريع والتحديد لأنها من أهم الطرق المستخدمة في إيجاد الحل الأمثل الصحيح، والتي فيها يتم تفريع المشكلة الرئيسية إلى عدة مشاكل فرعية. من خلال حل هذه المشاكل الفرعية يمكن الوصــول للحل الأمثل (Integer) للمشـكلة الرئيسـية، وهو أفضــل حل (Integer) من بين حلول هذه المشـــاكل الفرعية، كانت الفكرة من وراء انطلاق هذا البحث هو وجود اختلاف بين البحاث والمؤلفين في تحديد المتغير المستخدم في عملية التفريع، وعدم تقديمهم أي تفسير أو تحليل يؤيد نظريتهم في تحديد المتغير المستخدم في عملية التفريع.

وبعد الرجوع لأدبيات الموضوع والغوص في تحليل آلية عمل هذه الطريقة (B&B)، وتحليل خطواتها وتتبع آلية عملها باســـتخدام الطريقة البيانية، وتتبع حركة حلول المشاكل الفرعية المتولدة من المشكلة الرئيسية بيانياً. ثم تقديم تحليلات منطقية تبين أن عملية التفريع تتم على أســـاس التناوب بين المتغيرين، فإذا بدأت بالمتغير الأول فإنه في المســـتوى الثاني منه ســـيتم التفريع على أسـاس المتغير الثاني ثم الرجوع للمتغير الأول في المسـتوى الذي يليه وهكذا..، والعكس صحيح إذا بدأت عملية التفريع بالمتغير الثاني، كما أن عدد المشـــاكل الفرعية المتولدة من عملية التفريع قد تتســـاوى أو تختلف باختلاف المتغير المستخدم في عملية التفريع. وهذا يعتمد على الحل المتحصـــل عليه في أول عملية تفريع (المســـتوى الأول)، فيختلف عدد المشـاكل إذا كان حل المشـكلة الفرعية الأول حلاً صـحيحًا أو ليس لها حل أصـلاً لأنه لا يمكن القيام بعملية التفريع، وهنا يمكن العمل على هذه النقطة الاستفادة منها في بتقليل العمليات الحسابية أو الجهد المبذول للوصـول إلى الحل الأمثل للمشكلة الرئيسية.

المراجع

[1] العبيدي محمود، ومؤيد الفضل. (2004). *بحوث العمليات وتطبيقاتها في لإدارة الأعمال*. الدراق للنشر والتوزيع.

[2] طه حمدي. (1996). مقدمة في بحوث العمليات. دار المريخ للنشر.

[3] Winston, Wayne L, 2004. *Operations Research*, Brooks/Cole - Thomson Learning

[4] Carter, Michael W, Price, Camille C, Rabad, Ghaith (2019). Operations Research A Practical Introduction, CRC Press.

[5] A. H. Land and A. G. Doig.1960. *An Automatic Method of Solving Discrete Programming Problems.* Econometrica, Vol. 28, No3.

[6] Takuya Akiba and Yoichi Iwata. 2016. *Branch-and-reduce exponential/FPT algorithms in practice: A* case *study of vertex cover.* Theoretical Computer Science,609, 211-225.

[7] David R. Morrison, Sheldon H. Jacobson, Jason J. Sauppe, Edward C. Sewell. *2016*. Branch-*and-bound algorithms: A survey of recent advances in searching, branching, and pruning.* Discrete Optimization, *79-102.*

[8] رابح، بوعراب. (2016) تطبيقات في مقياس البرمجة المعمقة. كلية العلوم الاقتصـــادية والعلوم التجارية وعلوم التسيير.

[9] دريباتي، محمد مزيد. (2014). خوارزمية القطع والتفريغ الجديدة لحل مســائل الخطية الصحيحة. مجلة جامعة تشرين للبحوث والدراسات العلمية. مجلد 36 العدد (3).

[10] T. Achterberg, T. Koch, A. Martin. 2005. *Branching rules revisited*. Oper. Res. Lett. 33, 42-54.

[11] F. Ortega, L.A. Wolsey. 2003*. A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow Problem*. Networks, 143–158.

[12] M. Benichou, J.M. Gauthier, P. Girodet, G. Hentges, G. Ribiere, O. Vincent. 1971. *Experiments in mixed-integer linear.* programming, Math. Program, 76–94.

[13] D. Applegate, R.E. Bixby, V. Chv´atal, W. Cook.1995*. Finding cuts in the TSP. Technical Report* 95-05*,* DIMACS.

[14] J.T. Linderoth, M.W.P. Savelsbergh .1999. *A computational study of search strategies for mixed integer programming*. INFORMS J. Comput11, 173.

[15] J. Pryor, J.W. Chinneck*.* 2011*. Faster integer-feasibility in mixed-integer linear programs by branching to force change*. Comput Oper. Res, 1143-1152.

[16] M. Fischetti, M. Monaci*, Backdoor branching, in: O. G¨unl¨uk, G.J. Woeginger (Eds.). 2011. Integer Programming and Combinatoral Optimization,* in: Lecture Notes in Computer Science, vol. 6655, Springer, Berlin, Heidelberg, 183-191.

[17] A. Gilpin, T. Sandholm. 2011. Information-theoretic approaches to branching in search. Discrete Optim., 147–159.

[18] Elias Munapo. 2020.Improvement of the Branch and Bound Algorithm for Solving the Knapsack Linear Integer Problem .Eastern-European Journal of Enterprise Technologies2(4 (104)), 59-69.

[19] Lingying Huang, Xiaomeng Chen, Wei Huo, Jiazheng, Wang Fan Zhang, Bo Bai, Ling Shi.2021. *Branch and Bound in Mixed Integer Linear Programming Problems: A Survey* of *Techniques and Trends.* Preprint submitted to Discrete Optimization*,* https://arxiv.org/abs/2111.06257.