Algoritmos *branch-and-cut* para o Problema da Árvore Geradora de Custo Mínimo com Conflitos

Trabalho Prático 2 - MO420

Allan Sapucaia Barboza - RA 134796
Luis Henrique Pauleti Mendes - RA 117801

Junho de 2018

1 Introdução

Neste trabalho foi estudado o Problema da Árvore Geradora de Custo Mínimo com Conflitos (AGMCC) e implementado algoritmos *branch-and-cut* para resolvê-lo. Seja um grafo $G = (V, E)$ não direcionado, com $|V| = n$ e $|E| = m$. A cada aresta e de E está associado um custo c_e. Além disso, é dado um conjunto C de pares de arestas, chamado de *conjunto de conflito*, com $|C| = p$. Cada elemento de $\{e, f\}$ em C é chamado de um *par (de arestas) conflitante*. Uma árvore geradora T de G é dita ser livre de conflitos se nenhum par de arestas em T é conflitante. No AGMCC o que se deseja é encontrar uma árvore geradora de custo mínimo que seja livre de conflitos.

Dados G e o conjunto de conflitos $C \subset E \times E$, define-se o *grafo de conflitos* $\hat{G}$ cujos vértices correspondem ao conjunto de arestas do grafo G e cujas arestas são exatamente aquelas associadas a pares de conflitos presentes em C (i.e., $(e, f) \in E(\hat{G})$ *se e somente se* $(e, f) \in C$). Seja $x \in \mathbb{B}^m$ um vetor que representa uma solução do AGMCC e P_{tree} e P_{stab} respectivamente, as envoltórias convexas das árvores geradoras mínimas de G e dos conjuntos independentes (ou estáveis) de $\hat{G}$. Tem-se então que $x \in P_{tree} \cap P_{stab}$.

2 Modelagem PLI do AGMCC

Pode-se formular o AGMCC usando modelos PLI conhecidos para os problemas da árvore geradora mínima e do conjunto independente mínimo em grafos. Usando o vetor x definido anteriormente, um modelo para árvores geradoras seria:

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \forall S \subset V, S \neq \emptyset \quad (1)$$

$$\sum_{e \in E(S)} x_e = n - 1 \quad (2)$$

$$x_e \in \mathbb{B}, \forall e \in E \tag{3}$$

Usando o mesmo vetor binário x, um modelo simples para conjuntos independentes seria:

$$x_e + x_f \leq 1, \forall (e,f) \in C \tag{4}$$

Estes modelos são apresentados na seção 3.1 de Samer e Urrutia (2015). A partir deles, uma formulação para o AGMCC seria

$$\min\{\sum_{e \in E} c_e x_e : x \text{ satisfaz a (1), (2), (3) e (4)}\} \tag{5}$$

Um resultado conhecido é que o sistema linear composto pelas inequações (1) e (2) e as relaxações lineares de (3) descreve completamente a envoltória convexa de P_{tree}. No caso de P_{stab}, a seção 3.1 de Samer e Urrutia (2015) exibe duas desigualdades válidas conhecidas na literatura. São elas: a *desigualdade do ciclo impar* (OCI) (equação (5) de Samer e Urrutia (2015)) e a *desigualdade clique* (CLIQUE) (equação (6) de Samer e Urrutia (2015)). Repare que o número de desigualdades OCI e CLIQUE são em número exponencial em m e, da mesma forma, as desigualdades (1) são em número exponencial em n. Portanto, mesmo sem fazer uso das desigualdades OCI e CLIQUE, a única forma de usar a formulação para resolver o AGMCC é mediante um algoritmo *branch-and-cut*.

Em relação às desigualdades CLIQUE, sabe-se que elas só definem *facets* se o grafo suporte das mesmas for uma clique maximal. Assim, quando uma aresta (e,f) de $\hat{G}$ está em uma clique maximal Q de tamanho maior que 2 deste grafo, a restrição correspondente a (e,f) em (4) é dominada pela desigualdade CLIQUE associada a Q. Seja $\{Q_1, Q_2, \cdots, Q_q\}$ uma *cobertura de arestas* de $\hat{G}$ por *cliques maximais*, uma nova formulação para o AGMCC é obtida substituindo-se as restrições (4) pelas desigualdades CLIQUE associadas a $Q_1, Q_2, \cdots, Q_q$, ou seja:

$$\min\{\sum_{e \in E} c_e x_e : x \text{ satisfaz a (1), (2), (3) e } \sum_{g \in Q_i} x_g \leq 1, \forall i = 1, \cdots, q\} \tag{6}$$

3 Implementação

Nesta seção serão discutidos os detalhes de implementação do projeto, os algoritmos escolhidos e as decisões tomadas.

3.1 Preprocessamento

Foi visto no trabalho anterior que boas rotinas de preprocessamento praticamente resolviam as instâncias do tipo2. Assim, as rotinas de preprocessamento foram reaproveitadas e incluídas na resolução de todas as instâncias. Os dois tipos de preprocessamento são explicados a seguir.

Para o AGMCC, a fixação de variáveis, que correspondem a arestas, em 1 implica na remoção das arestas conflitantes e dos conflitos correspondentes. A fixação em 0 faz com que conflitos envolvendo aquela variável sejam removidos por não poderem mais ser violados. Portanto, fixação de variáveis podem reduzir consideravelmente o modelo. As heurísticas são aplicadas repetidamente até que não ocorra nenhuma mudança.

Neste trabalho é proposta uma técnica de preprocessamento baseada em otimalidade, e é adaptada a técnica baseada em viabilidade estudada em Samer e Urrutia (2015).

O preprocessamento por otimidade OPTFIX é uma modificação do algoritmo de Kruskal, assumindo que o conjunto de arestas possíveis seja $E' \subseteq E$, e estejam ordenada por custo crescendo, onde o desempate é o número crescente de conflitos, o algoritmo de Kruskal se baseia na ideia que fixar uma aresta leve que não gera ciclo e não piora o custo da solução final. A heurística OPTFIX fixa sucessivamente as arestas leves que não geram ciclos ou tem conflitos em 1. Isso implica que arestas que formam ciclos com a floresta em construção são fixadas em 0. Essa fixação permite resolver alguns conflitos. O procedimento para quando não existem arestas leves sem conflitos para serem fixadas. A corretude segue diretamente da corretude do algoritmo de Kruskal (CORMEN et al., 2009). A complexidade é a mesma do algoritmo de Kruskal.

O preprocessamento por viabilidade BRIDGEFIX é apresentando em Samer e Urrutia (2015). A ideia central é que se a remoção de uma aresta faz com que o grafo se torne desconexo, a variável correspondente deve ser fixada em 1. Da mesma forma, se a fixação de uma aresta implica, pelos seus conflitos, que o grafo se torna desconexo, a variável correspondente é fixada em 0. A fixação em 1 é testada pelo algoritmo de Tarjan (HALIM; HALIM, 2013) cuja complexidade é $O(n+m)$. A segunda parte, exige que para cada aresta uma busca em profundidade seja executada, resultando na complexidade $O(m(n+m))$, que domina o custo anterior, resultando numa complexidade total de $O(m(n+m))$. Uma segunda etapa envolve tentar fixar em 1 arestas e verificar sua implicações: a fixação de uma aresta implica que suas arestas conflitantes são zeradas e, além disso, surgem novas pontes. Se algumas dessas implicações é um grafo desconexo, a aresta é fixada em 0. As etapas são repetidas sucessivamente até que não haja mudança. No pior dos casos, a complexidade é $O(m^2(n+m))$.

3.2 Rotina de Separação das Desigualdades (1)

A separação das desigualdades de Subtour garante que o gráfo é conexo. E ela é feita resolvendo um problema de Fluxo Máximo (NEMHAUSER; WOLSEY, 1988).

Seja $H = (V, E)$ um grafo com o mesmo conjunto de vértices e arestas do grafo original $G = (V, E)$ e seja x_e a solução do PLI. A cada aresta $e \in E$ associamos a capacidade x_e. Se o corte mínimo em H é menor que 1, então o corte separa V em dois componentes U e U' não-conectados e podemos criar uma restrição para os mesmos, escolhendo apropriadamente $W = U$ ou $W = U'$ de forma que W seja o componente com mais arestas internas do que a restrição permite.

Esta separação é feita executando-se o algoritmo de Dinic, cuja complexidade é $O(n^2m)$, $n-1$ vezes. Em cada execução o vértice de origem é fixo, mas cada outro vértice é o destino uma vez. O algoritmo de Dinic implementado neste trabalho é uma modificação do código disponível em Singh (2018). A separação é feita em tempo $O(n^3m)$, logo as relaxações são resolvidas em tempo polinomial.

3.3 Rotina de Separação das Desigualdades OCI

Neste trabalho foi implementada uma rotina de separação das desigualdades OCI conforme descrito em Samer e Urrutia (2015).

Seja $x^\star$ a solução fracionária atual para o AGMCC, define-se uma nova função de custo c para adjacências no grafo de conflitos $\hat{G} = (E, C)$: $c(e, f) = \dfrac{(1 - x^\star_e - x^\star_f)}{2}$ para cada $\{e, f\} \in C$. Visto que as restrições (4) e as relaxações lineares das restrições (3) $(0 \leq x_e \leq 1, \forall e \in E)$ são satisfeitas por $x^\star$, tem-se que $c : C \to [0, \frac{1}{2}]$. Também, constrói-se um grafo bipartido auxiliar H duplicando-se $\hat{G}$ da seguinte maneira: H possui dois vértices e^+ e e^- para cada $e \in E$, assim como arestas $\{e^+, f^-\}$ e $\{e^-, f^+\}$ para cada $\{e, f\} \in C$, ambas com custo $c(e, f)$.

Então, para cada $e \in E$, computa-se um caminho (e^+, e^-) mínimo no grafo auxiliar H. Como a função de custo c é não-negativa, foi utilizou-se o algoritmo de Dijkstra, interrompendo-se sua execução assim que o vértice destino e^- é selecionado. Pela construção de H, os vértices e^+ e e^- estão em partições distintas, implicando que caminho tem comprimento ímpar. Omitindo-se os índices $^+$ e $^-$, o caminho corresponde a um passeio ímpar fechado em $\hat{G}$. Entretanto, este passeio pode incluir vértices e arestas repetidos, uma vez que o caminho mínimo é determinado em H. Um ciclo ímpar é possivelmente recuperado após remover tais repetições, inspecionando-se os vértices em sequência.

O custo de qualquer tal ciclo ímpar $U \subset C$ no grafo de conflitos, desconsiderando quaisquer cordas que ele possa ter, é $c(U) = \sum_{\{e,f\} \in U} c(e, f) = \sum_{\{e,f\} \in U} \dfrac{(1 - x^\star_e - x^\star_f)}{2}$ $= \dfrac{|U|}{2} - \dfrac{1}{2} \sum_{\{e,f\} \in U} (x^\star_e + x^\star_f) = \dfrac{|U|}{2} - \sum_{e \in V(U)} x^\star_e$, onde $V(U) \subseteq E$ denota o conjunto de vértices induzido por U.

Isto é, $\sum_{e \in V(U)} x^\star_e = \dfrac{|U|}{2} - c(U)$, implicando que $x^\star$ viola a inequação de ciclo ímpar correspondente $\sum_{e \in V(U)} x_e \leq \dfrac{|U| - 1}{2} = \dfrac{|U|}{2} - \dfrac{1}{2}$ se e somente se $c(U) \leq \dfrac{1}{2}$.

Pela construção de H, temos que ele possui $2m$ vértices e $2p$ arestas. Como a fila de prioridade do algoritmo de Dijkstra implementado neste trabalho, que é uma modificação do código disponível em Agrawal (2018), foi implementada utilizando-se um heap binário, temos que cada execução do algoritmo de Dijkstra em H tem complexidade $O(p \log m)$ (CORMEN et al., 2009). Como são computados m caminhos mínimos, tem-se que a rotina de separação das desigualdade OCI tem complexidade $O(mp \log m)$.

3.4 Rotina para construir Cobertura de Cliques

Neste trabalho foi implementada uma rotina para construir uma cobertura de cliques para as arestas de $\hat{G}$ conforme descrito em Nemhauser e Sigismondi (1992).

Para cada aresta $\{e, f\} \in C$ que ainda não foi coberta por nenhuma clique, cria-se uma clique parcial $Q = \{e, f\}$, que é então expandida enquanto for possível, isto é, até se tornar maximal. Para vértice $g \in Q$, verifica-se se existe algum $h \in N_{\hat{G}}[g]$ que pode ser adicionado à Q, onde $N_{\hat{G}}[g]$ denota o conjunto de vértices adjacente à g em $\hat{G}$. O vértice h pode ser adicionado à Q ele for adjacente a todos os vértices que já estão em Q, ou seja, se existe $\{h, i\} \in C$ para todo $i \in Q$. Neste caso, h é adicionado à Q e todas as arestas $\{h, i\} \in C$ para todo $i \in Q \setminus \{h\}$ são marcadas como cobertas. A rotina para quando todas as arestas $\{e, f\} \in C$ são cobertas por pelo menos uma clique maximal.

A cada iteração da expansão de Q, são analisados $|Q|$ vértices em Q e, para cada vértice em Q, são analisados no máximo m vértices de $\hat{G}$ e, por sua vez, para cada vértice em $\hat{G}$, são analisados $|Q|$ vértices em Q. Como na primeira iteração tem-se que $|Q| = 2$ e na última iteração tem-se, no pior caso, que $|Q| = m - 1$, tem-se que a complexidade da rotina para construir cobertura de cliques é $O(\sum_{j=1}^{m-1} mj^2) = O(m^4)$.

3.5 Heurística Primal

Neste trabalho foi projetada e implementada a heurística primal descrita a seguir, a qual foi executada após a conclusão do algoritmo de planos de corte em cada nó da árvore de enumeração.

Seja $x^\star$ a solução fracionária atual para o AGMCC e sejam $E^0 \subset E$ e $E^1 \subset E$ os subconjuntos de arestas que cujas variávies foram fixadas em 0 e em 1 no preprocessamento, respectivamente. Cria-se uma solução parcial T contendo apenas as arestas em E^1, ou seja, faz-se $T = E^1$. Percorre-se as arestas $e \in E$ em ordem não-crescente de $x^\star_e$ e adiciona-se e a T se $e \notin E^0$ e e não criar nenhum ciclo nem nenhum conflito em T, ou seja, se $e \notin E^0$ e $T' = T \cup \{e\}$ for acíclica e livre de conflitos, faz-se $T = T'$.

Se, ao final deste procedimento, T for conexa, tem-se que T é uma solução primal viável para o AGMCC, pois T é conexa, acíclica e livre de conflitos. Então a heurística primal para e a solução primal viável T é retornada.

Se T não for conexa, percorre-se novamente as arestas de $e \in E$ em ordem não-crescente de $x^\star_e$ e adiciona-se e a T se $e \notin E^0$ e e não criar nenhum ciclo em T, ou seja, se $e \notin E^0$ e $T' = T \cup \{e\}$ for acíclica, faz-se $T = T'$. Ao final deste procedimento, T é acíclica e conexa, porém não é livre de conflitos.

Então, remove-se de T toda aresta $e \in T$ que não está fixada e está envolvida em algum conflito, ou seja, seja $T' = \{e \in T \setminus E^1 : \exists\{e, f\} \in C \text{ para algum } f \in T\}$ faz-se $T = T \setminus T'$. Ao final deste procedimento, T é acíclica e livre de conflitos, porém não é conexa.

Por fim, seguindo a ideia do algoritmo de Kruskal para árvore geradora mínima (CORMEN et al., 2009), percorre-se as arestas $e \in E$ em ordem não-decrescente de custo c_e e adiciona-se e a T se $e \notin E^0$ e e não criar nenhum ciclo nem nenhum conflito em T, ou seja, se $e \notin E^0$ e $T' = T \cup \{e\}$ for acíclica e livre de conflitos, faz-se $T = T'$.

Se, ao final deste procedimento, T for conexa, tem-se que T é uma solução primal viável para o AGMCC, pois T é conexa, acíclica e livre de conflitos. Então a heurística primal para e a solução primal viável T é retornada Se T não for conexa, a heurística primal para, sem retornar nenhuma solução primal viável.

As ordenações dos vértices em E tem complexidade $O(m \log m)$. Para cada $e \in E$, verificar se $e \in E^0$ tem complexidade $O(|E^0| \log |E^0|)$. Como, $|E^0| \leq m$, tem-se que cada verificação tem complexidade $O(m \log m)$.

Para verificar se a adição de e cria um conflito em T, verifica-se se existe alguma aresta $f \in T$ tal que $\{e, f\} \in C$. Tal verificação tem complexidade $O(|T| \log p)$. Como $|T| < n$, tem-se que a verificação tem complexidade $O(n \log p)$.

Para verificar se a adição de e cria um ciclo em T, utilizou-se o procedimento FIND-SET descrito em Cormen et al. (2009). Note que a complexidade desta verificação é dominada pelas outras.

Logo, como cada verificação é realizada $O(m)$ vezes, tem-se que a complexidade da heurística primal é $O(m^2 \log m + nm \log p)$.

4 Resultados

Os experimentos computacionais foram executados em uma máquina com processador Intel® Pentium(R) CPU G4560 @ 3.50GHz x 4, com 8GB de memória RAM. Cada instância teve um tempo máximo de execução de 10 minutos no CPLEX para resolução do PLI, desconsiderando o tempo de construção do problema e de preprocessamento. No entando, o tempo total exibido leva em conta o preprocessamento.

Todos os experimentos foram executados para modelos Branch-and-Cut com preprocessamento, variando algumas configurações. As configurações podem usar ou não as desigualdades OCI; representar conflitos por pares ou cliques; e usar ou não a heurística primal proposta. Foram montadas 5 configurações, cujos resultados experimentais são mostrados a seguir.

Cada tabela apresenta 11 colunas, nome da instância(Inst), número de cortes do tipo SubTour(1)(**STC**), número de cortes do tipo OCI(**OCIC**), limitante dual da primeira relaxação(**FOrlx**), limitante dual na raíz(**FOroot**), número de nós explorados(**Nós**), nó onde a melhor solução inteira foi encontrada(**NoInt**), melhor limitante primal (**Prim**), melhor limitante dual (**Dual**), tempo total de execução(**Time**), e número de variáveis fixadas(**Fix**).

A primeira configuração BASE, **sem OCI, conflitos por pares e sem heurística** foi tomada como referência, os resultados são exibidos na Tabela 1.

A segunda configuração OCI, **com OCI, conflitos por pares e sem heurística**, foi utilizada para medir o impacto das desigualdades OCI. Os resultados são exibidos na Tabela 2.

Analisando-se a Tabela 2 pode-se verificar que a configuração OCI explorou menos nós da árvore de enumeração e obteve limitantes duais melhores do que a configuração BASE, porém obteve um tempo de execução maior e limitantes primais piores do que a configuração BASE.

A configuração CLIQUE, **sem OCI, conflitos por clique e sem heurística**, foi utilizada para medir o impacto dos conflitos por clique. Os resultados são exibidos na Tabela 3.

Tabela 1: Resultados para a configuração BASE

Inst	STC	OCIC	FOrlx	FOraiz	Nós	NoInt	Prim	Dual	Time	Fix
t1/z50-200-199	49	0	612.00	612.00	26	20	708	708.00	0.45	0
t1/z50-200-398	47	0	652.00	652.00	20	4	770	770.00	0.09	0
t1/z50-200-597	114	0	724.00	838.00	138	122	917	917.00	0.41	0
t1/z50-200-995	159	0	943.00	956.00	2940	2939	1324	1324.00	6.21	0
t1/z100-300-448	140	0	3440.00	3440.00	71	64	4041	4041.00	0.66	0
t1/z100-300-897	341	0	4637.00	4877.00	5570	5551	5658	5658.00	55.65	1
t1/z100-500-1247	80	0	3454.00	3454.00	24	22	4275	4275.00	0.46	0
t1/z100-500-2495	595	0	4844.00	5230.00	27799	13862	5997	5997.00	393.18	0
t1/z100-500-3741	649	0	5011.50	5398.75	39354	5952	9294	5961.50	600.03	0
t1/z200-600-1797	1138	0	12148.84	12433.00	6528	6177	15210	12803.00	600.06	0
t1/z200-800-3196	993	0	18415.67	19625.75	5794	4871	21781	20419.88	600.05	0
t2/z50-200-3903	1	0	1634.00	1634.00	0	0	1636	1636.00	0.06	162
t2/z50-200-4877	0	0	2043.00	2043.00	0	0	2043	2043.00	0.04	169
t2/z50-200-5864	0	0	2338.00	2338.00	0	0	2338	2338.00	0.05	177
t2/z100-300-8609	0	0	7434.00	7434.00	0	0	7434	7434.00	0.08	300
t2/z100-300-10686	0	0	7968.00	7968.00	0	0	7968	7968.00	0.08	300
t2/z100-300-12761	0	0	8166.00	8166.00	0	0	8166	8166.00	0.10	300
t2/z100-500-24740	100	0	6165.50	6442.50	601	571	12652	12652.00	34.23	68
t2/z100-500-30886	0	0	11232.00	11232.00	0	0	11232	11232.00	0.28	500
t2/z100-500-36827	2	0	11457.00	11457.00	0	0	11481	11481.00	0.37	472
t2/z200-400-13660	5	0	17680.00	17680.00	0	0	17728	17728.00	0.31	374
t2/z200-400-17089	0	0	18617.00	18617.00	0	0	18617	18617.00	0.28	388
t2/z200-400-20469	0	0	19140.00	19140.00	0	0	19140	19140.00	0.29	400
t2/z200-600-34504	0	0	20716.00	20716.00	0	0	20716	20716.00	0.43	600
t2/z200-600-42860	0	0	18025.00	18025.00	0	0	18025	18025.00	0.49	600
t2/z200-600-50984	0	0	20864.00	20864.00	0	0	20864	20864.00	0.58	600
t2/z200-800-62625	0	0	39895.00	39895.00	0	0	39895	39895.00	0.88	800
t2/z200-800-78387	0	0	37671.00	37671.00	0	0	37671	37671.00	1.06	800
t2/z200-800-93978	0	0	38798.00	38798.00	0	0	38798	38798.00	1.06	800
t2/z300-600-31000	0	0	43721.00	43721.00	0	0	43721	43721.00	0.60	600
t2/z300-600-38216	1	0	44260.00	44260.00	0	0	44267	44267.00	1.02	571
t2/z300-600-45310	0	0	43071.00	43071.00	0	0	43071	43071.00	0.97	600
t2/z300-800-59600	0	0	43125.00	43125.00	0	0	43125	43125.00	0.89	800
t2/z300-800-74500	0	0	42292.00	42292.00	0	0	42292	42292.00	1.17	800
t2/z300-800-89300	0	0	44114.00	44114.00	0	0	44114	44114.00	1.37	800
t2/z300-1000-96590	0	0	71562.00	71562.00	0	0	71562	71562.00	1.59	1000
t2/z300-1000-120500	0	0	76345.00	76345.00	0	0	76345	76345.00	1.68	1000
t2/z300-1000-144090	0	0	78880.00	78880.00	0	0	78880	78880.00	2.09	1000

Analisando-se a Tabela 3 pode-se verificar que a configuração CLIQUE obteve limitantes duais iniciais (**FOrlx** e **FOroot**) melhores do que a configuração BASE. Também verifica-se que a configuração CLIQUE explora mais nós da árvore de enumeração e tem um tempo de execução menor do que a configuração BASE, ou seja, o tempo gasto em cada nó é muito inferior. Isto deve-se ao fato de a formulação (6), utilizada na configuração CLIQUE, possuir menos restrições do que a formulação (5), utilizada na configuração

Tabela 2: Resultados para a configuração OCI

Inst	STC	OCIC	FOrlx	FOraiz	Nós	NoInt	Prim	Dual	Time	Fix
t1/z50-200-199	49	0	612.00	612.00	26	20	708	708.00	0.45	0
t1/z50-200-398	44	2	652.00	652.00	24	8	770	770.00	1.25	0
t1/z50-200-597	89	95	724.00	856.67	88	80	917	917.00	6.47	0
t1/z50-200-995	124	4486	943.00	1174.71	354	306	1324	1324.00	77.67	0
t1/z100-300-448	140	0	3440.00	3440.00	71	64	4041	4041.00	6.54	0
t1/z100-300-897	278	12442	4637.00	5179.33	1168	974	5658	5505.95	600.02	1
t1/z100-500-1247	87	5	3454.00	3454.00	32	31	4275	4275.00	14.00	0
t1/z100-500-2495	311	4659	4844.00	5569.38	334	276	6344	5678.25	601.18	0
t1/z100-500-3741	69	6981	5011.50	6280.08	112	-1	∞	6300.32	600.03	0
t1/z200-600-1797	284	7620	12148.84	12895.33	228	124	16974	12986.58	600.22	0
t1/z200-800-3196	206	4805	18415.67	20250.48	81	-1	∞	20349.35	600.69	0
t2/z50-200-3903	1	0	1634.00	1634.00	0	0	1636	1636.00	0.53	162
t2/z50-200-4877	0	0	2043.00	2043.00	0	0	2043	2043.00	0.35	169
t2/z50-200-5864	0	0	2338.00	2338.00	0	0	2338	2338.00	0.42	177
t2/z100-300-8609	0	0	7434.00	7434.00	0	0	7434	7434.00	0.98	300
t2/z100-300-10686	0	0	7968.00	7968.00	0	0	7968	7968.00	1.26	300
t2/z100-300-12761	0	0	8166.00	8166.00	0	0	8166	8166.00	1.50	300
t2/z100-500-24740	59	8825	6165.50	9164.33	25	-1	∞	9263.04	610.07	68
t2/z100-500-30886	0	0	11232.00	11232.00	0	0	11232	11232.00	7.94	500
t2/z100-500-36827	2	0	11457.00	11457.00	0	0	11481	11481.00	29.31	472
t2/z200-400-13660	5	0	17680.00	17680.00	0	0	17728	17728.00	13.33	374
t2/z200-400-17089	0	0	18617.00	18617.00	0	0	18617	18617.00	3.18	388
t2/z200-400-20469	0	0	19140.00	19140.00	0	0	19140	19140.00	4.01	400
t2/z200-600-34504	0	0	20716.00	20716.00	0	0	20716	20716.00	11.69	600
t2/z200-600-42860	0	0	18025.00	18025.00	0	0	18025	18025.00	15.56	600
t2/z200-600-50984	0	0	20864.00	20864.00	0	0	20864	20864.00	19.28	600
t2/z200-800-62625	0	0	39895.00	39895.00	0	0	39895	39895.00	32.48	800
t2/z200-800-78387	0	0	37671.00	37671.00	0	0	37671	37671.00	42.73	800
t2/z200-800-93978	0	0	38798.00	38798.00	0	0	38798	38798.00	53.00	800
t2/z300-600-31000	0	0	43721.00	43721.00	0	0	43721	43721.00	10.73	600
t2/z300-600-38216	1	0	44260.00	44260.00	0	0	44267	44267.00	27.93	571
t2/z300-600-45310	0	0	43071.00	43071.00	0	0	43071	43071.00	17.65	600
t2/z300-800-59600	0	0	43125.00	43125.00	0	0	43125	43125.00	32.08	800
t2/z300-800-74500	0	0	42292.00	42292.00	0	0	42292	42292.00	42.54	800
t2/z300-800-89300	0	0	44114.00	44114.00	0	0	44114	44114.00	52.75	800
t2/z300-1000-96590	0	0	71562.00	71562.00	0	0	71562	71562.00	71.04	1000
t2/z300-1000-120500	0	0	76345.00	76345.00	0	0	76345	76345.00	91.38	1000
t2/z300-1000-144090	0	0	78880.00	78880.00	0	0	78880	78880.00	112.61	1000

BASE, de modo que as soluções das relaxações lineares são muito mais rápidas de serem computadas.

O experimento OCICLIQUE, combinando as duas configurações anteriores, **com OCI, conflitos por clique, sem heurística**, foi utilizada para medir o impacto combinado das desigualdades OCI e os conflitos por clique. Os resultados são exibidos na tabela 4.

Tabela 3: Resultados para a configuração CLIQUE

Inst	STC	OCIC	FOrlx	FOraiz	Nós	NoInt	Prim	Dual	Time	Fix
t1/z50-200-199	57	0	612.00	612.00	38	31	708	708.00	0.08	0
t1/z50-200-398	38	0	652.00	652.00	11	5	770	770.00	0.05	0
t1/z50-200-597	121	0	726.00	726.00	180	167	917	917.00	0.47	0
t1/z50-200-995	168	0	998.80	1024.25	1386	1021	1324	1324.00	3.66	0
t1/z100-300-448	156	0	3440.00	3440.00	57	45	4041	4041.00	0.61	0
t1/z100-300-897	504	0	4658.33	4950.83	6802	6654	5658	5658.00	96.99	1
t1/z100-500-1247	135	0	3454.00	3454.00	48	45	4275	4275.00	0.76	0
t1/z100-500-2495	650	0	4892.50	5309.00	27699	22779	5997	5997.00	418.80	0
t1/z100-500-3741	531	0	5406.50	5787.00	29332	7907	8047	6395.50	600.04	0
t1/z200-600-1797	980	0	12180.71	12528.25	6827	5494	14704	13023.17	600.05	0
t1/z200-800-3196	1200	0	18475.00	19802.61	4906	4274	22410	20504.89	600.06	0
t2/z50-200-3903	1	0	1634.00	1634.00	0	0	1636	1636.00	0.11	162
t2/z50-200-4877	0	0	2043.00	2043.00	0	0	2043	2043.00	0.12	169
t2/z50-200-5864	0	0	2338.00	2338.00	0	0	2338	2338.00	0.18	177
t2/z100-300-8609	0	0	7434.00	7434.00	0	0	7434	7434.00	0.28	300
t2/z100-300-10686	0	0	7968.00	7968.00	0	0	7968	7968.00	0.41	300
t2/z100-300-12761	0	0	8166.00	8166.00	0	0	8166	8166.00	0.65	300
t2/z100-500-24740	27	0	11846.06	11921.36	21	6	12652	12652.00	11.59	68
t2/z100-500-30886	0	0	11232.00	11232.00	0	0	11232	11232.00	2.28	500
t2/z100-500-36827	2	0	11457.00	11457.00	0	0	11481	11481.00	3.55	472
t2/z200-400-13660	5	0	17680.00	17680.00	0	0	17728	17728.00	0.77	374
t2/z200-400-17089	0	0	18617.00	18617.00	0	0	18617	18617.00	1.10	388
t2/z200-400-20469	0	0	19140.00	19140.00	0	0	19140	19140.00	1.71	400
t2/z200-600-34504	0	0	20716.00	20716.00	0	0	20716	20716.00	2.21	600
t2/z200-600-42860	0	0	18025.00	18025.00	0	0	18025	18025.00	3.60	600
t2/z200-600-50984	0	0	20864.00	20864.00	0	0	20864	20864.00	5.85	600
t2/z200-800-62625	0	0	39895.00	39895.00	0	0	39895	39895.00	6.51	800
t2/z200-800-78387	0	0	37671.00	37671.00	0	0	37671	37671.00	11.54	800
t2/z200-800-93978	0	0	38798.00	38798.00	0	0	38798	38798.00	18.17	800
t2/z300-600-31000	0	0	43721.00	43721.00	0	0	43721	43721.00	2.24	600
t2/z300-600-38216	1	0	44260.00	44260.00	0	0	44267	44267.00	3.86	571
t2/z300-600-45310	0	0	43071.00	43071.00	0	0	43071	43071.00	5.92	600
t2/z300-800-59600	0	0	43125.00	43125.00	0	0	43125	43125.00	6.20	800
t2/z300-800-74500	0	0	42292.00	42292.00	0	0	42292	42292.00	11.13	800
t2/z300-800-89300	0	0	44114.00	44114.00	0	0	44114	44114.00	17.88	800
t2/z300-1000-96590	0	0	71562.00	71562.00	0	0	71562	71562.00	14.38	1000
t2/z300-1000-120500	0	0	76345.00	76345.00	0	0	76345	76345.00	25.62	1000
t2/z300-1000-144090	0	0	78880.00	78880.00	0	0	78880	78880.00	42.55	1000

Analisando-se a Tabela 4 pode-se verificar que a configuração OCICLIQUE explorou menos nós da árvore de enumeração, porém obteve um tempo de execução e limitantes primais piores do que a configuração CLIQUE.

No geral, nota-se que há um *trade-off* entre adicionar menos cortes, gastando-se menos tempo em cada nó da árvore de enumeração e, assim, explorando-se mais nós (configurações BASE e CLIQUE) e adicionar mais cortes, gastando-se mais tempo em cada nó da

Tabela 4: Resultados para a configuração OCICLIQUE

Inst	STC	OCIC	FOrlx	FOraiz	Nós	NoInt	Prim	Dual	Time	Fix
t1/z50-200-199	57	0	612.00	612.00	38	31	708	708.00	0.57	0
t1/z50-200-398	38	0	652.00	652.00	11	5	770	770.00	0.80	0
t1/z50-200-597	122	129	726.00	726.00	103	96	917	917.00	9.52	0
t1/z50-200-995	102	2799	998.80	1178.86	251	192	1324	1324.00	46.41	0
t1/z100-300-448	156	0	3440.00	3440.00	57	45	4041	4041.00	6.05	0
t1/z100-300-897	375	13823	4658.33	5186.45	1129	971	5676	5516.64	600.17	1
t1/z100-500-1247	82	1	3454.00	3454.00	12	11	4275	4275.00	7.93	0
t1/z100-500-2495	243	5492	4892.50	5561.61	288	235	6187	5686.62	600.28	0
t1/z100-500-3741	107	6897	5406.50	6281.26	132	-	∞	6313.54	602.30	0
t1/z200-600-1797	302	7223	12180.71	12897.74	242	-	∞	12966.76	600.25	0
t1/z200-800-3196	188	4855	18475.00	20287.10	80	-	∞	20481.82	600.05	0
t2/z50-200-3903	1	0	1634.00	1634.00	0	0	1636	1636.00	0.58	162
t2/z50-200-4877	0	0	2043.00	2043.00	0	0	2043	2043.00	0.43	169
t2/z50-200-5864	0	0	2338.00	2338.00	0	0	2338	2338.00	0.55	177
t2/z100-300-8609	0	0	7434.00	7434.00	0	0	7434	7434.00	1.16	300
t2/z100-300-10686	0	0	7968.00	7968.00	0	0	7968	7968.00	1.54	300
t2/z100-300-12761	0	0	8166.00	8166.00	0	0	8166	8166.00	2.07	300
t2/z100-500-24740	24	543	11846.06	12254.29	10	5	12652	12652.00	198.00	68
t2/z100-500-30886	0	0	11232.00	11232.00	0	0	11232	11232.00	9.96	500
t2/z100-500-36827	2	0	11457.00	11457.00	0	0	11481	11481.00	32.54	472
t2/z200-400-13660	5	0	17680.00	17680.00	0	0	17728	17728.00	13.92	374
t2/z200-400-17089	0	0	18617.00	18617.00	0	0	18617	18617.00	4.01	388
t2/z200-400-20469	0	0	19140.00	19140.00	0	0	19140	19140.00	5.51	400
t2/z200-600-34504	0	0	20716.00	20716.00	0	0	20716	20716.00	13.39	600
t2/z200-600-42860	0	0	18025.00	18025.00	0	0	18025	18025.00	18.67	600
t2/z200-600-50984	0	0	20864.00	20864.00	0	0	20864	20864.00	24.44	600
t2/z200-800-62625	0	0	39895.00	39895.00	0	0	39895	39895.00	38.11	800
t2/z200-800-78387	0	0	37671.00	37671.00	0	0	37671	37671.00	53.30	800
t2/z200-800-93978	0	0	38798.00	38798.00	0	0	38798	38798.00	70.14	800
t2/z300-600-31000	0	0	43721.00	43721.00	0	0	43721	43721.00	12.35	600
t2/z300-600-38216	1	0	44260.00	44260.00	0	0	44267	44267.00	30.77	571
t2/z300-600-45310	0	0	43071.00	43071.00	0	0	43071	43071.00	22.60	600
t2/z300-800-59600	0	0	43125.00	43125.00	0	0	43125	43125.00	37.48	800
t2/z300-800-74500	0	0	42292.00	42292.00	0	0	42292	42292.00	52.53	800
t2/z300-800-89300	0	0	44114.00	44114.00	0	0	44114	44114.00	69.25	800
t2/z300-1000-96590	0	0	71562.00	71562.00	0	0	71562	71562.00	83.65	1000
t2/z300-1000-120500	0	0	76345.00	76345.00	0	0	76345	76345.00	115.64	1000
t2/z300-1000-144090	0	0	78880.00	78880.00	0	0	78880	78880.00	153.42	1000

árvore de enumeração e, assim, explorando-se menor nós (configurações OCI e OCICLIQUE). Em particular, explorar menos nós leva a menos chamadas da heurística primal, e isso pode ser notado pelas instâncias para as quais não foi possível encontrar soluções viáveis.

O último experimento testou o impacto da heurística primal.

O experimento CLIQUEHEUR, **sem OCI, conflitos por clique, com heurística**, inclui a heurística primal na configuração do experimento CLIQUE. Os resultados são exibidos na Tabela 5.

Tabela 5: Resultados para a configuração CLIQUEHEUR

Inst	**STC**	**OCIC**	**FOrlx**	**FOraiz**	**Nós**	**NoInt**	**Prim**	**Dual**	**Time**	**Fix**
t1/z50-200-199	58	0	612.00	612.00	33	30	708	708.00	0.08	0
t1/z50-200-398	39	0	652.00	652.00	14	5	770	770.00	0.05	0
t1/z50-200-597	90	0	726.00	726.00	96	89	917	917.00	0.30	0
t1/z50-200-995	153	0	998.80	1024.25	1306	1015	1324	1324.00	4.04	0
t1/z100-300-448	170	0	3440.00	3440.00	98	91	4041	4041.00	0.96	0
t1/z100-300-897	499	0	4658.33	4950.83	7153	7123	5658	5658.00	92.24	1
t1/z100-500-1247	89	0	3454.00	3454.00	19	14	4275	4275.00	0.47	0
t1/z100-500-2495	740	0	4892.50	5309.00	26584	21443	5997	5997.00	509.63	0
t1/z100-500-3741	632	0	5406.50	5787.00	23228	19785	8859	6283.00	600.04	0
t1/z200-600-1797	1064	0	12180.71	12528.25	6492	4919	15040	12911.50	600.03	0
t1/z200-800-3196	1203	0	18475.00	19802.61	4356	4326	22105	20519.17	600.07	0
t2/z50-200-3903	1	0	1634.00	1634.00	0	0	1636	1636.00	0.10	162
t2/z50-200-4877	0	0	2043.00	2043.00	0	0	2043	2043.00	0.12	169
t2/z50-200-5864	0	0	2338.00	2338.00	0	0	2338	2338.00	0.18	177
t2/z100-300-8609	0	0	7434.00	7434.00	0	0	7434	7434.00	0.28	300
t2/z100-300-10686	0	0	7968.00	7968.00	0	0	7968	7968.00	0.41	300
t2/z100-300-12761	0	0	8166.00	8166.00	0	0	8166	8166.00	0.65	300
t2/z100-500-24740	26	0	11846.06	11921.36	17	3	12652	12652.00	10.79	68
t2/z100-500-30886	0	0	11232.00	11232.00	0	0	11232	11232.00	2.29	500
t2/z100-500-36827	2	0	11457.00	11457.00	0	0	11481	11481.00	3.52	472
t2/z200-400-13660	5	0	17680.00	17680.00	0	0	17728	17728.00	0.77	374
t2/z200-400-17089	0	0	18617.00	18617.00	0	0	18617	18617.00	1.10	388
t2/z200-400-20469	0	0	19140.00	19140.00	0	0	19140	19140.00	1.71	400
t2/z200-600-34504	0	0	20716.00	20716.00	0	0	20716	20716.00	2.22	600
t2/z200-600-42860	0	0	18025.00	18025.00	0	0	18025	18025.00	3.57	600
t2/z200-600-50984	0	0	20864.00	20864.00	0	0	20864	20864.00	5.83	600
t2/z200-800-62625	0	0	39895.00	39895.00	0	0	39895	39895.00	6.48	800
t2/z200-800-78387	0	0	37671.00	37671.00	0	0	37671	37671.00	11.56	800
t2/z200-800-93978	0	0	38798.00	38798.00	0	0	38798	38798.00	18.10	800
t2/z300-600-31000	0	0	43721.00	43721.00	0	0	43721	43721.00	2.25	600
t2/z300-600-38216	1	0	44260.00	44260.00	0	0	44267	44267.00	3.85	571
t2/z300-600-45310	0	0	43071.00	43071.00	0	0	43071	43071.00	5.87	600
t2/z300-800-59600	0	0	43125.00	43125.00	0	0	43125	43125.00	6.17	800
t2/z300-800-74500	0	0	42292.00	42292.00	0	0	42292	42292.00	11.12	800
t2/z300-800-89300	0	0	44114.00	44114.00	0	0	44114	44114.00	17.84	800
t2/z300-1000-96590	0	0	71562.00	71562.00	0	0	71562	71562.00	14.38	1000
t2/z300-1000-120500	0	0	76345.00	76345.00	0	0	76345	76345.00	25.55	1000
t2/z300-1000-144090	0	0	78880.00	78880.00	0	0	78880	78880.00	42.60	1000

Analisando-se a Tabela 5 pode-se verificar que não há uma diferença significativa nos resultados obtidos no configuração CLIQUEHEUR em relação aos obtidos na configuração CLIQUE.

5 Conclusão

Neste trabalho, foi estudado o impacto de diversas técnicas na tentativa de melhorar o tempo de resolução do AGMCC. Também foi observado que a rotina de separação de subtour se comporta bem na prática, seguindo o resultado esperado da separação polinomial.

Podemos ver que modelos mais complexos não são necessáriamente melhores, já que existe um *trade-off* entre o tempo gasto e a qualidade dos limitantes. No nosso caso, o uso das desigualdades OCI melhorou o limitante, como esperado, mas o tempo gasto dificultou a melhora do limitante primal dentro do tempo limite.

Da mesma forma, pela dificuldade do problema, uma simples heurística primal não foi suficiente.

Assim, dentre todas as configurações exploradas, a configuração CLIQUE demonstrou melhores resultados e seria escolhida para resolver o problema.

Melhorias podem ser feitas na tentativa de tornar os cortes mais rápidos, em especial OCI, e encontrar heurísticas primais melhores.

Referências

SAMER, P.; URRUTIA, S. A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters*, v. 9, n. 1, p. 41–55, Jan 2015. ISSN 1862-4480. Disponível em: <https://doi.org/10.1007/s11590-014-0750-x>.

CORMEN, T. H. et al. *Introduction to algorithms*. [S.l.]: MIT press, 2009.

HALIM, S.; HALIM, F. *Competitive Programming 3: The New Lower Bound of Programming Contests*. Lulu.com, 2013. ISBN 9788392212355. Disponível em: <https://books.google.com.br/books?id=vUc-nwEACAAJ>.

NEMHAUSER, G. L.; WOLSEY, L. A. Integer and combinatorial optimization. Wiley-Interscience, 1988.

SINGH, N. *Dinic's algorithm for Maximum Flow*. 2018. Disponível em: <https://www.geeksforgeeks.org/dinics-algorithm-maximum-flow/>.

AGRAWAL, S. *Dijkstra's Shortest Path Algorithm using priority queue of STL*. 2018. Disponível em: <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-using-priority_queue-stl/>.

NEMHAUSER, G. L.; SIGISMONDI, G. A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of the Operational Research Society*, Taylor & Francis, v. 43, n. 5, p. 443–457, 1992. Disponível em: <https://doi.org/10.1057/jors.1992.71>.