

Cortes do tipo capacidade para o Problema de Cobertura Multi-Veículo

Bruno Davi Mattos de Oliveira

Universidade Federal Fluminense

Rua Passo da Pátria, 156. São Domingos, Niterói - RJ

bruno_mattos@id.uff.br

Artur Alves Pessoa

Universidade Federal Fluminense

Rua Passo da Pátria, 156. São Domingos, Niterói - RJ

arturpessoa@id.uff.br

Marcos Costa Roboredo

Universidade Federal Fluminense

Rua Passo da Pátria, 156. São Domingos, Niterói - RJ

mroboredo@id.uff.br

RESUMO

O Problema do Roteamento Multi-Veículos com Cobertura (m -CTP) envolve um depósito e três subconjuntos distintos que representam clientes, facilidades obrigatórias e facilidades opcionais. Cada cliente está vinculado a um subconjunto específico de facilidades opcionais que definem seu conjunto de cobertura. O objetivo é determinar um conjunto de rotas com custo mínimo que satisfaçam as seguintes restrições: cada rota começa e termina no depósito; cada facilidade obrigatória é visitada exatamente uma vez em uma única rota; cada rota visita não mais do que p facilidades e tem um custo máximo de q ; para cada cliente, pelo menos uma facilidade opcional de seu conjunto de cobertura deve ser visitada por uma das rotas. Neste artigo, apresentamos uma nova família de cortes tipo capacidade e relatamos várias experiências que comprovam a eficácia do algoritmo e cortes propostos.

PALAVRAS CHAVE. Problema de Cobertura Multi-veículo, Branch-and-cut-and-price, Desigualdades.

ABSTRACT

The Multi-Vehicle Covering Tour Problem (m -CTP) involves a depot and three distinct subsets representing customers, mandatory facilities, and optional facilities. Each customer is linked to a specific subset of optional facilities that define its coverage set. The goal is to determine a set of routes with minimal cost that satisfy the following constraints: each route begins and ends at the depot; every mandatory facility is visited exactly once on a single route; each route visits no more than p facilities and has a maximum cost of q ; for each customer, at least one optional facility from its coverage set must be visited by one of the routes. In this paper, we present a new family of capacity-like cuts and report several experiments that prove the effectiveness of the proposed algorithm and cuts.

KEYWORDS. Multi-Vehicle Covering Tour Problem. Branch-and-cut-and-price, Inequalities.

1. Introdução

O m -CTP foi proposto por Hachicha et al. [2000] e é formalmente definido da seguinte maneira. Seja $G = (V, E)$ um grafo completo e não direcionado. O conjunto de nós é $V = 0 \cup M \cup O \cup C$, onde 0 representa o depósito, $M = 1, \dots, m$ representa o conjunto de instalações obrigatórias, $O = m + 1, \dots, m + o$ representa o conjunto de instalações opcionais e $C = m + o + 1, \dots, m + o + c$ representa o conjunto de clientes. Cada aresta $e \in E$ é associada a um custo de viagem c_e . Cada cliente $j \in C$ está associado a um conjunto de cobertura $\phi(j) \subseteq O$. O objetivo do m -CTP é projetar um conjunto de rotas finitas com custo total de viagem mínimo satisfazendo as seguintes restrições: cada rota deve começar e terminar no depósito; cada instalação obrigatória $i \in M$ é visitada exatamente uma vez em todas as rotas; para cada cliente $j \in C$, há pelo menos uma instalação opcional $i \in \phi(j)$ que é visitada por uma das rotas; cada rota visita no máximo p instalações; o custo máximo de cada rota é q . Para os casos específicos do m -CTP com $q = +\infty$ ou $p = +\infty$, os referimos, respectivamente, como m -CTP- p e m -CTP- q . Ao longo deste trabalho, podemos usar o termo m -CTP para denotar as instâncias do problema onde ambos os tipos de restrições estão presentes, o que deve ficar claro a partir do contexto.

Para ilustrar o m -CTP, apresentamos uma instância simplificada com $M = \{1\}$, $O = \{2, 3, 4\}$, $C = \{5, 6\}$, $p = 2$, $q = +\infty$, $\phi(5) = \{3, 4\}$ e $\phi(6) = \{2\}$. Omitimos os custos de viagem. A Figura 1 mostra um exemplo de uma solução viável para a instância proposta, onde os nós marcados em amarelo, vermelho, verde e azul representam, respectivamente, o depósito, a instalação obrigatória, as instalações opcionais e os clientes. Em torno de cada cliente, delineamos uma circunferência vermelha tracejada. As instalações opcionais dentro da circunferência representam o conjunto de cobertura do cliente.

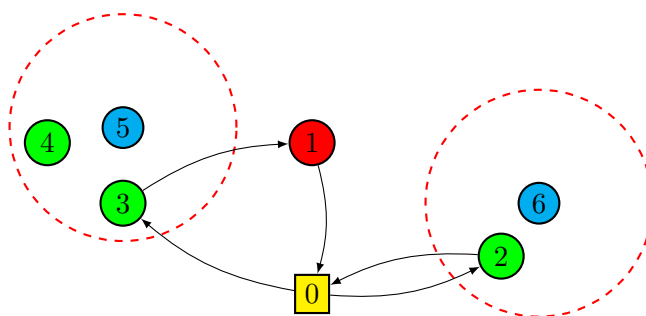


Figura 1: Exemplo de uma solução para uma instância do m -CTP.

O PCMV foi inicialmente proposto por Hachicha et al. [2000], que também introduziu heurísticas para sua resolução. Ao longo do tempo, outros algoritmos foram apresentados por diversos autores, como Naji-Azimi et al. [2012], Lopes et al. [2013], Ha et al. [2013], Kammoun et al. [2015], Glize et al. [2020].

Além da versão clássica do PCMV, a literatura também aborda algumas variantes, como a variante com múltiplos depósitos, proposta por Allahyari et al. [2015], a variante em que os clientes devem ser atendidos mais de uma vez, mencionada em Pham et al. [2017], e a variante com otimização de velocidade, discutida por Margolis et al. [2021].

Para o m -CTP- p o melhor algoritmo exato é o proposto por Glize et al. [2020], onde foi desenvolvido um algoritmo *branch-and-cut-price* (BCP) capaz de resolver instâncias com até $|V| = 200$. No entanto, algumas instâncias não foram resolvidas até a otimalidade ou exigiram um tempo considerável para serem solucionadas por ambos os métodos. Já para o m -CTP, não há nenhum método exato na literatura, apenas métodos heurísticos, como a proposta por Pham et al. [2017].

2. Modelo BCP

Nesta seção, apresentamos o modelo BCP para o m -CTP. Um modelo BCP é composto por uma formulação MIP onde as variáveis são chamadas de *variáveis originais*, e um subproblema que define um conjunto adicional de variáveis (ou colunas), cada uma associada a uma solução do subproblema. As variáveis originais na formulação MIP são então requeridas a ser uma combinação linear das colunas, fortalecendo ainda mais a relaxação linear da formulação MIP. No caso do m -CTP, o subproblema é um Problema do Caminho Mais Curto com Restrição de Recurso (RCSP), e cada variável original conta o número de vezes que uma aresta de G é percorrida por uma solução. Para definir o subproblema RCSP, mostramos como traduzir G em um grafo direcionado G' que é usado como entrada para o solucionador RCSP. Chamamos isso de *grafo gerador de caminho*. Para empregar técnicas avançadas que melhoram o desempenho do BCP, também definimos *conjuntos de embalagem*, que são subconjuntos de vértices de G' que podem ser usados no máximo uma vez em uma solução completa.

2.1. Grafo gerador de caminho

Seja $G' = (V', A')$ um grafo direcionado tal que $V' = v_0, v_1, \dots, v_{|M|+|O|}$ e $A' = (v_i, v_j), (v_j, v_i) | i, j \in V', i < j$. Seja ρ um caminho que começa e termina no nó v_0 tal que este nó não seja visitado no meio do caminho. O caminho ρ está associado a dois números não negativos S_ρ^1 e S_ρ^2 que representam o consumo acumulado de dois recursos. S_ρ^1 e S_ρ^2 são calculados da seguinte forma. Quando ρ começa, $S_\rho^1 = S_\rho^2 = 0$. Sempre que uma aresta $(v_i, v_j) \in A'$ é percorrida, os valores de S_ρ^1 e S_ρ^2 são aumentados da seguinte forma:

Se $i = 0$ ou $j = 0$, o consumo de S_ρ^1 é acrescido em 0,5. Caso contrário, é acrescido em 1,0. Para S_ρ^2 , seu consumo é acrescido em $c_{(i,j)}$.

Dizemos que o caminho ρ é restrito por recursos se $S_\rho^1 \leq p$ e $S_\rho^2 \leq q$. A Figura 2 ilustra o grafo gerador de caminhos para o exemplo simplificado proposto, onde marcamos em amarelo, verde e vermelho o depósito, as facilidades opcionais e obrigatórias, respectivamente. Em cada aresta, denotamos por s_1 e s_2 o aumento no valor de S_ρ^1 e S_ρ^2 , respectivamente.

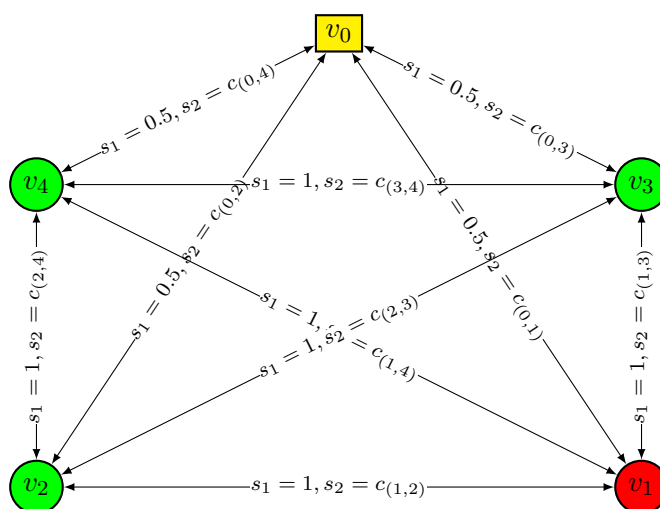


Figura 2: Grafo gerador de caminho para o exemplo dado.

2.2. Formulação Mestre

Nesta seção, introduzimos uma formulação MIP que considera variáveis de caminho sobre G' definido anteriormente. A ideia por trás da formulação proposta é que cada rota que visita no

máximo p facilidades e com um custo total menor ou igual a q está associada a uma variável de caminho em G' . Especificamente, a formulação usa uma variável inteira x_e para cada $e \in E''$, onde $E'' = e = (i, j) \in E | i, j \in 0 \cup M \cup O$. Cada variável x_e indica quantas vezes a aresta e é usada na solução. Seja P o conjunto de caminhos restritos por recursos sobre o grafo G' . Para cada $\rho \in P$, a formulação considera uma variável inteira λ_ρ para indicar o número de vezes que o caminho ρ é usado na solução. Além das variáveis, a formulação usa os seguintes conjuntos: o conjunto $\delta(S) = (i, j) \in E'' | (i \in S \wedge j \notin S) \vee (i \notin S \wedge j \in S)$, para um dado $S \subseteq 0 \cup M \cup O$; a constante h_ρ^a indica o número de vezes que uma aresta $a \in A'$ é usada no caminho $\rho \in P$; o conjunto $M(e) = (v_i, v_j), (v_j, v_i)$, para cada $e = (i, j) \in E''$. A formulação segue.

$$\text{Min } \sum_{e \in E''} c_e x_e \quad (1a)$$

$$\text{s.t. } \sum_{e \in \delta(\{i\})} x_e = 2, \quad i \in M; \quad (1b)$$

$$\sum_{e \in \delta(\{i\})} x_e \leq 2, \quad i \in O; \quad (1c)$$

$$\sum_{e \in \delta(\phi(j))} x_e \geq 2, \quad j \in C; \quad (1d)$$

$$x_e = \sum_{\rho \in P} \sum_{a \in M(e)} h_\rho^a \lambda_\rho, \quad e \in E''; \quad (1e)$$

$$1 \leq \sum_{\rho \in P} \lambda_\rho \leq |M| + |O|; \quad (1f)$$

$$x_e \in \mathbb{Z}, \quad \forall e \in E''; \quad (1g)$$

$$\lambda_\rho \geq 0, \quad \rho \in P \quad (1h)$$

A função objetivo (1a) minimiza o custo total das rotas. As restrições (1b) garantem que cada nó obrigatório seja visitado exatamente uma vez. As restrições (1c) garantem que cada nó opcional seja visitado no máximo uma vez. As restrições (1d) garantem que haja pelo menos um nó visitado em $\phi(j)$, para cada cliente $j \in C$. As restrições (1e) garantem a relação entre as variáveis λ e x . Para um dado $e = (i, j) \in E''$, o valor de x_e é dado pelo número de vezes que as arestas em $M(e)$ são percorridas considerando todos os caminhos da solução ótima. As restrições (1f) garantem que a solução tenha pelo menos 1 caminho e no máximo $|M| + |O|$.

Como permitimos que os caminhos não elementares sejam soluções do RCSPP, é possível encontrar uma solução inteira para as variáveis x , mas uma solução fracionária para as variáveis λ , onde o grau de uma instalação opcional $i \in O$ é igual a 1, ou seja, existe $e^* \in \delta(i)$ tal que $\sum_{e \in \delta(i)} x_e = x_{e^*} = 1$. Para evitar ter que fazer branch nas variáveis λ , inserimos a seguinte restrição sob demanda, por inspeção: $x_{e^*} \leq \sum_{e \in \delta(i) \setminus e^*} x_e$.

2.3. Packing Sets

Nesta seção, definimos os *packing sets* que podem ser explorados por solucionadores BCP para ativar recursos avançados. Seja $\mathcal{S}^V \subset 2^{V' \setminus v_0}$ uma coleção de subconjuntos mutuamente disjuntos de $V' \setminus v_0$. Dizemos que os conjuntos em $\mathcal{S}^V$ são *packing sets* se houver pelo menos uma solução ótima para a formulação (1), satisfazendo as seguintes restrições:

$$\sum_{\rho \in P} \left(\sum_{v \in S} h_v^\rho \right) \lambda_\rho \leq 1, \quad S \in \mathcal{S}^V, \quad (2)$$

onde h_v^ρ indica quantas vezes o nó v aparece no caminho ρ . Em outras palavras, para cada conjunto de agrupamento $S \in \mathcal{S}^V$, no máximo um nó neste conjunto pode ser percorrido por um caminho e ele pode ocorrer no máximo uma vez. Em nosso modelo, definimos um conjunto de agrupamento diferente v_i para cada $i \in M \cup O$ uma vez que cada facilidade é visitada no máximo uma vez. Para detalhes sobre como os elementos do estado-da-arte de BCP para roteamento podem ser ativados com base nos packing sets fornecidos, consulte Pessoa et al. [2020].

3. Uma nova família de cortes para o m -CTP

Nesta seção, introduzimos um conjunto de desigualdades válidas para o m -CTP. Essas desigualdades podem ser vistas como uma generalização das seguintes restrições propostas por Gendreau et al. [1997] para uma versão do m -CTP que considera apenas uma única rota:

$$\sum_{e \in \delta(S)} x_e \geq 2, \forall S \subseteq M \cup O: S \cap M \neq \emptyset \text{ ou } \phi(j) \subseteq S, \text{ para algum } j \in C \quad (3)$$

As restrições (3) garantem que um determinado conjunto $S \subseteq M \cup O$ deve ser visitado por pelo menos um veículo (que entra e sai de S) quando há uma instalação obrigatória em S , ou quando todas as instalações do conjunto de cobertura de um determinado cliente estão em S . Note que, se tivermos $S = \phi(j)$, $j \in C$, obtemos as restrições (1d).

O conjunto proposto de desigualdades válidas visa identificar casos em que o número de veículos que devem visitar um determinado subconjunto S de instalações é maior que um. Para isso, utilizamos do limite p no número de instalações que podem ser visitadas por cada veículo. Se pudermos provar que pelo menos α instalações de S devem ser visitadas em qualquer solução viável, então o número de veículos que entram (e saem) de S deve ser pelo menos $\lceil \alpha/p \rceil$. Nas desigualdades propostas, α é calculado como o número de instalações obrigatórias em S mais o valor objetivo ótimo para uma relaxação linear de um Problema de Cobertura de Conjuntos com uma função objetivo modificada que minimiza o número de instalações opcionais de S que são visitadas. Esse método leva à seguinte desigualdade:

$$\sum_{e \in \delta(S)} x_e \geq 2K(S), \quad \forall S \subseteq M \cup O, \quad (4)$$

onde,

$$K(S) = \left\lceil \left(|S \cap M| + \min_{\xi \in \mathbb{R}_+^{|O|}} \left\{ \sum_{i \in S \cap O} \xi_i \mid \sum_{i: i \in \phi(j)} \xi_i \geq 1, \forall j \in C \right\} \right) / p \right\rceil.$$

A partir de agora, nos referiremos à desigualdade (4) como um *Corte de Capacidade de Cobertura (CCC)*. Para ilustrar a desigualdade, considere uma instância do m -CTP com $p = 2$. Para esta instância, tomamos $S = i_1, i_2, i_3$, onde $i_1 \in M$, $i_2, i_3 \in O$ e existem dois clientes $j_1, j_2 \in C$ tal que $\phi(j_1) = i_2$ e $\phi(j_2) = i_3$. A Figura 3 ilustra a instância e o conjunto S .

A Proposição 1 demonstra que o CCC é válido para o m -CTP.

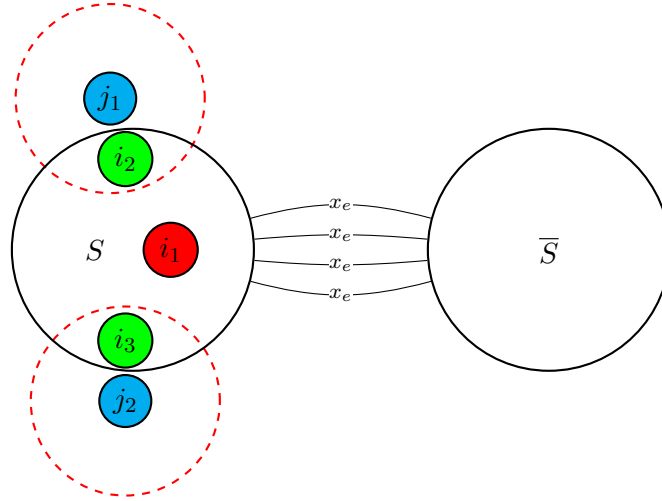


Figura 3: Uma ilustração dos cortes propostos.

Proposição 1. *O CCC é válido para o m-CTP.*

Demonstração. Seja $S \subseteq M \cup O$. Primeiro, provamos que pelo menos $|S \cap M| + z^*$ instalações em S devem ser visitadas em qualquer solução viável do m-CTP, onde

$$z^* = \min_{\xi \in \mathbb{R}_+^{|O|}} \left\{ \sum_{i \in S \cap O} \xi_i \mid \sum_{i: i \in \phi(j)} \xi_i \geq 1, \forall j \in C \right\}$$

. De fato, pela definição do m-CTP, todas as $|S \cap M|$ instalações obrigatórias em S devem ser visitadas. Em relação às instalações opcionais em S , z^* representa um limite inferior para o número mínimo de instalações opcionais visitadas em S em qualquer solução viável. Portanto, nenhuma solução do m-CTP pode visitar menos de $|S \cap M| + z^*$ instalações em S . Agora, seja $\bar{x}$ uma solução viável para (1). Como nenhuma rota pode visitar mais do que p instalações, o número total de vezes que a solução percorre uma aresta de $\delta(S)$ deve ser pelo menos $2 \lceil (|S \cap M| + z^*)/p \rceil$. Assim, (4) deve ser satisfeita por $\bar{x}$. $\square$

3.1. Algoritmo de separação

Nossa abordagem para encontrar os CCC violados envolve resolver um problema de separação. Para cada valor fixo de $K(S)$, modelamos este problema como um MIP. Aqui, $S \subseteq M \cup O$ é buscado de forma a minimizar o lado esquerdo da desigualdade (4), com a restrição de que $K(S)$ seja maior ou igual a k , onde k varia de 1 a $k_{\max}$.

O valor máximo de k , denotado como $k_{\max}$, é obtido quando S contém todos os nós obrigatórios e opcionais ($M \cup O$). Esta abordagem nos permite explorar diferentes configurações de S , buscando identificar as violações dos CCC.

Para cada i pertencente a $M \cup O$, introduzimos uma variável binária z_i , que indica se i pertence a S ($z_i = 1$) ou não ($z_i = 0$). Além disso, para cada $e \in E''$, onde E'' é o conjunto de arestas do grafo G' , introduzimos uma variável binária r_e que indica se e pertence à fronteira de S ($r_e = 1$) ou não ($r_e = 0$). Essas variáveis são cruciais para a formulação do problema e ajudam a definir as condições de validade dos cortes CCC.

A partir dessas variáveis de primeiro nível, formulamos o problema de separação, buscando determinar o conjunto S que satisfaz as condições impostas e viola a desigualdade (4).

Isso nos permite identificar eficientemente os cortes violados e, assim, fortalecer o modelo de otimização.

$$\text{Min } \sum_{e \in E''} \bar{x}_e r_e \quad (5a)$$

$$\text{s.t. } r_e \geq z_i - z_j, \quad e = (i, j) \in E'' \quad (5b)$$

$$r_e \geq z_j - z_i, \quad e = (i, j) \in E'' \quad (5c)$$

$$z_0 = 0 \quad (5d)$$

$$\sum_{i \in M} z_i + \min_{z' \in \mathbb{R}_+^{|O|}} \left\{ \sum_{i \in O} z_i \xi_i \mid \sum_{i \in \phi(j)} \xi_i \geq 1, \forall j \in C \right\} \geq (k-1)p + \epsilon \quad (5e)$$

$$r_e \in \{0, 1\}, \quad e \in E'' \quad (5f)$$

$$z_i \in \{0, 1\}, \quad i \in M \cup O, \quad (5g)$$

onde ϵ é um número pequeno que ainda é suficientemente grande para garantir que o lado esquerdo de (5e) seja estritamente maior que $(k-1)p$, apesar de erros numéricos.

A função objetivo (5a) visa minimizar o lado esquerdo de (4). As restrições (5b) e (5c) garantem a relação entre as variáveis r e z . A restrição (5d) garante que o depósito não está em S . A restrição (5e) garante que $K(S) \geq k$. Para ver isso, observe que ela assegura que

$$\left(\sum_{i \in M} z_i + \min_{z' \in \mathbb{R}_+^{|O|}} \left\{ \sum_{i \in O} z_i \xi_i \mid \sum_{i \in \phi(j)} \xi_i \geq 1, \forall j \in C \right\} \right) / p > (k-1). \quad (6)$$

Como (5) não pode ser otimizada em sua forma atual, obtemos uma formulação equivalente substituindo o problema de otimização que aparece na Restrição (5e) por seu dual. Para isso, para cada $j \in C$, seja y_j a variável dual associada à restrição $\sum_{i \in \phi(j)} z'_i \geq 1$. A formulação segue.

$$\text{Min (5a)} \quad (7a)$$

$$\text{s.t. (5b), (5c), (5d), (5f), (5g)} \quad (7b)$$

$$\sum_{i \in M} z_i + \left\{ \begin{array}{l} \text{Max } \sum_{j \in C} y_j \\ \text{s.t. } \sum_{j \in C | i \in \phi(j)} y_j \leq z_i, \quad i \in O \\ y_j \geq 0, \quad j \in C \end{array} \right\} \geq (k-1)p + \epsilon, \quad (7c)$$

que é equivalente a:

$$\text{Min (5a)} \tag{8a}$$

$$\text{s.t. (5b), (5c), (5d), (5f), (5g)} \tag{8b}$$

$$\sum_{i \in M} z_i + \sum_{j \in C} y_j \geq (k-1)p + \epsilon \tag{8c}$$

$$\sum_{j \in \sigma(i)} y_j \leq z_i, \quad i \in O \tag{8d}$$

$$y_j \geq 0 \quad j \in C. \tag{8e}$$

Nós utilizamos (8) para encontrar *CCC* violados, para $k = 1, \dots, k_{\max}$.

4. Resultados Computacionais

Nesta seção, compartilhamos os resultados obtidos pelo nosso modelo em três conjuntos distintos de instâncias. Os dois primeiros conjuntos são os mesmos utilizados por Glize et al. [2020] e Pham et al. [2017] para investigar o *m*-CTP-*p* e o *m*-CTP, respectivamente. O terceiro conjunto é uma contribuição original deste trabalho, inspirado no conjunto de instâncias de CVRP proposto por Uchoa et al. [2017] e abrangendo até 393 instalações. Todas as instâncias utilizadas nos experimentos estão disponíveis para acesso público no repositório online https://github.com/brunomattos1/m-ctp_instances.

Optamos por resolver o modelo utilizando o framework VRPSolver, desenvolvido por Pessoa et al. [2020] com o intuito de simplificar a implementação de algoritmos BCP eficientes para problemas de roteamento de veículos e outros problemas relacionados. Escolhemos o framework VRPSolver para implementar o algoritmo BCP devido às suas características de ponta, tais como um algoritmo de rotulação bidirecional Righini e Salani [2006] no conceito de baldes, proposto por Sadykov et al. [2021], para resolver o subproblema de precificação, relaxamento de rota *ng* Baldacci et al. [2011], cortes de Chvátal-Gomory de classificação-1 com memória limitada, Pecin et al. [2016], enumeração de rotas, Contardo e Martinelli [2014], ramificação forte, Achterberg et al. [2005], técnica automática de estabilização dual, Pessoa et al. [2018]. Além disso, destacamos o sucesso comprovado do VRPSolver em várias variantes de VRP e de escalonamento, conforme demonstrado em Roboredo et al. [2023], Soares e Roboredo [2023], Ben Mohamed et al. [2023], Bulhões et al. [2020] e Pessoa et al. [2020].

Todos os testes foram realizados em um processador Intel(R) Core(TM) i7-10700 CPU@ 2.90GHz. O sistema operacional utilizado foi o Ubuntu e o algoritmo foi implementado em Julia utilizando o framework VRPSolver v0.4.1a (<https://vrpsolver.math.u-bordeaux.fr/>). O CPLEX 12.10 foi utilizado para resolver formulações LP e MIP. As tabelas apresentadas neste artigo contêm apenas estatísticas médias. Dados detalhados para cada instância testada são fornecidos em um material suplementar online.

4.1. Resultados para Glize et al. [2020] e Pham et al. [2017]

A Tabela 1 apresenta os indicadores de desempenho do nosso método sobre instâncias da literatura, e uma comparação entre as duas versões do algoritmo BCP proposto (com e sem *CCC*) e o melhor método exato da literatura para o *m*-CTP-*p* proposto por Glize et al. [2020]. Para as instâncias de *m*-CTP e *m*-CTP-*p*, nosso método utiliza como limites superiores iniciais os valores de solução encontrados pelas heurísticas propostas por Pham et al. [2017] e Kammoun et al. [2015], respectivamente. Em relação à comparação entre nosso algoritmo e o proposto por Glize et al. [2020], destacamos dois pontos. Primeiro, cada tempo de execução relatado pela literatura foi dividido por 1,5. Esse fator é a razão entre os scores do processador utilizado por nós e pela

literatura obtidos em www.cpubenchmark.net, para um único thread. O segundo ponto é que a literatura considerou um limite de tempo de 7200s. Por esse motivo, definimos um limite de tempo de 4800s ($7200/1.5$) para experimentos sobre instâncias de m -CTP- p . Para as demais instâncias, definimos um limite de tempo de 7200s. Para cada problema (m -CTP ou m -CTP- p), indicamos a categoria das instâncias (Coluna *Categoria*) e o número de instâncias na categoria correspondente (Coluna *#Inst.*). Além disso, para cada metodologia comparada, apresentamos o número de soluções ótimas comprovadas dentro do limite de tempo (Coluna *#Opt*) e o tempo total médio consumido em segundos (Coluna *T(s)*).

Problema	Categoria	#Inst	#Opt	T(s)	BCP proposto			
					Glize et al. [2020]	Sem CCC	Com CCC	
m -CTP- p	$ V' = 100, M = 0$	32	32	4.2	32	1.7	32	1.9
m -CTP- p	$ V' = 100, M > 0$	32	31	≥ 151.3	32	55.1	32	2.1
m -CTP- p	$ V' = 200$	32	27	≥ 966.3	32	132.5	32	6.1
m -CTP	$ V' = 100, M = 0$	64	-	-	64	3.8	64	14.0
m -CTP	$ V' = 100, M > 0$	64	-	-	64	1.7	64	2.0
m -CTP	$ V' = 200$	64	-	-	63	206.1	63	128.1

Tabela 1: Resultado nas instância da literatura.

A análise da Tabela 1 revela uma notável superioridade do nosso novo algoritmo BCP em relação ao método proposto por Glize et al. [2020] em todas as categorias avaliadas, tanto em termos de tempo de execução quanto no número de instâncias resolvidas de forma ótima. Uma observação significativa é que todas as instâncias de m -CTP- p foram resolvidas de forma ótima, mesmo na ausência do uso dos cortes CCC. Destaca-se que cinco instâncias de m -CTP- p foram solucionadas de forma ótima pela primeira vez neste estudo: A2-20-100-100-6, A2-20-100-100-8, B2-1-100-100-8, B2-20-100-100-6 e B2-20-100-100-8, com custos de solução ótima de, respectivamente, 20966, 18415, 13137, 25960 e 22082. A aplicação dos cortes CCC resultou em uma melhoria significativa no desempenho do nosso algoritmo, levando a ganhos de mais de uma ordem de magnitude.

Quanto às instâncias de m -CTP, ambas as versões do nosso algoritmo alcançaram a solução ótima para 195 das 196 instâncias. É interessante observar que a vantagem obtida com a aplicação dos cortes foi mais expressiva para a categoria de instâncias de maior porte.

A Tabela 2 fornece uma visão da redução do *gap* pela introdução de diferentes tipos de cortes em instâncias grandes da literatura, onde o número de vértices é de 200. Cada entrada na tabela representa a diferença percentual entre a melhor solução conhecida e o limite inferior obtido pela geração de colunas pura no nó raiz (Coluna *Gap GC*). Esta diferença percentual diminui quando aplicamos os cortes da literatura (3) na relaxação da geração de colunas (Coluna *Cortes Lit.*), indicando uma redução significativa na diferença de otimalidade. A introdução adicional dos CCC amplifica essa redução, como demonstrado na Coluna *Cortes CCC*. Além disso, a aplicação dos cortes de Chvátal-Gomory Rank-1, fornecidos pelo framework VRPSolver, resulta em uma redução adicional na diferença de otimalidade, conforme apresentado na Coluna *Cortes RIC*.

Esses resultados destacam a importância dos cortes na melhoria da qualidade das soluções obtidas, mostrando como diferentes tipos de cortes podem trabalhar em conjunto para reduzir a lacuna entre o melhor valor conhecido e o limite inferior. Tal redução é essencial para aproximar os resultados obtidos do valor ótimo e demonstra a eficácia dos cortes na resolução de problemas de

Instância	$m\text{-CTP-}p$						$m\text{-CTP}(q = 500)$						$m\text{-CTP}(q = 250)$					
	GC			Cortes			GC			Cortes			GC			Cortes		
	Gap	Lit.	CCC	Rank-1	Gap	Lit.	Gap	Lit.	CCC	Rank-1	Gap	Lit.	Gap	Lit.	CCC	Rank-1	Gap	Lit.
A2-1-100-100-4	4.22	0.20	77.8	100.00	3.25	0.00	6.80	0.35	100.00	-	3.63	0.54	39.61	16.73				
A2-1-100-100-5	2.14	15.07	100.00	-	1.86	11.58	0.40	2.33	100.00	-	7.71	0.26	100.00	-				
A2-1-100-100-6	6.69	2.39	100.00	-	5.47	4.20	3.63	0.54	83.62	100.00	1.91	0.79	89.13	100.00				
A2-1-100-100-8	6.50	4.57	65.60	100.00	8.32	0.24	7.71	0.26	26.21	15.13	3.49	2.56	25.82	9.89				
A2-20-100-100-4	1.82	0.00	38.84	56.08	1.80	0.21	1.91	0.79	36.74	60.07	3.49	2.56	44.25	41.28				
A2-20-100-100-5	4.15	1.13	48.28	89.74	3.39	0.63	3.49	2.56	62.44	82.77	5.67	1.96	65.21	100.00				
A2-20-100-100-6	4.33	1.76	54.99	100.00	3.42	1.39	5.67	1.96	74.26	100.00	8.46	0.98	39.85	78.70				
A2-20-100-100-8	7.06	1.92	46.12	85.59	5.07	1.07	8.46	0.98	63.82	86.57	4.70	0.80	38.49	89.67				
B2-1-100-100-4	3.49	0.78	21.66	71.94	4.94	0.65	4.70	0.80	10.04	26.70	5.44	1.33	17.84	34.03				
B2-1-100-100-5	2.15	2.05	100.00	-	4.51	1.36	5.44	1.33	25.58	59.89	7.19	0.54	14.04	41.96				
B2-1-100-100-6	4.24	0.79	44.16	75.64	4.86	0.82	7.19	0.54	37.57	36.06	17.16	0.15	23.26	38.68				
B2-1-100-100-8	5.36	1.99	25.51	100.00	6.18	0.49	17.16	0.15	45.97	81.90	1.35	3.49	2.36	17.84				
B2-20-100-100-4	1.35	3.27	100.00	-	1.35	3.49	1.35	3.49	100.00	-	2.95	1.04	82.84	100.00				
B2-20-100-100-5	2.97	0.92	30.72	78.67	2.95	0.81	2.95	1.04	33.60	73.56	3.03	1.65	31.35	79.97				
B2-20-100-100-6	3.01	0.64	39.82	50.32	3.04	1.90	3.03	1.65	46.84	78.88	4.49	1.61	39.84	45.38				
B2-20-100-100-8	4.85	1.68	31.05	100.00	4.83	2.06	4.49	1.61	33.52	100.00	5.27	1.27	32.21	60.82				
Average	4.02	2.44	57.78	83.99	4.08	1.93	5.27	1.27	55.01	69.34	42.88	56.99						

Tabela 2: Comparação na redução dos gaps.

otimização.

5. Conclusão

Neste trabalho, abordamos o m -CTP, uma variante dos Problemas de Roteamento de Veículos onde as rotas são responsáveis por visitar facilidades para atender clientes dentro de uma rede dada. Nesse cenário, as rotas são limitadas tanto pelo número de facilidades que podem visitar quanto pelo custo total incorrido. Nosso foco foi desenvolver uma família de cortes robustos visando dois aspectos fundamentais do problema: cobertura de facilidades e capacidade do veículo. Além de um algoritmo para separá-los.

Para avaliar a eficácia de nosso método e cortes propostos, conduzimos experimentos computacionais extensos divididos em duas fases. Inicialmente, aplicamos nosso modelo, tanto com quanto sem os cortes propostos, a uma coleção de 288 instâncias de referência da literatura. Os resultados indicaram que nosso método resolveu de forma ótima quase todas as instâncias, com apenas uma exceção. Além disso, comparamos nosso algoritmo com o melhor método existente, adaptado para a variante do problema onde as limitações das rotas são baseadas apenas no número de facilidades visitadas. Impressionantemente, nossa abordagem superou o método existente mesmo sem a incorporação dos cortes propostos.

Referências

- Achterberg, T., Koch, T., e Martin, A. (2005). Branching rules revisited. *Operations Research Letters*, 33(1):42–54.
- Allahyari, S., Salari, M., e Vigo, D. (2015). A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem. *European Journal of Operational Research*, 242(3):756–768. ISSN 0377-2217. URL <https://www.sciencedirect.com/science/article/pii/S0377221714008704>.
- Baldacci, R., Mingozzi, A., e Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283.
- Ben Mohamed, I., Klibi, W., Sadykov, R., Şen, H., e Vanderbeck, F. (2023). The two-echelon stochastic multi-period capacitated location-routing problem. *European Journal of Operational Research*, 306(2):645–667.
- Bulhões, T., Sadykov, R., Subramanian, A., e Uchoa, E. (2020). On the exact solution of a large class of parallel machine scheduling problems. *Journal of Scheduling*, 23.
- Contardo, C. e Martinelli, R. (2014). A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12.
- Gendreau, M., Laporte, G., e Semet, F. (1997). The covering tour problem. *Operations Research*, 45(4):568–576.
- Glize, E., Roberti, R., Jozefowicz, N., e Nogueve, S. U. (2020). Exact methods for mono-objective and bi-objective multi-vehicle covering tour problems. *European Journal of Operational Research*, 3:812–824.
- Ha, M. H., Bostel, N., Langevin, A., e Rousseau, L. M. (2013). An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. *European Journal of Operational Research*, 226:211–220.

- Hachicha, M., Hodgson, M. J., Laporte, G., e Semet, F. (2000). Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research*, 27:29–42.
- Kammoun, M., Derbel, H., Ratli, M., e Jarboui, B. (2015). A variable neighborhood search for solving the multi-vehicle covering tour problem. *Electronic Notes in Discrete Mathematics*, 47: 285–292.
- Lopes, R., Souza, V. A., e DA Cunha, A. S. (2013). A branch-and-price algorithm for the multi-vehicle covering tour problem. *Electronic Notes in Discrete Mathematics*, 44:61–66.
- Margolis, J. T., Song, Y., e Mason, S. J. (2021). A multi-vehicle covering tour problem with speed optimization. *Networks*, 79:119–142.
- Naji-Azimi, Z., Renaud, J., Ruiz, A., e Salari, M. (2012). A covering tour approach to the location of satellite distribution centers to supply humanitarian aid. *European Journal of Operational Research*, 222:596–605.
- Pecin, D., Pessoa, A., Poggi, M., e Uchoa, E. (2016). Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9.
- Pessoa, A., Sadykov, R., Uchoa, E., e Vanderbeck, F. (2020). A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183:483–523.
- Pessoa, A., Sadykov, R., Uchoa, E., e Vanderbeck, F. (2018). Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30.
- Pham, T. A., Hà, M. H., e Nguyen, X. H. (2017). Solving the multi-vehicle multi-covering tour problem. *Computers & Operations Research*, 88:258–278.
- Righini, G. e Salani, M. (2006). Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3(3): 255–273.
- Roboredo, M., Sadykov, R., e Uchoa, E. (2023). Solving vehicle routing problems with intermediate stops using vrp solver models. *Networks*, 81(3):399–416.
- Sadykov, R., Uchoa, E., e Pessoa, A. (2021). A bucket graph-based labeling algorithm with application to vehicle routing. *Transportation Science*, 55(1):4–28.
- Soares, V. C. e Roboredo, M. (2023). On the exact solution of the multi-depot open vehicle routing problem. *Optim Lett.*
- Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., e Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858.