Cortes de Posto 1 com memória limitada para o Problema de Roteamento de Veículos

Diego Pecin
Polytechnique Montréal e GERAD
diego.pecin@gerad.ca

Artur Pessoa
Departamento de Engenharia de Produção, UFF
artur@producao.uff.br

Marcus Poggi
Departamento de Informática, PUC-Rio
poggi@inf.puc-rio.br

Eduardo Uchoa
Departamento de Engenharia de Produção, UFF
uchoa@producao.uff.br

Haroldo G. Santos
Departamento de Ciência da Computação, UFOP
haroldo@iceb.ufop.br

RESUMO

Pecin et al. (2014) introduziu uma técnica de memória limitada que permite um uso eficiente dos cortes de subconjunto de linhas (*Subset Row Cuts*) (Jepsen et al. (2008)) ou qualquer outro corte de Posto 1 definido sobre a Formulação de Partição de Conjuntos do Problema de Roteamento de Veículos. Isso motiva uma investigação mais aprofunda desses cortes mais genéricos. Este trabalho apresenta um estudo poliedral que determinou os melhores conjuntos possíveis de multiplicadores para cortes de Posto 1 com até 5 restrições. Experimentos com instâncias clássicas do PRVC mostram que os multiplicadores recentemente descobertos melhoram significativamente os limites duais. Na verdade, esses cortes contribuiram decisivamente para resolver uma instância aberta de 420 clientes, a maior resolvida até o momento em que este artigo foi escrito.

PALAVRAS CHAVE. Partição de Conjuntos, Combinatória Poliédrica, Algoritmos de branch-cut-and-price

ABSTRACT

Pecin et al. (2014) introduced a "limited memory" technique that allows an efficient use of the Subset Row Cuts (Jepsen et al. (2008)) or any other cut of rank 1 in the Set Partitioning Formulation of Vehicle Routing Problems. This motivates a deeper investigation of those more generic cuts. This work presents a computational polyhedral study that determined the best possible sets of multipliers for Rank-1 cuts with up to 5 rows. Experiments with classical CVRP instances show that the newly discovered multipliers lead to significantly improved dual bounds. In fact, those cuts contributed decisively for solving an open instance with 420 customers, the largest one solved up to the time this paper was written.

KEYWORDS. Set Partitioning, Polyhedral Combinatorics, Branch-cut-and-price algorithms

1. Introdução

O Problema de Roteamento de Veículos (PRV) está entre os problemas de otimização combinatória mais amplamente estudados devido a sua aplicação direta em sistemas de distribuição de bens e serviços, vitais para as economias modernas. Dado a grande variedade de condições presentes nesses sistemas, a literatura do PRV compreende dezenas de variantes. Por exemplo, existem variantes que levam em consideração as capacidades dos veículos, janelas de tempo, múltiplos depósitos, frota heterogênea, coletas e entregas, etc. A Formulação de Partição de Conjuntos (FPC) [Balinski e Quandt, 1964] pode ser usada para modelar variantes que requerem que cada cliente seja visitado exatamente uma vez. Seja $V = \{1, \ldots, n\}$ o conjunto de clientes e Ω o conjunto de todas as rotas que respeitam as condições da variante considerada. Ainda, para cada rota $r \in \Omega$, seja c_r o custo de r, a_i^r um coeficiente binário que indica se r visita o cliente $i \in V$ e λ_r uma variável que indica se a rota r é usada ou não na solução. A formulação é a seguinte:

$$\text{Min:} \quad \sum_{r \in \Omega} c_r \lambda_r \tag{1}$$

$$\text{S.t.:} \quad \sum_{r \in \Omega} a_i^r \lambda_r = 1, \quad \forall i \in V, \tag{2}$$

$$\lambda_r \in \{0, 1\}, \quad \forall r \in \Omega. \tag{3}$$

Devido ao grande número de variáveis, a relaxação linear da FPC deve ser resolvida por geração de colunas. O subproblema de geração de colunas consiste em encontrar rotas com custo reduzido mínimo, um problema que é muitas vezes modelado como um problema de caminho mais curto com restrições de recursos em uma rede com custos de arco positivo ou negativo e resolvidos por *algoritmos de expansão de caminhos* (ou algoritmos de rotulagem), veja [Pugliese e Guerriero, 2013] um compilado recente sobre o assunto.

A relaxação linear da FPC geralmente não é forte o suficiente para ser a base de bons algoritmos exatos para a PRV, pelo menos para as variantes mais clássicas e competitivas: o PRV capacitado (PRVC) e o PRV com janelas de tempo (PRVJT). Para esse efeito, a FPC deve ser fortalecida com cortes adicionais válidos. Um algoritmo que combina geração de colunas e cortes em uma árvore de busca enumerativa é chamado de algoritmo de branch-cut-and-price (BCP). De acordo com a classificação proposta em [Poggi de Aragão e Uchoa, 2003], *cortes robustos* são aqueles que não mudam a estrutura do subproblema de geração de colunas. Em contraste, *cortes não-robustos* podem alterar a estrutura do subproblema, de tal forma que cada corte adicional torna mais difícil a sua resolução, ao ponto que a adição sucessiva de cortes torna o pricing intratável. Cortes robustos podem ser eficazes em alguns casos. Na verdade, alguns algoritmos BCP de sucesso [Kohl et al., 1999; Fukasawa et al., 2006; Kallehauge et al., 2006; Røpke, 2012] usam apenas tais cortes. No entanto, parece que o potencial para cortes robustos (pelo menos para as variantes do PRV clássicas) está esgotado: nenhuma nova família significativa desses cortes foi encontrada na última década. Como consequência, uma importante linha de pesquisa do PRV recente consiste em encontrar cortes não-robustos eficazes que não são tão prejudiciais para os algoritmos de rotulagem utilizados no subproblema de geração de colunas.

As restrições de Partição de Conjuntos (2) são as fontes naturais de cortes não-robustos. No entanto, famílias de cortes bem conhecidas tais como cortes de clique e ciclos ímpares [Grötschel et al., 2012] complicam bastante o subproblema de geração de colunas [Spoorendonk e Desaulniers, 2010]. Um avanço importante foi a introdução dos cortes de subconjuntos (*Subset Row Cuts* – SRCs) por [Jepsen et al., 2008]. Os SRCs utilizados na prática são subfamílias de clique e ciclos ímpares fortalecidos, cada SRC tem a particularidade de ser definido por um subconjunto de pequena cardinalidade das linhas em (2). Como resultado, os SRCs podem ser tratados por algoritmos de rotulagem. Na verdade, alguns dos melhores algoritmos exatos para a PRVC e PRVJT usam SRCs [Jepsen et al., 2008; Desaulniers et al., 2008; Baldacci et al., 2011; Contardo e Martinelli, 2014]. Uma generalização direta dos SRCs consiste em considerar qualquer corte de Chvátal-

Gomory [Chvátal, 1973] de Posto 1 obtido a partir de um pequeno subconjunto de linhas da FPC. Experimentos preliminares com os cortes de Posto 1 foram realizadas em [Petersen et al., 2008].

Os algoritmos acima mencionados separam SRCs de uma maneira muito cuidadosa, limitando o número de cortes adicionados a fim de evitar um impacto excessivo no subproblema de geração de colunas. Como resultado, o potencial desses cortes não é totalmente alcançado, especialmente em instâncias maiores ou mais difíceis. Pecin et al. [Pecin et al., 2014] introduziu a técnica de *memória limitada* para fazer com que os SRCs se tornassem muito menos prejudiciais aos algoritmos de rotulagem, sem necessariamente comprometer sua efetividade. Os resultados computacionais em instâncias do PRVC mostraram um grande avanço: devido aos melhores limites duais alcançados, o tamanho da maior instância resolvida aumentou de 150 para 360 clientes. A memória limitada também produziu melhorias muito significativas no PRVJT, permitindo a solução de muitas instâncias com 200 clientes [Pecin et al.]. Neste novo contexto, quando o limite SRC pode ser alcançado mesmo para instâncias razoavelmente grandes, há uma clara motivação para se tentar melhorar ainda mais os limites através da separação de cortes mais complexos de Posto 1.

Este trabalho foi elaborado para responder a seguinte pergunta: *quais são os conjuntos ideais de multiplicadores para cortes de Posto 1 com até 5 linhas?* Isto é feito a partir de uma investigação poliédrica computacional do Poliedro da Partição de Conjuntos. Experimentos em instâncias clássicas do PRVC revelam que os recém-descobertos cortes melhoram os limites duais. Finalmente, mostramos como esses cortes permitiram resolver a instância Golden_20 (420 clientes) na otimalidade. Esse artigo está organizado como a seguir. A Seção 2 revê os SRCs, os cortes de Posto 1 e a técnica de memória limitada. A Seção 3 descreve a metodologia utilizada para descobrir os conjuntos ideais de multiplicadores. A Seção 4 apresenta resultados computacionais sobre instâncias do PRVC. Finalmente, a Seção 5 contém algumas conclusões.

Neste artigo usamos uma notação não usual para representar vetores. Embora cada vector é assumido como sendo uma matriz de coluna única, muitas vezes é apresentado por uma lista de seus componentes, separados por vírgulas e entre parênteses. Além disso, quando é evidente que todos os componentes do vetor são números inteiros com um único dígito, as vírgulas são descartadas.

2. Cortes de memória limitada de Posto 1

Dado um conjunto base $C \subseteq V$ e um multiplicador p_i para cada $i \in C$, a seguinte desigualdade válida para a FPC é um Corte de Posto 1 (CP1):

$$\sum_{r \in \Omega} \left\lfloor \sum_{i \in C} p_i a_i^r \right\rfloor \lambda_r \leq \left\lfloor \sum_{i \in C} p_i \right\rfloor \tag{4}$$

Os cortes de subconjunto de linhas (SRCs) foram propostos em [Jepsen et al., 2008] e correspondem ao caso particular em que todos os multiplicadores não-nulos têm o mesmo valor $p = 1/k$, para algum inteiro k. Os seguintes tamanhos de conjunto de base e multiplicadores foram investigados nesse trabalho:

- 3SRCs: $|C| = 3$ e $p = 1/2$. Esses cortes têm o lado direito da desigualdade igual a 1 e podem ser vistos como cortes de clique enfraquecidos (porque eles só levam em conta os coeficientes das linhas em C). No entanto, eles ainda são muito eficazes na melhoria de limites duais e foram os únicos SRCs realmente separados em [Jepsen et al., 2008; Desaulniers et al., 2008; Baldacci et al., 2011; Contardo e Martinelli, 2014].

- 5,2SRCs: $|C| = 5$ and $p = 1/2$. Esses cortes têm lado direito igual a 2 e podem ser vistos como cortes de ciclo ímpar enfraquecidos, obtidos considerando-se apenas as linhas em C.

[Pecin et al., 2014] separa 3SRCs, 5,2SRCs e também:

- 4SRCs: $|C| = 4$ e $p = 2/3$. Apesar de não ter um multiplicador de formato $1/k$, eles ainda eram chamados de SRCs nesse trabalho.

- 5,1SRCs: $|C| = 5$ e $p = 1/3$.

A separação foi realizada de forma exaustiva, testando todas as possibilidades para determinados valores de C e k. Os autores também testaram encontrar os multiplicadores utilizando a formulação de Programação Inteira Mista proposta em [Fischetti e Lodi, 2007]. Os experimentos indicaram que cortes de Posto 1 mais complexos que SRCs poderiam de fato melhorar os limites duais. No entanto, esses cortes não puderam ser incorporados em códigos do estado-da-arte: não apenas a separação em si é muito cara, mas os cortes adicionados perturbam muito o algoritmo de rotulagem, tornando-o intratável.

Dado $C \subseteq V$, um vetor de multiplicadores p de dimensão $|C|$, um conjunto de memória M, $C \subseteq M \subseteq V$, o Corte de memória limitada (C, M, p)-Posto 1 (CP1-ml) é:

$$\sum_{r \in \Omega} \alpha(C, M, p, r)\lambda_r \leq \left\lfloor \sum_{i \in C} p_i \right\rfloor, \tag{5}$$

onde o coeficiente de uma rota r é computado como a seguir:

function $\alpha(C, M, p, r)$
$coeff \leftarrow 0$, $state \leftarrow 0$
for every vertex $i \in r$ (in order) **do**
  if $i \notin M$ **then**
    $state \leftarrow 0$
  else if $i \in C$ **then**
    $state \leftarrow state + p_i$
    if $state \geq 1$ **then**
      $coeff \leftarrow coeff + 1$, $state \leftarrow state - 1$
return $coeff$

A variável $coeff$ armazena o coeficiente a ser retornado. Cada vez que um vértice i em C é visitado, a variável $state$ é aumentada em p_i. Quando $state$ torna-se maior ou igual a 1, o seu valor é reduzido de 1 unidade e $coeff$ é incrementado. Quando $M = V$, a função α retorná á$\lfloor \sum_{i \in C} p_i a_i^r \rfloor$ e o CP1-ml será idêntico a um CP1. Por outro lado, quando M é estritamente contido em V, o CP1-ml pode ser um enfraquecimento de seu CP1 correspondente. Isso acontece porque cada vez que a rota r sai de M, a variável $state$ é redefinida para zero, potencialmente diminuindo o coeficiente retornado. A vantagem potencial do CP1-ml sobre o CP1 é o seu impacto consideravelmente reduzido sobre o algoritmo de rotulagem utilizado na geração de colunas, quando $|M| << |V|$. No entanto, usando um ajuste dinâmico dos conjuntos de memória, os CP1-ml ainda podem alcancar os mesmos limites duais obtidos com os CP1.

3. Encontrando Multiplicadores Ótimos

No contexto de geração de colunas, não é suficiente separar cortes que são definidos apenas para as variáveis presentes no PL do mestre restrito corrente, uma vez que outras variáveis, atualmente não incluídas, serão geradas pelo pricing. Isso significa que cada corte precisa ter coeficientes bem definidos para cada variável possível de ser inserida. Dessa forma, para obtenção de bons cortes de Posto 1 definidos sobre um subconjunto pequeno de restrições da FPC, precisamos investigar o seguinte poliedro, denotado aqui como o Poliedro Completo de Partição de Conjuntos (PCPC) de ordem n (PCPC(n)) e definido como:

$$PCPC(n) = \text{Conv}\left\{ \sum_{j=1}^{2^n-1} b_j \lambda_j = e_n,\ \lambda \in \{0,1\}^{2^n-1} \right\},$$

onde e_n é p vetor unitário n-dimensional e b_j é o vetor n-dimensional que codifica a representação binária do número j. Por exemplo, se $n = 3$ então $b_1 = (001), b_2 = (010), b_3 = (011), b_4 =$

$(100), b_5 = (101), b_6 = (110), b_7 = (111)$. Na sequencia, descrevemos cada passo da metodologia computacional de estudo do $PCPC(n)$, $2 \leq n \leq 5$.

1. Inicialmente enumeramos o conjunto de pontos binários $P(n)$ do $PCPC$. $P(n)$ é então submetido ao software PORTA (Christof e Löbel [2012]) para computar a representação mínima de $PCPC$ como um conjunto de igualdades e desigualdades. Seja $F(n)$ o conjunto de desigualdades computadas, excluindo algumas que são equivalentes a $\lambda_j \geq 0$ (facetas triviais), para algum $j, 1 \leq j \leq 2^n - 1$. Mais precisamente, $F(n)$ contém pares (π, π_0)

   ($\pi \in \mathbb{R}^n$ e $\pi_0 \in \mathbb{R}$ representam a desigualdade $\pi^\top \lambda \leq \pi_0$)

   que definem as facetas não triviais de $PCPC$. As desigualdades geradas pelo PORTA já são normalizada de modo que os coeficientes (π, π_0) são todos inteiros não negativos. Todas as facetas não triviais para o $PCPC$ podem ser expressas nesse formato.

   Alguns detalhes sobre os conjuntos:

   - $P(2) = \{(110), (001)\}$ e $F(2) = \emptyset$ ($PCPC(2)$ é definido por duas igualdades e por desigualdades triviais). Nenhum corte pode ser obtido através de somente duas restrições da FPC.
   - $P(3) = \{(1101000), (1000010), (0100100), (0011000), (0000001)\}$ e $F(3) = \{((0010111), 1)\}$. O corte correspondente à $\lambda_3 + \lambda_5 + \lambda_6 + \lambda_7 \leq 1$ é facilmente identificável como um SRC, obtido com o vetor de multiplicadores $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. Nenhum outro corte interessante pode ser obtido a partir de três restrições do FPC.
   - $|P(4)| = 15$ e $|F(4)| = 8$. Ainda é fácil mostrar que todas as desigualdades em $F(4)$ são cortes de posto 1. Quatro deles podem ser obtidos usando multiplicadores $(0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ e suas permutações, de modo que os mesmos são 3SRCs. As quatro desigualdades restantes são cortes de posto 1 obtidos com os multiplicadores $(\frac{2}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ e suas permutações. Essa última família de cortes domina os 4SRCs usados em Pecin et al. [2014]. De fato, pode-se notar que as 4SRCs são bastante fracas: são a soma dessas 8 facetas.
   - $|P(5)| = 52$ e $|F(5)| = 294$. As desigualdades em $F(5)$ foram analisadas e são descritas a seguir.

2. Para reduzir a análise de facetas não-triviais para um número muito menor do que as originalmente geradas pelo PORTA, deve-se identificar quais facetas pertencem ao mesmo grupo de simetria. Considere um ponto binário λ de $PCPC(n)$, e uma permutação σ de $(1, \ldots, n)$, um outro ponto λ' é simétrico a λ com respeito a σ se para cada j e k temos que $b_{k\sigma_i} = b_{ji}$ $(i = 1, \ldots, n)$, $\lambda'_k = \lambda_j$. Por exemplo, a aplicação da permutação $\sigma = (1, 3, 2)$ ao ponto binário $\lambda = (0011000)$ de $P(3)$ gera o ponto binário $\lambda' = (0100100)$, uma vez que $j = 3$ $(b_j = (110))$ corresponde a $k = 5$ $(b_k = (101))$, e $j = 4$ $(b_j = (001))$ corresponde a $k = 2$ $(b_k = (010))$. Como resultado, a mesma família de transformações pode ser aplicada aos coeficientes das variáveis λ_j em uma faceta, produzindo cortes simétricos. Duas facetas pertencem ao mesmo grupo de simetria se elas tem o mesmo lado direito e existe uma permutação σ que mapeia o vetor de coeficientes π de uma faceta para a outra. De modo complementar a essa checagem direta, é importante checar se uma faceta pode ser obtida de outra após a aplicação da permutação σ e a posterior multiplicação por uma constante e adição de uma combinação linear das igualdades que definem $PCPC(n)$. Isso pode ser checado pela resolução de um sistema de equações lineares.

3. Para cada desigualdade (π, π_0) em $F(5)$ (depois da remoção das desigualdades simétricas), resolvemos o sequinte Programa Linear, onde variáveis u_i and f_j representam, respectiva-

mente, o i-ésimo multiplicador que define o corte e a parte fracionária do coeficiente λ_j no corte (ou o lado direito do corte para $j = 0$) antes do arredondamento para baixo:

$$\begin{aligned}
\text{minimize:} \quad & \sum_{i=0}^{n} u_i \\
\text{sujeito a:} \quad & \sum_{i=1}^{n} u_i b_{ji} = \pi_j + f_j \;\; \forall j \in \{1, \ldots, 2^n - 1\} \\
& \sum_{i=1}^{n} u_i = \pi_0 + f_0 \\
& 0 \leq u_i \leq 1 - \epsilon \;\; \forall i \in \{1, \ldots, n\} \\
& 0 \leq f_j \leq 1 - \epsilon \;\; \forall j \in \{1, \ldots, 2^n - 1\}
\end{aligned}$$

Esse PL é equivalente ao problema de Fischetti e Lodi [2007] com algumas variáveis fixas. Assim como no trabalho anteriormente mencionado, usamos a tolerência $\epsilon = 0.01$.

4. Para as facetas provadamente de posto 1, de modo a expressar os multiplicadores obtidos como frações com pequenos denominadores, resolvemos novamente a formulação anterior substituindo cada u_i por v_i/D onde D é um denominador a se testar e $v_i \in \{0, \ldots, D-1\}$ é uma variável inteira. O Problema de Programação Inteira Mista (PIM) resultante é resolvido para valores progressivamente maiores de D iniciando de 2 até a obtenção de uma solução factível. Felizmente, o maior valor necessário foi $D = 5$.

Os cortes de posto 1 considerados em Petersen et al. (inequação 4) foram separados usando o PIM proposto em Fischetti and Lodi Fischetti e Lodi [2007], denotado aqui como CG-SEP. Em nosso trabalho, restringimos o espaço de busca a multiplicadores úteis descobertos neste trabalho.

As descobertas sobre $PCPC(5)$ são detalhadas nos próximos parágrafos.

Considerando o $PCPC(5)$, 294 facetas não triviais existem, 117 das quais podem ser geradas com cortes de posto 1, considerando multiplicadores de `5,1SRCs` and `5,2SRCs`, bem como as seguintes famílias (denotadas pelos vetores de multiplicadores correspondentes):

$$\left(\frac{2}{4}, \frac{2}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) \tag{6}$$

$$\left(\frac{3}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) \tag{7}$$

$$\left(\frac{3}{5}, \frac{2}{5}, \frac{2}{5}, \frac{1}{5}, \frac{1}{5}\right) \tag{8}$$

$$\left(\frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}\right) \tag{9}$$

$$\left(\frac{3}{4}, \frac{3}{4}, \frac{2}{4}, \frac{2}{4}, \frac{1}{4}\right) \tag{10}$$

Esses multiplicadores e suas permutações produzem cliques e ciclos ímpares reforçados. Deve-se enfatizar que a separação direta de cliques e ciclos ímpares iria destruir a estrutura do pricing, que é mantida intacta com o uso de cortes de rank 1.

Outra questão interessante é se existem multiplicadores capazes de gerar cortes não definidores de facetas mas que ainda assim sejam úteis para melhorar o limite inferior para C até 5. Através da geração de todos os pontos extremos do poliedro definido por nossas facetas de posto

1, denotado aqui por 1-$CSPP(n)$, e a execução da separação CGSEP sobre essa formulação descobrimos que esses multiplicadores *não* existem, de modo que os multiplicadores descobertos são, de fato, os únicos úteis para produzir qualquer corte violado de posto 1. Os pontos fracionários do 1-$CSPP(n)$ pertencem a um dos dois grupos de simetria: os 60 pontos simétricos a

$$\left(0, \frac{1}{4}, 0, \frac{1}{4}, \frac{1}{4}, 0, \frac{1}{4}, \frac{1}{4}, 0, \frac{1}{4}, 0, \frac{1}{4}, 0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, 0, 0, 0, 0, 0, \frac{1}{4}, 0, 0, 0, 0, 0, 0\right),$$

e os 12 pontos simétricos a

$$\left(\frac{2}{11}, \frac{2}{11}, 0, \frac{2}{11}, 0, \frac{3}{11}, \frac{1}{11}, \frac{2}{11}, \frac{3}{11}, 0, \frac{1}{11}, \frac{3}{11}, 0, 0, 0,\right.$$
$$\left.\frac{2}{11}, \frac{3}{11}, \frac{3}{11}, 0, 0, \frac{1}{11}, 0, 0, 0, 0, \frac{1}{11}, 0, \frac{1}{11}, 0, 0, 0\right).$$

4. Experimentos computacionais

Um algoritmo de separação para os novos CP1 (na verdade, CP1-ml) foi implementado e adicionado ao código BCP para o PRVC descrito em [Pecin et al., 2014]. Experimentos foram realizados para determinar a eficiência desses cortes, medindo-se os intervalos de dualidade (*duality gaps*). A Tabela 1 apresenta os *gaps* médios obtidos pelos seguintes mecanismos de obtenção de limites duais sucessivamente mais fortes: a relaxação linear da FPC, essa relaxacção mais os cortes definidos sobre as variáveis de aresta (portanto, cortes robustos) usados em [Fukasawa et al., 2006], mais 3SRCs, mais 4SRCs e 5SRCs (o mecanimo de obtenção de limites usado em [Pecin et al., 2014]), e, finalmente, mais os CP1 recentemente descobertos. Como pode ser visto nas últimas duas linhas, os novos cortes removeram 30% do *gap* residual. A separação destes cortes é eficiente porque usa os multiplicadores pré-calculados apresentados na seção anterior.

	Gap(%)
Somente GC (rotas elementares)	2.63
+ cortes robustos	0.98
+ 3SRCs	0.35
+ 4SRCs + 5SRCs	0.24
CG1s com até 5 linhas	0.17

Os limites duais mais fortes no código BCP melhorado com os novos CP1 permitiu, pela primeira vez, encontrar a melhor solução para a instância Golden 20, com 420 clientes, como mostrado na Figura 1. As melhores soluções conhecidas obtidas pelos recentes métodos heurísticos foram: [Vidal et al., 2012] 1.818,32, [Groër et al., 2011] 1.818,25, [Jin et al., 2014] 1.817,89 e [Liu e Li, 2014] 1.817,86. O BCP original (sem os cortes de Posto 1) utilizado encontra o limite inferior de 1814,4. Com os novos cortes de Posto 1 esse limite aumentou para 1.815,0. A árvore de busca gerou 370 nós, o tempo total de CPU gasto foi de 7,1 dias em um único núcleo de um processador de 3.30GHz i7-3960X. O limite inferior melhorado na raiz foi decisivo, estima-se que sem os novos cortes de Posto 1 o tempo de solução seria uma ordem de grandeza maior.

Referências

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.

Balinski, M. e Quandt, R. (1964). On an integer program for a delivery problem. *Operations Research*, 12(2):300–304.

Christof, T. e Löbel, A. (2012). Porta–polyhedron representation transformation algorithm, 2008. *URL: http://www. zib. de/Optimization/Software/Porta*.

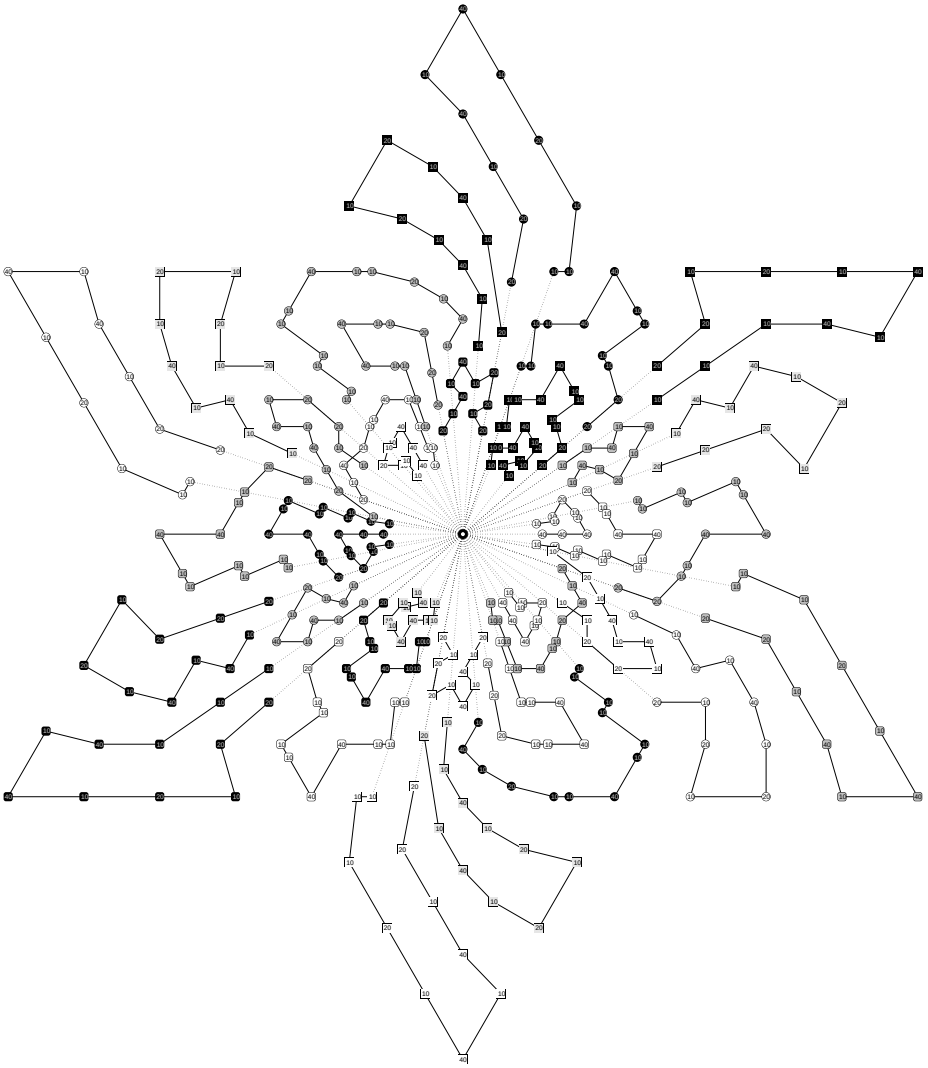

Figura 1: Solução ótima da instância Golden 20

Chvátal, V. (1973). Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete mathematics*, 4(4):305–337.

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*, 12:129–146.

Desaulniers, G., cois Lessard, F., e Hadjar, A. (2008). Tabu search, generalized k-path inequalities, and partial elementarity for the vehicle routing problem with time windows. 42(3):387–404.

Fischetti, M. e Lodi, A. (2007). Optimizing over the first chvátal closure. *Mathematical Programming*, 110(1):3–20.

Fukasawa, R., Longo, H., Lysgaard, J., Poggi de Aragão, M., Reis, M., Uchoa, E., e Werneck, R. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical programming*, 106(3):491–511.

Groër, C., Golden, B., e Wasil, E. (2011). A parallel algorithm for the vehicle routing problem. *INFORMS Journal on Computing*, 23(2):315–330.

Grötschel, M., Lovász, L., e Schrijver, A. (2012). *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media.

Jepsen, M., Petersen, B., Spoorendonk, S., e Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511.

Jin, J., Crainic, T. G., e Løkketangen, A. (2014). A cooperative parallel metaheuristic for the capacitated vehicle routing problem. *Computers & Operations Research*, 44:33–41.

Kallehauge, B., Larsen, J., e Madsen, O. (2006). Lagrangian duality applied to the vehicle routing problem with time windows. 33(5):1464–1487.

Kohl, N., Desrosiers, J., Madsen, O. B., Solomon, M. M., e cois Soumis, F. (1999). 2-Path cuts for the vehicle routing problem with time windows. 33(1):101–116.

Liu, W. e Li, X. (2014). A problem-reduction evolutionary algorithm for solving the capacitated vehicle routing problem. *Mathematical Problems in Engineering*, 501:165476.

Pecin, D., Contardo, C., Desaulniers, G., e Uchoa, E. New enhancements for the exact solution of the vehicle routing problem with time windows, institution = Les cahiers du GERAD, number = G-2016-13, year = 2016. Technical report.

Pecin, D., Pessoa, A., Poggi, M., e Uchoa, E. (2014). Improved branch-cut-and-price for capacitated vehicle routing. In *Integer Programming and Combinatorial Optimization*, p. 393–403. Springer.

Petersen, B., Pisinger, D., e Spoorendonk, S. (2008). Chvátal-gomory rank-1 cuts used in a dantzig-wolfe decomposition of the vehicle routing problem with time windows. In *The Vehicle Routing Problem: Latest Advances and New Challenges*, p. 397–419. Springer.

Poggi de Aragão, M. e Uchoa, E. (2003). Integer program reformulation for robust branch-and-cut-and-price. In Wolsey, L., editor, *Annals of Mathematical Programming in Rio*, p. 56–61, Búzios, Brazil.

Pugliese, L. D. P. e Guerriero, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200.

Røpke, S. (2012). Branching decisions in branch-and-cut-and-price algorithms for vehicle routing problems. *Presentation in Column Generation 2012*.

Spoorendonk, S. e Desaulniers, G. (2010). Clique inequalities applied to the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 48(1):53–67.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., e Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624.