é-traitement12

Université Clermont Auvergne

Rapport

Pré-traitement d'un graphe pour la résolution du problème MSTC

Réalisé par :
Katia MEDJANI

Table des matières

1 Introduction

Nous étudions des approches pour la solution du problème MSTC. Étant donné le graphe G (V, E) et un ensemble $C \in$ E x E de paires d'arêtes en conflit, le problème consiste à trouver un arbre de recouvrement minimal sans conflit, c'est-à-dire que les solutions possibles sont autorisées à inclure au plus une des arêtes de chaque paire en C. Dans cet article, nous nous appuyons sur la représentation des contraintes de conflit en utilisant un graphe de conflit auxiliaire $\hat{G}$(E, C), où les ensembles stables correspondent à des sous-ensembles de E sans conflit. Nous introduisons une méthode générale de pré-traitement et un algorithme.
Pour améliorer le processus de solution, Il est envisageable d'utiliser des conditions de faisabilité spécifiques à un problème MSTC pour concevoir l'algorithme de pré-traitement. La figure 1 décrit la méthode globale, que je décris ensuite. Il convient de noter que, l'algorithme suivant peut être utilisé et être intégré à toute solution technique au problème.

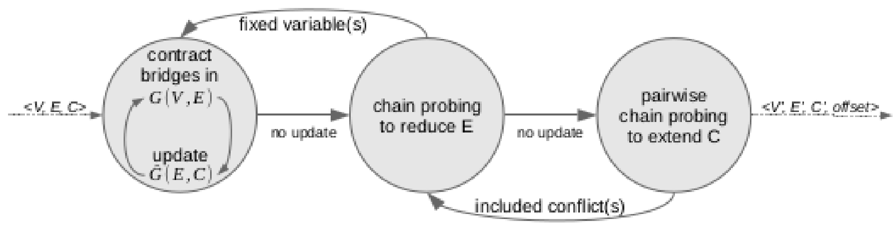

FIGURE 1 – Caption

1. Une contraction dans la première étape des arêtes isthme (bridges) et suppréssion des arêtes qui sont en conflit avec.
2. une suppression d'arête e tel que $\delta_{\hat{G}}(e) > 0$, si la suppression de ses arêtes en conflits déconnecte le graphe, puis mise à jour de C.
3. Choisir e_1, e_2 tel que $\{e_1, e_2\} \notin C$, si la suppression de leurs conflits déconnecte le graphe alors ajouter $\{e_1, e_2\} \in C$.

2 définition générale

Pour une meilleur compréhension du problème je présente dans ce document quelque définitions . Étant donné un graphe G(V, E) et un ensemble C(E $\times$ E) de paires

d'arêtes en conflit, le problème consiste à trouver un arbre couvrant minimal sans conflit (MST) : un arbre recouvrant de G, de coût minimal, qui comprend au plus un de e_i ou e_k pour chaque paire $\{e_i, e_k\} \in C$. Une définition équivalente exploité ici utilise le concept de graphe de conflit $\hat{G}$(E, C) : en désignant chaque arête du graphe d'origine comme un nœud dans $\hat{G}$, chaque contrainte de conflit est représenté par une arête reliant les nœuds correspondants dans $\hat{G}$. Le problème est donc de trouver un sous-ensemble de E de coût minimum, correspondant à la fois à un arbre couvrant de G que je noterai G(T) et à un ensemble stable dans $\hat{G}$ que je noterai $\hat{G}(S)$.
je défini ci-dessous quelque autres définitions qui vont nous permettre de bien comprendre les formules de ce document.

- isthme : Un isthme ou un pont est, en théorie des graphes, une arête d'un graphe dont l'élimination induit un graphe avec plus de composantes connexes que le graphe initial. De façon équivalente, une arête est un isthme si et seulement si elle n'est pas contenue dans un cycle.
  Nous pouvons aussi définir un isthme par une arête e_i tel que $G \setminus \{e_i\}$ n'est pas connexe.
- $\chi(e_i)$ est l'ensemble des arête qui sont en conflits avec e_i
- C_e le poids associé à l'arête e.

3 première phase de pré-traitement

La première phase vérifie les arêtes isthme dans G, en utilisant la recherche en profondeur d'abord. Tant que le graphe d'origine est connecté, toute isthme est contractée, son coût C_{e1} est ajouté en tant que décalage à la valeur optimale du problème réduit (le cas échéant) et les paires en conflit sont supprimées de G et de $\hat{G}$, c'est-à-dire que nous pouvons fixer des variables correspondant à e_k à zéro, pour tout $e_k \in E$ tel que $\{e_1, e_k\} \in C$.
Si nous vérifions à tout moment que G n'est pas connecté, le problème initial est irréalisable.
D'autre part, si le graphique obtenu est un arbre sans conflit, il constitue également l'unique solution envisageable au problème.
Une définition équivalente du problème est comme la suivante :

Soit G=(V,E) un graphe et $\hat{G} = (E, C)$ le graphe de conflits associé à G.
Si $e_i \in E$ est un isthme alors :

1. Contracter e_i.
2. $E' = E/\chi(e_i)$ dans G et dans $\hat{G}$

3. retourner à l'étape 01.

Si G n'est pas connexe alors pas de solution.

D'où on peut découler les corollaires suivants :

Corollaire 1 *Toute arête définissant un isthme du graphe est inclut dans la solution du problème MSTC.*

Corollaire 2 *Toute arête qui est en conflit avec un isthme du graphe n'est pas incluse à la solution du problème MSTC.*

4 Deuxième phase de pré-traitement

Dans cette phase, on vérifie la connectivité des sous-graphes de G incluant une arête donné e (avec degré dans le graphe de conflit $\delta\hat{G}(e) > 0$. Si la chaîne supprimant les paires en conflit et corrigeant les arêtes éventuellement impliquées par la sélection conduit à un graphe déconnecté, nous pouvons supprimer e de E et les conflits correspondants de C. Dans ce cas, nous revenons à la première phase car G pourrait inclure de nouveaux isthme.
Nous pouvons Traduire cette phase de la manière suivante :

Soit $e_i \in E$ Telque , $\delta_{\hat{G}}(e_i) > 0$.
$\forall e_j \in \chi(e_i)$, Supprimer e_i de G SSI $G = (V, E')$ est non connexe avec $E' = E/\chi(e_i)$
Mettre à jour C.

Démonstration par l'absurde
Supposant que l'arête e_i appartient à la solution du problème, cela signifie que $x_{e_i} = 1$, mais on sais que si la solution inclut x_{e_i} donc on doit supprimer toutes les arêtes qui y sont en conflit, $e_j = 0 \; \forall e_j \in \chi(e_i)$;
Or $G \setminus \chi(e_i)$ est non connexe.
contradiction avec $G(T)$ est un arbre.

5 Troisième phase de pré-traitement

une troisième phase effectue une évaluation similaire sur la connectivité de G, en examinant maintenant des paires d'arêtes. On commence à fixer les arêtes de solu-

tion e_1 et e_2 (qui ne sont pas en conflit), et procède en supprimant les arêtes qui sont en conflit avec elles et en incluant toutes les arêtes de coupe impliquées par la sélection. Maintenant, si G devenait déconnecté, la nouvelle paire de conflits (e_1, e_2) est incluse dans C, et nous pourrions revenir à la deuxième phase pour vérifier s'il est possible de supprimer d'autres arêtes.
Notez que, si un arbre couvrant de G est obtenu sans conflit pendant les deux dernières phases, nous pouvons le garder en tant que solution réalisable primaire.
On peut aussi traduire cette phase de la manière suivante :

Soit $e_i, e_j \in E$ Tel que $\{e_i, e_j\} \notin C$.
Soit $\chi(e_i)$ et $\chi(e_j)$ l'ensemble des arêtes en conflit avec e_i et e_j respectivement.
Supposant que G$\backslash(\chi(e_i) \cup \chi(e_j))$ n'est pas connexe, alors ajouter $\{e_i, e_i\} \in C$

Démonstration par l'absurde
Soit $e_i, e_j \in E$ TQ $\{e_i, e_j\} \notin C$.
Si G$\backslash(\chi(e_i) \cup \chi(e_j))$ n'est pas connexe.
Mettant :

— $x_{e_i} = 1 \quad \Rightarrow x_e = 0, \forall e \in \chi(e_i)$

— $x_{e_j} = 1 \quad \Rightarrow x_e = 0, \forall e \in \chi(e_j)$

OR G$\backslash(\chi(e_i) \cup \chi(e_j))$ n'est pas connexe.

Contradiction
$G(T)$ est un arbre.

5.1 généralisation de la phase 03

Nous voulons maintenant généraliser l'étape précédente à trois arêtes, Soit $e_i, e_j, e_k \in E$ ne sont pas deux à deux en conflits
Supposant que G$\backslash(\chi(e_i) \cup \chi(e_j) \cup \chi(e_k))$ n'est pas connexe.

— Il suffit dans ce cas de choisir deux arêtes parmi les trois pour les mettre en conflits, le choix des deux arêtes dépend du poids de .

6 Algorithme de pré-traitement

Nous présentons dans ce chapitre le squelette de l'algorithme de pré-traitement du problème MSTC

Étape 01

For $e_{ij} \in E(G)$
  G'=G$\backslash e_{ij}$
  if ($\exists$ chemin de i à j)
    "e_{ij} n'est pas un isthme".
  if
    G=G$\backslash\{e_{ij}\}$
    Supprimer les sommets i et j
    créer un sommet v qui a comme voisin les voisins de i et les voisins de j
    for e in $\chi(e_{ij})$
      G'=G$\backslash e_{ij}$
      supprimer e dans G
      G'=G$\backslash e_{ij}$
      supprimer e dans $\hat{G}$
      mettre a jours C
    end
  end
end

Étape 02

for e_i in E
  if (G $\backslash\chi(e_i)$ est non connexe)
    $E' = E \setminus \{e_i\}$
    Mettre à jour G
    Mettre à jour $\hat{G}$
    Mettre à jour C
    Aller à l'étape 01
  end
end

Étape 03

for e_i,e_j in E
  if ($\{e_i, e_j\} \notin C$ **and** G $\backslash(\chi(e_i) \cup \chi(e_j))$ est non connexe)
    C'=C$\cup\{e_i, e_j\}$
    Mettre à jour $\hat{G}$
    Aller à l'étape 02
  end
end

7 Conclusion

Ce travail contribue à une approche de pré-traitement au problème MSTCC. Nous présentons un algorithme général de pré-traitement basé sur les implications des conditions de faisabilité, qui pourrait être intégré à différentes méthodologies pour le problème.

Des recherches ultérieures pourraient élaborer sur une analyse spécifique du polytope MSTCC et éventuellement indiquer la forme des inégalités valides, qu'elles soient également valables pour la relaxation de l'ensemble stable ou non. Il serait intéressant de savoir dans quelles conditions les inégalités de trous irréguliers et de cliques maximales sont également déterminantes pour le nouveau polytope. Bien que ces questions soient intéressantes en elles-mêmes, l'approche algorithmique peut nécessiter des formulations améliorées et la mise à profit de techniques plus établies de la littérature des ensembles stables. Nous cherchons également à mieux comprendre la dureté et la facilité de résolution des problèmes connexes sous des contraintes disjonctives, telles que celles décrites par Darmann et al. (2011).

Bibliographie

[1] Phillippe Samer · Sebastián Urrutia, A branch and cut algorithm for minimum spanning trees under conflict constraints, Manuscript accepted for publication on Optimization Letters The final publication is available at Springer via http ://dx.doi.org/10.1007/s11590-014-0750-x.