

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

**Um estudo do método da geração de colunas
em programação inteira: uma aplicação ao
problema do separador de vértices**

Edna A. Hoshino Cid C. de Souza

Technical Report - IC-07-003 - Relatório Técnico

February - 2007 - Fevereiro

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

Um estudo do método da geração de colunas em programação inteira: uma aplicação ao problema do separador de vértices

Edna Ayako Hoshino*

Cid Carvalho de Souza†

Resumo

Neste relatório é feita uma investigação experimental sobre a aplicação do método da geração de colunas ao problema do separador de vértices. Os resultados computacionais são reportados para discutir diferentes aspectos de implementação e técnicas para acelerar o processo de geração de colunas. São abordados problemas de convergência, de oscilação dos valores dos duais ótimos e de degenerescência. Dois métodos de decomposição são utilizados: decomposição de Dantzig-Wolfe e mestre explícito.

1 Introdução

Dado um grafo conexo não-orientado $G = (V, E)$, um inteiro $b \leq |V|$ e um custo inteiro $c_i \geq 0$ associado a cada vértice i em V , o **problema do separador de vértices**, ou *vertex separator problem* (VSP), consiste em encontrar uma partição de V em subconjuntos não-vazios A , B e C tal que (i) os vértices em A não sejam adjacentes a vértices em B , (ii) $\max\{|A|, |B|\} \leq b$ e (iii) $\sum(c_i : i \in C)$ seja mínima quando sujeito a (i) e (ii). Os subconjuntos A e B são chamados **shore** e C é denominado **separador**.

O problema do separador de vértices é NP-difícil. Balas e de Souza [BdS05a] estudaram algumas classes de desigualdades válidas e as condições em que elas definem facetas para o poliedro do VSP. Resultados computacionais da aplicação delas em um algoritmo *branch-and-cut*, também, são reportados [BdS05b]. Borndörfer, Ferreira e Martin, também, propuseram um algoritmo *branch-and-cut*, mas com uma família diferente de cortes e para um problema mais geral que o VSP, que procura por um separador para um número fixo β de *shores* (o VSP é um caso especial em que $\beta = 2$). O objetivo deste estudo consiste em avaliar o desempenho de algoritmos de geração de colunas na resolução do VSP. Em particular, este relatório visa reportar os experimentos computacionais do uso de algumas técnicas para acelerar a convergência de algoritmos de geração de colunas, utilizando-se dois métodos de decomposição, a decomposição clássica de Dantzig-Wolfe e a decomposição de mestre explícito, no VSP. Os experimentos mostraram que o método da geração de colunas não é adequado para a resolução do VSP, pelo menos com o uso das técnicas de aceleração

*Departamento de Computação e Estatística, Universidade Federal de Mato Grosso do Sul. Este autor tem suporte financeiro da Capes, através de bolsa PICDT.

†Instituto de Computação, Universidade Estadual de Campinas. Pesquisa apoiada pelo CNPq (Processo No. 30773/2004-3) e FAPESP (Processo No. 03/09925-5).

de convergência e nas formulações PLI selecionadas. Por outro lado, as experiências aqui reportadas podem contribuir para a aplicação de geração de colunas em outros problemas de PLI.

As notações adotadas no texto são reportadas em seguida. Seja i um vértice em V . A vizinhança de i em V é denotada por $N(i)$, ou seja, $N(i) = \{j \in V \mid ij \in E\}$. Se $S \subseteq V$ então $N(S)$ representa a vizinhança de S , isto é, $N(S) = \cup_{i \in S} N(i)$. Seja (A, B, C) uma solução viável do VSP. Denota-se por C_A, C_B e C_N os subconjuntos do separador C definidos da seguinte forma: $C_A = C \cap N(A)$, $C_B = C \cap N(B)$ e $C_N = C \setminus (C_A \cup C_B)$. Se (P) é um programa linear $v(P)$ é o valor ótimo de (P) .

O relatório está organizado em cinco seções. A segunda seção apresenta as formulações por programação linear inteira (PLI) para o VSP. Uma breve revisão do método da geração de colunas e dos métodos de decomposição de PLI está reunida na Seção 3. Nesta seção, também, são discutidos alguns problemas relacionados a algoritmos de geração de colunas e algumas estratégias que podem ser incorporadas nestes algoritmos. A Seção 4 discute a aplicação destes métodos para a resolução do VSP. Algoritmos de geração de colunas, em geral, apresentam convergência lenta. Técnicas para acelerar a convergência e, os resultados computacionais da aplicação destas técnicas nos algoritmos de geração de colunas para resolver o VSP, também são discutidos na Seção 4. Por fim, na última seção, comentários finais e conclusões dos experimentos são dados.

2 Formulação por programação linear inteira

Seja $G = (V, E)$ um grafo conexo não-orientado. Uma formulação por programação linear inteira (PLI) para o VSP, proposta por Balas e de Souza [BdS05b], utiliza as seguintes variáveis de decisão associadas a cada vértice i em V :

$$u_{i1} = \begin{cases} 1 & , \text{ se o vértice } i \text{ é associado ao } \textit{shore} A \\ 0 & , \text{ caso contrário.} \end{cases}$$

$$u_{i2} = \begin{cases} 1 & , \text{ se o vértice } i \text{ é associado ao } \textit{shore} B \\ 0 & , \text{ caso contrário.} \end{cases}$$

Esta formulação, denominada (F1), é apresentada em seguida:

$$\begin{aligned}
\text{(F1)} \quad & \max \sum_{i \in V} c_i u_{i1} + c_i u_{i2} \\
\text{sujeito a} \quad & u_{i1} + u_{i2} \leq 1, \forall i \in V & (2.1) \\
& u_{i1} + u_{j2} \leq 1, \forall ij \in E & (2.2) \\
& u_{j1} + u_{i2} \leq 1, \forall ij \in E & (2.3) \\
& \sum_{i \in V} u_{ik} \leq b, k = 1, 2 & (2.4) \\
& \sum_{i \in V} u_{ik} \geq 1, k = 1, 2 & (2.5) \\
& u_{ik} \geq 0, k = 1, 2 \text{ e } \forall i \in V \\
& u_{i1} \text{ inteiro}, \forall i \in V
\end{aligned}$$

As restrições (2.1) garantem que os *shore* A e B seja disjuntos. As restrições (2.2) e (2.3) garantem que os *shores* A e B não sejam adjacentes, enquanto (2.4) e (2.5) impõem restrições de cardinalidade aos *shores*.

2.1 Uma formulação compacta

Uma formulação alternativa para o VSP inclui novas variáveis de decisão para representar os subconjuntos C_A e C_B .

$$z_{i1} = \begin{cases} 1 & \text{, se o vértice } i \text{ é associado à partição } C_A \\ 0 & \text{, caso contrário.} \end{cases}$$

$$z_{i2} = \begin{cases} 1 & \text{, se o vértice } i \text{ é associado à partição } C_B \\ 0 & \text{, caso contrário.} \end{cases}$$

$$(F2) \quad \max \sum_{i \in V} c_i u_{i1} + c_i u_{i2}$$

$$\text{sujeito a} \quad 2u_{i1} + 2u_{i2} + z_{i1} + z_{i2} \leq 2, \forall i \in V \quad (2.6)$$

$$u_{ik} + z_{ik} - u_{jk} \geq 0, k = 1, 2 \text{ e } \forall ij \in E \quad (2.7)$$

$$u_{jk} + z_{jk} - u_{ik} \geq 0, k = 1, 2 \text{ e } \forall ij \in E \quad (2.8)$$

$$u_{ik} + z_{ik} \leq 1, k = 1, 2 \text{ e } \forall i \in V \quad (2.9)$$

$$\sum_{i \in V} u_{ik} \leq b, k = 1, 2 \quad (2.10)$$

$$\sum_{i \in V} u_{ik} \geq 1, k = 1, 2 \quad (2.11)$$

$$u_{ik} \geq 0, \forall i \in V$$

$$z_{ik} \geq 0, \forall i \in V$$

$$u_{i1} \text{ inteiro}, \forall i \in V$$

As restrições (2.6) garantem que os *shores* A e B sejam disjuntos, assim como $A \cap C_B = \emptyset$, $A \cap C_A = \emptyset$, $B \cap C_A = \emptyset$ e $B \cap C_B = \emptyset$. As restrições (2.7) e (2.8) garantem que a vizinhança dos elementos de um shore estejam no mesmo shore ou no separador.

3 Revisão dos métodos utilizados

3.1 Método da geração de colunas

O método da geração de colunas foi introduzido por Dantzig e Wolfe na década de 60 para a resolução de problemas de programação linear de grande escala [DW60]. Desde a aplicação clássica do método por Gilmore e Gomory na resolução do problema de corte e empacotamento e com o sucesso da incorporação do método num algoritmo de *branch-and-bound* na resolução do VRP por Desrosiers, Soumis e Desrochers [DSD84], o método tem sido bastante estudado e utilizado na resolução de problemas de programação linear inteira [BJN⁺98, VW96, Sav97, dAU03, LD, RG, FLL⁺06].

Considere o seguinte programa linear, chamado **problema mestre** (PM) no contexto de geração de colunas:

$$\begin{aligned} & \max \sum_{j \in N} c_j x_j \\ \text{sujeito a} \quad & \sum_{j \in N} a_{ij} x_j \leq b_i, \forall i \in M \\ & x_j \geq 0, \forall j \in N. \end{aligned} \quad (3.1)$$

O passo de **pricing** em um algoritmo *Simplex* na resolução do problema mestre consiste em encontrar $j = \arg \max \{\bar{c}_j = c_j - \sum_{i \in M} a_{ij} \pi_i\}$, onde π é a variável dual associada às restrições (3.1). Portanto, o passo de *pricing* toma tempo proporcional a $|N|$.

Métodos de geração de colunas são aplicados quando $|N|$ é muito grande, possivelmente exponencial no tamanho da entrada do problema. A idéia deste método consiste em resolver o problema mestre restrito a um subconjunto menor de colunas, $\tilde{N} \subseteq N$. Este problema menor é chamado **problema mestre restrito** (PMR). Obtido o valor das variáveis duais $\tilde{\pi}$ com a otimização do PMR, o método da geração de colunas realiza o passo de *pricing* resolvendo-se um outro problema de otimização ao invés de avaliar, explicitamente, o custo reduzido \bar{c}_j , para todo $j \in N \setminus \tilde{N}$. Para tanto, as colunas a_j , $j \in N$, devem estar associadas a elementos de um conjunto Λ e deve existir uma função $c : \Lambda \mapsto Q$, que compute o valor de c_j . O passo de *pricing* pode, então, ser realizado resolvendo-se o seguinte problema de otimização $\bar{c}^* = \max\{c(a) - \tilde{\pi}a \mid a \in \Lambda\}$. Quando $\bar{c}^* \leq 0$, $\tilde{\pi}$ é dual viável, em relação ao PM, e portanto, a solução do PMR é também solução do PM. Caso contrário, a coluna a gerada pelo *pricing* (isto é, $a = \arg \max\{c(a) - \tilde{\pi}a \mid a \in \Lambda\}$) é incluída em \tilde{N} e o PMR é novamente otimizado. A convergência do método da geração de colunas é garantida pelo fato de que cada coluna $a \in \Lambda$ é gerada, no máximo, uma vez pelo *pricing*, já que todas as colunas em \tilde{N} têm custo reduzido negativo após a otimização do PMR.

A princípio, pode parecer que determinar \bar{c}^* seja tão difícil quanto avaliar o custo reduzido de cada coluna em $N \setminus \tilde{N}$. No entanto, para tomar vantagem do método, o modelo de programação linear do problema mestre é selecionado de tal forma que as colunas $a \in \Lambda$ possuam propriedades específicas da aplicação que permitam caracterizar Λ por um número linear de restrições (usualmente, da ordem de $|M|$ e muito menor que $|N|$). Em geral, $a \in \Lambda$, representam objetos combinatórios como árvores, caminhos e outras estruturas bem definidas.

Nas próximas subseções serão discutidos métodos de decomposição de PLI, cujos problemas de otimização resultantes são resolvidos por algoritmos de geração de colunas.

3.2 Decomposição de Dantzig-Wolfe

Considere, inicialmente, o seguinte programa linear,

$$(P) \quad \max \sum_{j \in N} c_j x_j$$

$$\text{sujeito a} \quad \sum_{j \in N} a_{ij} x_j \leq b_i, \forall i \in M \tag{3.2}$$

$$x \in P = \{x \in \mathbb{R}^{|N|} \mid Dx \leq d, x \geq 0\} \tag{3.3}$$

Sejam $n = |N|$ e $m = |M|$ lineares no tamanho da entrada do problema. No contexto da decomposição de Dantzig-Wolfe, o programa linear acima é chamado **formulação compacta**. Considere P não-vazio e limitado, para simplificar a explanação (o método da decomposição de Dantzig-Wolfe também se aplica nos demais casos [DW60]). Sejam Q o conjunto dos pontos extremos de P e p_q , para $q \in Q$, o vetor característico de q . Portanto, cada ponto $x \in P$ pode ser representado pela combinação convexa dos pontos em Q , ou seja,

$$x = \sum_{q \in Q} p_q \lambda_q, \sum_{q \in Q} \lambda_q = 1, \lambda \in R_+^{|Q|}. \tag{3.4}$$

A substituição de x na formulação compacta por (3.4) e a aplicação das transformações lineares $c^q = cp_q$ e $a^q = Ap_q$, dá o seguinte programa equivalente, chamado **formulação extensiva**:

$$\begin{aligned} & \max \sum_{q \in Q} c^q \lambda_q \\ \text{sujeito a} \quad & \sum_{q \in Q} a_i^q \lambda_q \leq b_i, \forall i \in M \end{aligned} \quad (3.5)$$

$$\sum_{q \in Q} \lambda_q = 1 \quad (3.6)$$

$$\lambda_q \geq 0, \forall q \in Q.$$

A restrição (3.6) é chamada **restrição de convexidade**. Uma vez que o número de colunas na formulação extensiva é muito grande, o método da geração de colunas é utilizado na resolução do problema, que é um caso bastante particular de problema mestre. Nele, as colunas são dadas por pontos extremos do poliedro convexo P . Sejam $\tilde{\pi}, \tilde{\mu}$ os duais ótimos obtidos com a otimização do PMR correspondente (μ é o dual associado à restrição de convexidade). Devido às transformações lineares acima, a determinação de $\bar{c}^* = \max_{q \in Q} \{c^q - \tilde{\pi}a^q - \tilde{\mu}\}$ equivale a resolver o seguinte programa linear:

$$\bar{c}^* = \max\{(c - \tilde{\pi}A)x - \tilde{\mu} \mid Dx \leq d, x \geq 0\}.$$

Agora, considere o caso em que $x \in X = P \cap \mathbb{Z}_+$ na formulação compacta, ou seja,

$$\begin{aligned} & \max \sum_{j \in N} c_j x_j \\ \text{sujeito a} \quad & \sum_{j \in N} a_{ij} x_j \leq b_i, \forall i \in M \\ & x \in X = P \cap \mathbb{Z}_+ \end{aligned} \quad (3.7)$$

Considerando-se $\text{conv}(X)$ no lugar de X , obtém-se um programa equivalente, no qual aplicando-se a decomposição de Dantzig-Wolfe obtém-se a seguinte formulação extensiva:

$$\begin{aligned} & \max \sum_{q \in Q} c^q \lambda_q \\ \text{sujeito a} \quad & \sum_{q \in Q} a_i^q \lambda_q \leq b_i, \forall i \in M \end{aligned} \quad (3.8)$$

$$\sum_{q \in Q} \lambda_q = 1 \quad (3.9)$$

$$\lambda_q \geq 0, \forall q \in Q$$

$$x = \sum_{q \in Q} p_q \lambda_q \quad (3.10)$$

$$x \in \mathbb{Z}_+^n.$$

Relaxando-se a restrição de integralidade de x , a restrição (3.10) torna-se desnecessária. O problema mestre resultante,

$$\begin{aligned} \text{(DW)} \quad & \max \sum_{q \in Q} c^q \lambda_q \\ \text{sujeito a} \quad & \sum_{q \in Q} a_i^q \lambda_q \leq b_i, \forall i \in M \end{aligned} \tag{3.11}$$

$$\begin{aligned} & \sum_{q \in Q} \lambda_q = 1 \\ & \lambda_q \geq 0, \forall q \in Q, \end{aligned} \tag{3.12}$$

pode ser resolvido por geração de colunas e o valor da solução ótima obtida dá um limitante superior para o problema inteiro. Geoffrion tem mostrado que este limitante, obtido pela decomposição de Dantzig-Wolfe, é sempre menor ou igual ao limitante dado pela relaxação linear do problema compacto. De fato, Geoffrion tem mostrado que o limitante obtido pela decomposição de Dantzig-Wolfe coincide com o valor do limitante dado por outros dois métodos de decomposição, relaxação lagrangeana e algoritmos de planos de corte. Portanto, a decomposição de Dantzig-Wolfe pode ser utilizada para obter limitantes mais apertados que a relaxação linear, útil por exemplo em algoritmos de *Branch-and-Bound*. Também, tem sido mostrado que o limitante superior dado pela decomposição de Dantzig-Wolfe coincide com o valor da relaxação linear do problema compacto quando $P = \text{conv}(X)$. Isso decorre da observação de que, enquanto a relaxação linear resolve o problema compacto com $x \in P$, o método da geração de colunas, implicitamente, resolve o problema compacto com $x \in \text{conv}(X)$. Portanto, escolhas adequadas de se decompor um problema compacto na decomposição de Dantzig-Wolfe devem ser consideradas para se evitar um limitante igual à relaxação linear.

3.3 Decomposição por Mestre Explícito

Considere a formulação compacta (P) da subseção anterior. O mestre explícito proposto por Poggi e Uchoa [dAU03], consiste em manter as variáveis originais x da formulação compacta no problema mestre e manter, explicitamente, a relação entre x e os pontos em Q . O seguinte é o problema mestre, segundo a decomposição por mestre explícito, já com a restrição de integralidade relaxada:

$$\begin{aligned}
(\text{ME}) \quad & \max \sum_{j \in N} c_j x_j \\
\text{sujeito a} \quad & \sum_{j \in N} a_{ij} x_j \leq b_i, \forall i \in M
\end{aligned} \tag{3.13}$$

$$x = \sum_{q \in Q} p_q \lambda_q \tag{3.14}$$

$$\sum_{q \in Q} \lambda_q = 1 \tag{3.15}$$

$$\lambda_q \geq 0, \forall q \in Q$$

$$x_j \geq 0, \forall j \in N.$$

Sejam π, ν, μ as variáveis duais associadas às restrições (3.13), (3.14) e (3.15). O dual do problema mestre é:

$$(\text{DME}) \quad \min \sum_{i \in M} b_i \pi_i + \mu$$

$$\text{sujeito a} \quad \sum_{i \in M} a_{ij} \pi_i - \nu_j \geq c_j, \forall j \in N \tag{3.16}$$

$$\sum_{j \in N} p_{qj} \nu_j + \mu \geq 0, \forall q \in Q \tag{3.17}$$

$$\pi_i \geq 0, \forall i \in M.$$

O subproblema de pricing do mestre explícito consiste em:

$$\begin{aligned}
(\text{Pricing}) \quad & \max \quad - \sum_{j \in N} \nu_j x_j - \mu \\
\text{s.a.} \quad & Dx \leq d \\
& x \in \mathbb{Z}_+^n.
\end{aligned}$$

Uma das vantagens do mestre explícito em relação à decomposição de Dantzig-Wolfe está relacionada à fixação de variáveis por custo reduzido. Uma vez que no mestre explícito mantém-se as variáveis originais do problema, a otimização do mestre explícito já fornece o custo reduzido de cada uma delas e, então, pode-se fixar aquelas cujo custo reduzido seja maior que o *gap* de otimalidade.

Poggi e Uchoa mostraram que os limitantes dados pela decomposição de Dantzig-Wolfe e pelo mestre explícito se coincidem.

Teorema 3.1 (Poggi e Uchoa [dAU03]) $v(DW) = v(ME)$.

Prova: Sejam λ^* e (π^*, μ^*) soluções ótimas primais e duais, respectivamente, do problema mestre (DW).

É fácil checar que $(\lambda^*, x^* = \sum_{q \in Q} p_q \lambda_q^*)$ é primal viável em (ME). Resta mostrar que $(\pi^*, \nu^* = \pi^* A - c, \mu^*)$ é dual viável em (ME).

Como (π^*, μ^*) é uma solução dual ótima de (DW) segue que

$$\begin{aligned} a^q \pi^* + \mu^* &\geq c^q, \forall q \in Q \\ \Rightarrow A p_q \pi^* + \mu^* &\geq c p_q, \forall q \in Q \\ \Rightarrow p_q (A \pi^* - c) + \mu^* &\geq 0, \forall q \in Q. \end{aligned}$$

Considerando $\nu^* = A \pi^* - c$, então

$$\begin{cases} p_q \nu^* + \mu^* \geq 0, \forall q \in Q \\ A \pi^* - \nu^* = c \end{cases}$$

e, portanto, (π^*, ν^*, μ^*) é dual viável em (ME).

Suponha que $(\lambda^*, x^* = \sum_{q \in Q} p_q \lambda_q^*)$ não seja solução ótima de (ME), ou seja, existe solução viável (λ^o, x^o) de (ME) com $c x^o > c x^*$. Como (λ^o, x^o) é viável em (ME), tem-se que $x^o = \sum_{q \in Q} p_q \lambda_q^o$, do qual segue que λ^o é viável em (DW). Como $c \sum_{q \in Q} p_q \lambda_q^o = c x^o > c x^* = c \sum_{q \in Q} p_q \lambda_q^*$, tem-se que λ^* não é solução primal ótima de (DW).

3.4 Limitantes superiores para o problema mestre

Considere o problema mestre (DW) obtido pela decomposição de Dantzig-Wolfe discutido na Subseção 3.2. Sejam π^* e μ^* os duais ótimos do problema mestre restrito em alguma iteração do algoritmo de geração de colunas. Sejam $ZPric$ o valor ótimo do subproblema de pricing, $v(PM)$ o valor ótimo do problema mestre e $v(PMR)$ o valor ótimo do problema mestre restrito. Para fins de generalização, considere k , $k \geq 1$, o *rhs* da restrição de convexidade. Nesta subseção são discutidos limitantes superiores para o valor ótimo do problema mestre, úteis quando a convergência da geração de colunas apresenta-se lenta e pode-se abrir mão da otimalidade, por exemplo, em um algoritmo *Branch-and-Price*.

3.4.1 Limitante dual de Lasdon

Lasdon [Las70] tem mostrado que

$$UB = v(PMR) + k ZPric$$

é um limitante superior para o valor ótimo do problema mestre.

Uma vez que $ZPric$ define o custo reduzido mais positivo, tem-se que

$$\begin{aligned} ZPric &\geq c_j - \pi^* A^j - \mu^*, \forall j \in N \\ \Rightarrow c_j - \pi^* A^j - \mu^* - Zpric &\leq 0, \forall j \in N \\ \Rightarrow \pi^* A^j + \mu^* + Zpric &\geq c_j, \forall j \in N. \end{aligned}$$

Portanto, $(\pi^*, \mu^* + Zpric)$ é dual viável. Por dualidade fraca, tem-se que

$$v(PM) \leq \sum_{i \in M} b_i \pi_i^* + K \mu^* + K Zpric = v(PMR) + K Zpric,$$

logo, UB é um limitante dual para o problema mestre.

3.4.2 Limitante dual de Farley

Farley [Far90] propôs um limitante inferior para o valor ótimo de um problema mestre de minimização. A discussão que se segue estende este limitante para problemas de maximização. Como antes, considere o problema mestre dado por (DW) e seja x^* uma solução ótima para o PMR e (π^*, μ^*) seus duais ótimas. Da mesma forma que em [Far90] considere somente os casos em que $c^q \geq 0, \forall q \in Q$.

Defina $\alpha_q = a^q \pi^* + \mu^*$, para cada q em Q .

Se $\alpha_q \geq c_q, \forall q \in Q$ então (π^*, μ^*) é dual viável para PM e, então, x^* , também, é solução ótima do problema mestre. Caso contrário, seja $\frac{c_k}{\alpha_k} = \max_{q|\alpha_q < c_q} \frac{c_q}{\alpha_q}$. Note que

$$\frac{c_k}{\alpha_k} > 1. \quad (3.18)$$

Fato 3.2 $(\pi^* \frac{c_k}{\alpha_k}, \mu^* \frac{c_k}{\alpha_k})$ é dual viável.

Este fato segue das seguintes observações:

Caso 1 Se $\alpha_q < c_q$ então, por definição, $\frac{c_k}{\alpha_k} \geq \frac{c_q}{\alpha_q} \Rightarrow \alpha_q \frac{c_k}{\alpha_k} \geq c_q$;

Caso 2 Se $\alpha_q \geq c_q$ então, como $c_q \geq 0$, segue que $\alpha_q \geq 0$. Deste fato e da Equação 3.18 segue que $\alpha_q \frac{c_k}{\alpha_k} > \alpha_q \geq c_q \Rightarrow \alpha_q \frac{c_k}{\alpha_k} \geq c_q$.

Logo, para todo $q \in Q$, vale $\alpha_q \frac{c_k}{\alpha_k} \geq c_q$, ou seja,

$$\begin{aligned} (a^q \pi^* + \mu^*) \frac{c_k}{\alpha_k} &\geq c_q, \forall q \in Q \\ \Rightarrow a^q \pi^* \frac{c_k}{\alpha_k} + \mu^* \frac{c_k}{\alpha_k} &\geq c_q, \forall q \in Q. \end{aligned}$$

Portanto, $(\pi^* \frac{c_k}{\alpha_k}, \mu^* \frac{c_k}{\alpha_k})$ é dual viável em PM. Logo,

$$\begin{aligned} UB &= \sum_{i \in M} b_i \pi_i^* \frac{c_k}{\alpha_k} + K \mu^* \frac{c_k}{\alpha_k} \\ &= Z(PMR) \frac{c_k}{\alpha_k} \end{aligned}$$

define um limitante superior para o valor ótimo do problema mestre.

Obter o limitante dual de Lasdon implica em resolver um problema de otimização, cuja função objetivo é $\max_{q \in Q} c^q - \alpha_q$, que é o próprio problema de pricing, enquanto que para obter o limitante dual de Farley é necessário resolver outro problema de otimização, cuja função objetivo é $\max_{q \in Q} \frac{c^q}{\alpha_q}$. Portanto, pode ser custoso obter o limitante dual de Farley, além do fato de que, em alguns casos, pode-se tratar de um problema de otimização não-linear. Para exemplificar, considere a formulação (F2) do VSP dada na Seção 2. O problema mestre da decomposição de Dantzig-Wolfe deste modelo utiliza as seguintes transformações lineares:

$$a_i^q = 2u_{ik} + z_{ik}, q \in Q, k = 1, 2$$

e

$$c^q = \sum_{i \in V} c_i u_{ik}, q \in Q, k = 1, 2.$$

Na construção do limitante dual de Farley, $\alpha_q = \sum_{i \in V} (2u_i + z_i)\pi_i^* + \mu^*$. Portanto, para obter o limitante dual de Farley, o seguinte problema de otimização não-linear deve ser resolvido:

$$\begin{aligned} \max \quad & \frac{\sum_{i \in V} c_i u_i}{\sum_{i \in V} (2\pi_i^* u_i + \pi_i^* z_i) + \mu^*} \\ \text{s.a.} \quad & (u, z) \in Q. \end{aligned}$$

3.5 Limitante dual de Vanderbeck e Wolsey

O limitante dual definido por Vanderbeck e Wolsey é útil em uma implementação Branch-and-Price, pois o limitante é calculado considerando-se as restrições de branching incluídas na formulação de cada nó da árvore de enumeração. No nó raiz, os limitantes duais de Vanderbeck e Wolsey e de Lasdon se coincidem.

4 Aplicação dos métodos

Esta seção discute a aplicação do método da geração de colunas na resolução do VSP e algumas técnicas para acelerar o processo de geração de colunas. À medida que as técnicas são discutidas, os resultados computacionais da aplicação delas na resolução do VSP são reportados. Os experimentos computacionais foram realizados em uma máquina Pentium IV 2.66GHz com 900MB de RAM, utilizando-se a biblioteca de otimização do XPRESS¹.

O cabeçalho das tabelas desta seção tem o seguinte significado:

¹XPRESS é um produto comercial da Dash Corporation, que tem sido cedido na forma de licença acadêmica para o IC/UNICAMP.

sigla	significado
LP	valor ótimo da relaxação linear da formulação compacta
IP	valor ótimo da instância do problema <i>VSP</i>
PMR	valor da melhor solução da relaxação linear do problema mestre
UB	limitante superior para a relaxação linear do problema mestre
tPMR	tempo gasto com a otimização da relaxação do problema mestre
tPric	tempo total gasto com <i>pricing</i>
iter	número de iterações do processo de geração de colunas
cols	número de colunas geradas
PMR/LP	razão entre PMR e LP
PMR/IP	razão entre PMR e IP
LB	valor da melhor solução primal para o problema mestre
iLB	iteração em que a melhor solução primal foi gerada
totH	total de boas soluções primais geradas pela heurística primal
isH	setado se melhor primal foi gerado pela heurística primal

Em todas as tabelas, os tempos são dados em segundos.

4.1 Decomposição Dantzig-Wolfe para o VSP

A decomposição de Dantzig-Wolfe aplicada ao modelo do VSP dado pela formulação (F2) possui um gerador natural para as colunas. Neste modelo, cada coluna representa um *shore* e sua vizinhança em C . Reescrevendo-se (F2), esta decomposição fica mais evidente:

$$\begin{aligned}
 \text{(F2)} \quad & \max \sum_{i \in V} c_i u_{i1} + c_i u_{i2} \\
 \text{sujeito a} \quad & 2u_{i1} + 2u_{i2} + z_{i1} + z_{i2} \leq 2, \forall i \in V \\
 & (u_1, z_1), (u_2, z_2) \in P,
 \end{aligned} \tag{4.1}$$

onde $P = \{(u, z) \in B^{n \times n} \mid u_i + z_i - u_j \geq 0, \forall ij \in E, u_j + z_j - u_i \geq 0, \forall ij \in E, u_i + z_i \leq 1, \forall i \in V, 1 \leq \sum_{i \in V} u_i \leq b\}$.

Cada coluna (u, z) gerada pelo passo de *pricing* é adicionada ao problema mestre restrito aplicando-se a transformação linear $a_i^q = 2u_i + z_i$, para todo $i \in V$ e $q_{n+1} = 1$.

4.1.1 Problema mestre restrito

Seja $c^q = \sum_{i \in V} c_i u_i$.

O problema mestre obtido pela decomposição de Dantzig-Wolfe é:

$$\begin{aligned} \text{(PM)} \quad & \max \sum_{q \in Q} c^q \lambda_q \\ \text{sujeito a} \quad & \sum_{q \in Q} a_i^q \lambda_q \leq 2, \forall i \in V \end{aligned} \quad (4.2)$$

$$\sum_{q \in Q} \lambda_q = 2 \quad (4.3)$$

$$\lambda_q \leq 1, \forall q \in Q \quad (4.4)$$

$$\lambda_q \geq 0, \forall q \in Q$$

No problema mestre, as restrições de bound (4.4) são desnecessárias, uma vez que, em toda coluna q , $a_i^q = 2$, para algum $i \in V$, o que impede que λ_q assumam valor maior que 1.

4.1.2 Pricing

Sejam π^* e μ^* , os duais ótimos associados às restrições (4.2) e (4.3), respectivamente. O subproblema de pricing para o VSP consiste em:

$$\begin{aligned} \text{(Pricing)} \quad & -\mu^* + \max \sum_{i \in V} (c_i - \pi_i^*) u_i - \pi_i^* z_i \\ \text{sujeito a} \quad & 1 \leq \sum_{i \in V} u_i \leq b \\ & \sum_{i \in V} (u_i + z_i) \leq |V| - 1 \\ & u_i + z_i \leq 1, \forall i \in V \\ & u_i + z_i - y_j \geq 0, \forall ij \in E \\ & -z_i + \sum_{j \in N(i)} u_j \geq 0, \forall i \in V \\ & u_i, z_i \in \{0, 1\}, \forall i \in V \end{aligned} \quad (4.5)$$

Seja S um shore obtido pelo *pricing*. Devido à restrição (4.5), existe um i em $V \setminus (S \cup N(S))$. Uma vez que $N(i) \cap S = \emptyset$, segue que $\bar{S} = \{i\}$ define um shore complementar a S , isto é, S e \bar{S} definem uma solução viável para PM. Portanto, desde que $c_i > 0$, para todo $i \in V$, segue que, em (PM) pode-se substituir a restrição (4.3) por $\sum_{q \in Q} \lambda_q \leq 2$.

4.2 Decomposição do VSP por mestre explícito

Seja $x^k = (u_{1k}, u_{2k}, \dots, u_{nk}, z_{1k}, \dots, z_{nk})$, para $k = 1, 2$, e seja $A = [2I|I]$. O mestre explícito para o VSP consiste em:

$$\begin{aligned} & \max cx^1 + cx^2 \\ \text{sujeito a } & Ax^1 + Ax^2 \leq 2 \end{aligned} \tag{4.6}$$

$$\sum_{q \in Q} p_q \lambda_q - x^1 - x^2 = 0 \tag{4.7}$$

$$\sum_{q \in Q} \lambda_q = 2 \tag{4.8}$$

$$\lambda_q \leq 1, \forall q \in Q \tag{4.9}$$

$$\lambda_q \geq 0, \forall q \in Q$$

$$x \geq 0.$$

Sejam π, ν e μ as variáveis duais associadas às restrições (4.6), (4.7) e (4.8), respectivamente. O dual do mestre explícito é:

$$\begin{aligned} & \min 2\pi + 2\mu \\ \text{sujeito a } & Q\nu^1 + \mu^1 \geq 0 \\ & p_q \nu + \mu \geq 0, \forall q \in Q \\ & -\nu + A\pi \geq c \\ & \pi \geq 0 \\ & \nu, \mu \text{ qualquer.} \end{aligned}$$

As colunas em Q são geradas resolvendo-se o seguinte problema de pricing:

$$\begin{aligned} \text{(Pricing)} \quad & \max -\nu(u, z) - \mu \\ \text{sujeito a } & 1 \leq \sum_{i \in V} u_i \leq b \\ & \sum_{i \in V} (u_i + z_i) \leq |V| - 1 \\ & u_i + z_i \leq 1, \forall i \in V \\ & u_i + z_i - y_j \geq 0, \forall ij \in E \\ & -z_i + \sum_{j \in N(i)} u_j \geq 0, \forall i \in V \\ & u_i, z_i \in \{0, 1\}, \forall i \in V. \end{aligned}$$

Um limitante superior para a relaxação linear do mestre explícito pode ser obtido usando-se o limitante de Lasdon. Seja $ZPric$ o valor da solução ótima do problema de pricing. Então,

$$\begin{aligned} ZPric & \geq -\nu x - \mu, \forall x \in P \\ \Rightarrow \nu x + (ZPric + \mu) & \geq 0, \forall x \in P. \end{aligned}$$

Logo, $(\pi, \nu, ZPric + \mu)$ é dual viável e, portanto, um limitante superior para o valor ótimo do mestre explícito é

$$UB = 2\pi + 2(\mu + ZPric) = z_{PMR} + 2ZPric,$$

que é exatamente o limitante de Lasdon.

4.3 Análise da qualidade dos limitantes

Como discutido na Seção 3.2, o limitante superior dado pela decomposição de Dantzig-Wolfe pode ser mais apertado que o limitante dado pela relaxação linear da formulação compacta. Nos testes realizados nas instâncias MM-I, este limitante é cerca de 10% mais apertado que a relaxação linear.

As Figuras 1 e 2 ilustram a distância entre o limitante superior e a solução do PMR a cada iteração. Nestes testes, um limitante superior mais apertado que a relaxação linear do problema foi encontrado depois de, aproximadamente, 100 iterações.

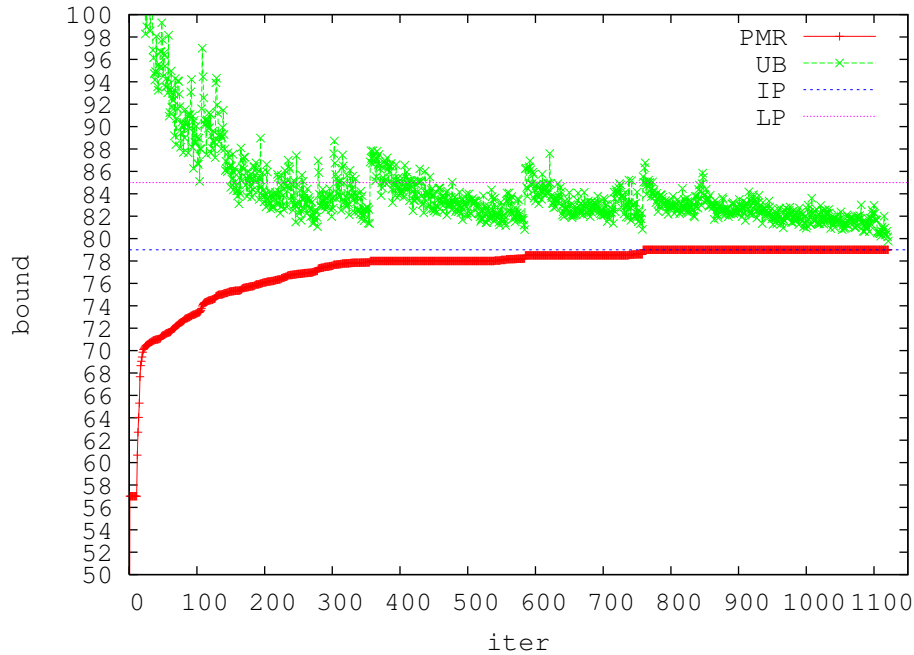


Figura 1: Instância mat.ash219.

A coluna PMR/LP na Tabela 1 representa a razão entre o valor ótimo do problema mestre e o valor ótimo da relaxação linear da formulação compacta. Pode-se notar que o limitante dual dado pela relaxação linear do problema mestre é, na maioria dos casos, mais apertado que o limitante dual dado pela relaxação linear da formulação compacta. Além disso, em muitos casos, o processo de geração de colunas também resolveu o problema inteiro.

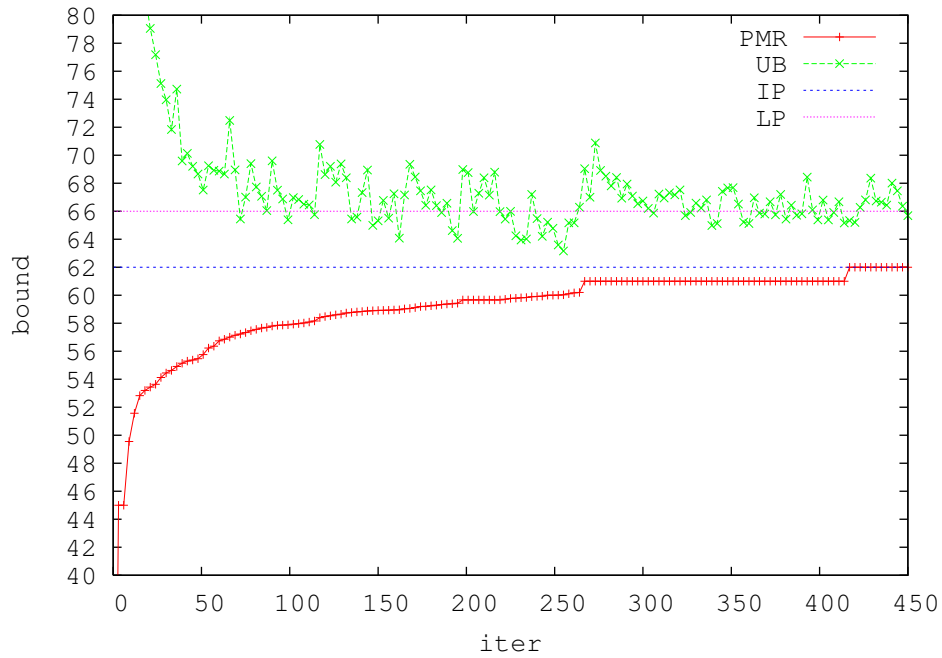


Figura 2: Instância mat.dwt66.

4.4 Pool de colunas

As Tabelas 1 e 2 ilustram o ganho no desempenho com o uso de um pool de colunas, isto é, com a adição de uma ou mais colunas de custo reduzido positivo, em contraposição à estratégia que inclui apenas a coluna de maior custo reduzido. Embora o número de colunas aumente com o uso de pool, o tempo total gasto, na média, foi reduzido em 50%.

Instancia	b	LP	IP	PMR	UB	PMR /LP	PMR /IP	tPMR
ash219	56	85.00	79.00	79.00	79.00	0.93	1.00	1880.35
dwt72	48	72.00	68.00	68.00	68.00	0.94	1.00	640.25
can62	41	62.00	56.00	56.39	56.39	0.91	1.00	586.90
dwt66	44	66.00	62.00	62.00	62.00	0.94	1.00	1348.38
bcpwr02	32	49.00	44.00	44.00	44.00	0.90	1.00	97.09
dwt59	39	59.00	51.00	51.77	51.77	0.88	1.00	966.89
bcpwr01	26	39.00	36.00	36.00	36.00	0.92	1.00	25.74

Tabela 1: Instâncias MM-I com $b=\lfloor 2n/3 \rfloor$ sem pool de colunas

Considerando que o valor ótimo do VSP é inteiro, isto é, o custo das variáveis na função objetivo do problema mestre é inteiro, apenas limitantes duais inteiros podem ser considerados. Seja UB um limitante superior para o valor ótimo do problema mestre. Neste

Instancia	b	LP	IP	PMR	UB	PMR /LP	PMR /IP	tPMR
ash219	56	85.00	79.00	79.00	79.00	0.93	1.00	1159.87
dwt72	48	72.00	68.00	68.00	68.00	0.94	1.00	321.16
can62	41	62.00	56.00	56.39	56.39	0.91	1.00	186.50
dwt66	44	66.00	62.00	62.00	62.00	0.94	1.00	768.02
bcpwr02	32	49.00	44.00	44.00	44.00	0.90	1.00	56.11
dwt59	39	59.00	51.00	51.77	51.77	0.88	1.00	249.36
bcpwr01	26	39.00	36.00	36.00	36.00	0.92	1.00	18.64

Tabela 2: Instâncias MM-I com $b=\lfloor 2n/3 \rfloor$ com pool de colunas

caso, diz-se que a otimização do problema mestre restrito pode ter um término mais cedo, uma vez que $\lfloor UB \rfloor$ é um limitante superior mais apertado que UB . A Tabela 3 resume os resultados da utilização deste limitante mais apertado nas instâncias MM-I. As colunas t_{PMR1} e t_{PMR2} indicam o tempo gasto, respectivamente, sem e com o término mais cedo, isto é, utilizando-se o piso do limitante superior. A coluna r_{Tempo} representa a razão entre o tempo gastot t_{PMR1} e t_{PMR2} . As colunas Col1 e Col2 indicam o número de colunas geradas, respectivamente, sem e com o término mais cedo. O ganho com o término mais cedo foi, em média, de 19% no tempo de processamento.

Instancia	b	tPMR1	tPMR2	rTempo	Col1	Col2
ash219	56	1159.87	1009.51	0.87	4118	3786
dwt72	48	321.16	297.42	0.98	2253	2136
can62	41	186.50	162.24	0.87	1112	1036
dwt66	44	768.02	342.59	0.47	2181	1844
bcpwr02	32	56.11	50.40	0.90	645	610
dwt59	39	249.36	218.18	0.87	1014	980
bcpwr01	26	18.64	13.24	0.71	379	283

Tabela 3: Comparação entre Instâncias MM-I com pool de colunas (1) sem e (2) com término mais cedo

Deste ponto em diante, a implementação da geração de colunas, usando a decomposição de Dantzig-Wolfe, com o pool de colunas e com término mais cedo será denominada **geração de colunas clássica**.

Outras melhorias podem ser incluídas nesta estratégia para melhorar o desempenho da geração de colunas. Uma delas consiste em manter no pool, as colunas de custo reduzido positivo geradas em iterações anteriores e selecionar um percentual delas (desde que continuem com custo reduzido positivo) para ser incluído no problema mestre restrito. Outra melhoria refere-se a remover colunas do problema mestre restrito que não são mais promissoras, ou seja, que têm custo reduzido muito negativo (maior que o *gap* de otimalidade).

4.5 Pricing parcial

Outra estratégia para diminuir os efeitos da convergência lenta, que ocorre nos métodos de geração de colunas, consiste em utilizar uma heurística para o *pricing*. Esta estratégia é conhecida como **pricing parcial** e consiste em gerar colunas de custo reduzido positivo, não necessariamente a de maior custo reduzido (que é aquela gerada pelo **pricing exato**). Quando a heurística falha, ou seja, não consegue gerar uma coluna de custo reduzido positivo, o problema de *pricing* é otimizado. Esta estratégia teve bons resultados quando aplicado ao VSP, principalmente pela qualidade das colunas geradas e por diminuir efetivamente o tempo de otimização.

A Tabela 4 resume os resultados no uso de uma heurística GRASP (*Greedy Randomized Adaptive Search Procedure*) para o pricing parcial. Nesta tabela, a coluna **tLP** refere-se ao tempo total gasto pelo solver de LP do XPRESS para a solução do PMR, excluídos o tempo gasto com o subproblema de pricing. A coluna **tHeu** mostra o tempo total gasto pelo GRASP. A coluna **tPMR** é a soma de **tLP** mais o tempo total gasto pelo solver de IP do XPRESS para a solução do problema de pricing exato. Portanto, o tempo total gasto na geração de colunas é dado por **tPMR** + **tHeu**.

Os resultados apresentados na Tabela 4 mostram um ganho de até 5 vezes no desempenho, em relação ao uso de um solver para a solução exata do subproblema de pricing. Além disso, a inclusão das colunas geradas pelo GRASP tornou possível a obtenção de soluções melhores para o PMR mais cedo. Isso implica que, pelo menos na análise preliminar, o uso da heurística não apenas ajudou a diminuir o overhead de tempo de processamento para a solução do subproblema de pricing, como também gerou colunas “melhores” (colunas que, embora não sejam a de maior custo reduzido, foram usadas na obtenção de soluções do PMR).

Na Figura 3 todos os limitantes superiores gerados são mostrados. Ela mostra a efetividade do uso do solver apenas nos casos em que a heurística GRASP falha na obtenção de colunas com custo reduzido positivo. A maioria dos limitantes gerados pelo solver não são de boa qualidade (são maiores que o valor da relaxação linear da formulação compacta). As colunas geradas pelo solver não contribuíram para acelerar a busca pelo melhor valor do PMR.

4.6 Heurística primal

A Tabela 5 ilustra o ganho no limitante primal, utilizando-se uma heurística primal simples. A heurística consiste em computar uma coluna complementar à coluna gerada pelo pricing para obter uma solução inteira para o problema mestre. A coluna (**b, c, g**) na Tabela 5 representa o percentual de cada tipo de coluna na melhor solução do problema mestre restrito. Os tipos **b, c** e **g** correspondem, respectivamente, a colunas que estavam na base inicial, que foram geradas pela heurística primal e aquelas obtidas pelo pricing. As colunas geradas pela heurística foram utilizadas na maioria das soluções ótimas do problema mestre.

Instancia	b	PMR	UB	tPMR	tLP	tHeu	Col	Iter	LB	itLB
ash219	56	79.00	79.96	199.63	48.72	0.79	1630	859	2.00	0
dwt72	48	68.00	68.55	55.21	20.44	0.58	1111	739	49.00	4
can62	41	56.38	56.98	46.51	8.31	0.34	798	422	56.00	155
dwt66	44	62.00	62.91	47.13	4.81	0.27	613	352	62.00	325
bcpwr02	32	44.00	44.51	6.59	1.02	0.12	275	151	44.00	151
dwt59	39	51.77	51.97	36.17	3.29	0.25	424	239	2.00	0
bcpwr01	26	36.00	36.98	9.47	2.42	0.14	481	314	36.00	58

Tabela 4: Instâncias MM-I no Grasp com $\alpha = 0.5$ e iterações= 3.

Instancia	b	PMR	UB	tPMR	tPric	Cols	iter	LB	itLB	(b,c,g)
ash219	56	79.50	80.72	1800.09	936.76	13108	2132	79.00	1438	(0.00,0.09,0.91)
dwt72	48	68.00	68.90	454.50	261.88	6402	1382	68.00	1123	(0.00,0.11,0.89)
can62	41	57.00	57.90	252.16	193.11	4294	753	56.00	157	(0.00,0.04,0.96)
dwt66	44	62.00	62.56	407.93	316.68	4906	994	62.00	747	(0.00,0.18,0.82)
bcpwr02	32	44.23	44.92	125.61	100.80	3188	627	37.00	5	(0.00,0.00,1.00)
dwt59	39	52.61	52.82	299.52	262.56	3408	541	51.00	63	(0.00,0.00,1.00)
bcpwr01	26	36.00	36.95	17.16	15.43	928	210	36.00	40	(0.00,0.03,0.97)
ash85	56	73.26	78.35	1799.52	1734.22	4218	334	62.00	5	(0.00,0.01,0.99)

Tabela 5: Geração de colunas clássica com o uso de heurística primal nas instâncias MM-I.

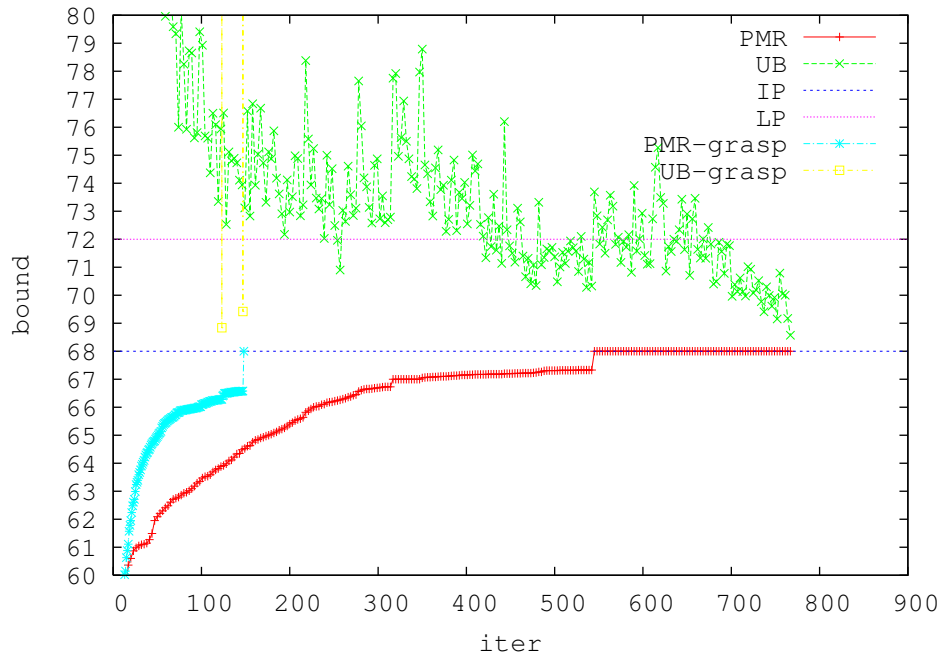


Figura 3: Limitantes superiores para mat.dwt72 com o uso e sem o uso de GRASP.

4.7 Mestre explícito com pricing parcial

As Tabelas 6 e 7 resumem os resultados computacionais da geração de colunas, usando-se a decomposição do mestre explícito com pricing exato e parcial, respectivamente. A coluna *Grasp* na Tabela 7 indica o número de iterações da geração de colunas em que a heurística não falhou na obtenção de colunas de custo reduzido positivo.

Instancia	b	PMR	UB	tPMR	tPric	Cols	iter	LB	itLB
ash219	56	78.40	81.84	1803.20	787.01	2763	944	78.00	200
dwt72	48	68.00	68.78	1166.67	474.09	2885	1068	68.00	126
can62	41	56.39	56.71	281.44	225.14	968	318	56.00	305
dwt66	44	62.00	62.86	972.61	526.48	2266	1005	62.00	987
bcpwr02	32	44.00	44.80	95.10	77.61	665	269	44.00	49
dwt59	39	51.75	51.99	399.30	346.51	932	320	51.00	193
bcpwr01	26	36.00	36.64	37.76	32.04	459	216	35.00	42

Tabela 6: Geração de colunas com mestre explícito e pricing exato em instâncias MM-I.

Elas mostram, novamente, um ganho considerável no tempo de processamento ao utilizar-se uma heurística GRASP no pricing.

Instancia	b	PMR	UB	tPMR	tPric	tGrasp	Grasp	Cols	iter	LB	itLB
ash219	56	79.00	79.89	312.88	142.31	1.34	519	1308	595	78.00	512
dwt72	48	68.00	68.93	244.71	32.37	1.50	721	1508	798	68.00	201
can62	41	56.39	56.91	58.24	43.46	0.27	195	538	234	56.00	166
dwt66	44	62.00	62.89	98.14	63.88	0.55	286	721	387	61.00	148
bcpwr02	32	44.00	44.62	7.72	4.65	0.09	120	270	129	42.00	86
dwt59	39	51.77	51.84	37.45	26.48	0.31	190	447	205	51.00	204
bcpwr01	26	36.00	36.61	6.72	3.62	0.09	143	317	162	36.00	48

Tabela 7: Geração de colunas com mestre explícito e pricing parcial em instâncias MM-I.

4.8 Efeito *tailing off*

A Figura 3 ilustra o efeito *tailing off* que ocorre no processo de geração de colunas. Nela, pode-se observar também a grande oscilação no valor das variáveis duais. As próximas subseções discutem estratégias para acelerar a convergência dos algoritmos de geração de colunas na tentativa de diminuir os efeitos de *tailing off*.

4.9 Métodos de estabilização

Quando o problema mestre é degenerado, o dual dele tem um número infinito de soluções ótimas. Os solvers de LP retornam um ponto extremo do poliedro dual, levando as variáveis duais a ter valores pequenos ou grandes demais. Este fato pode ser observado na grande oscilação nos valores dos limitantes duais obtidos ao longo do processo de geração de colunas. Muitas aproximações melhores para os custos marginais reais seriam obtidos se as variáveis duais tomassem valores no centro (interior) do poliedro dual ótimo. Para prevenir as variáveis duais de tomar valores extremos, a estratégia é tentar limitar a distância percorrida pelas variáveis duais no espaço dual de uma iteração a outra. Existem duas técnicas para isso: box e limitantes com penalização. Du Merle et al [dMVDH99] propuseram um método para estabilização dos algoritmos de geração de colunas que combina estas duas técnicas. Para expor o método de Du Merle et al, considere a seguinte formulação do problema mestre e seu dual:

$$\begin{aligned}
 \text{(BoxP)} \quad & \min \sum_{j \in N} c_j x_j \\
 \text{sujeito a} \quad & \sum_{j \in N} a_{ij} x_j \geq 1, \forall i \in M \\
 & x_j \geq 0, \forall j \in N.
 \end{aligned}$$

$$\begin{aligned}
 \text{(BoxD)} \quad & \max \sum_{i \in M} \pi_i \\
 \text{sujeito a} \quad & \sum_{i \in M} a_{ij} \pi_i \leq c_j, \forall j \in N \\
 & \pi_i \geq 0, \forall i \in M.
 \end{aligned}$$

A idéia é impor limites (d^- , d^+) sobre as variáveis duais π_i associadas com as restrições de (BoxP), mas permitindo que elas tomem valores fora destes limites (usando variáveis “de folga” w^- e w^+). Quando o valor das variáveis w^- ou w^+ são diferentes de zero (isto é, as variáveis duais têm valores fora dos limites), uma penalização é aplicada na função objetivo (ε^- e ε^+). O problema mestre modificado e seu dual estabilizado é:

Este método de estabilização é conhecido como *Box Stabilization* pelo fato de que o valor de d_- e d_+ são definidos por um centro de box, Box_c e pela largura do box, ξ da seguinte forma:

$$d_- = Box_c - \xi$$

e

$$d_+ = \text{Box}_c + \xi.$$

Se uma boa estimativa do valor dos duais ótimas é conhecida, ela é usada para inicializar o valor de Box_c . Caso contrário, o valor dos duais inicial em P é usado.

O algoritmo de geração de colunas estabilizado consiste em, cada iteração, resolver o problema mestre restrito $\overline{\text{Box}P}$ no lugar de $(\text{Box}P)$, incluir em $\overline{\text{Box}P}$ a coluna de custo reduzido negativo gerada pelo *pricing* e atualizar os parâmetros d_-, d_+, ε_- e ε_+ . Note que $(\overline{\text{Box}P})$ é uma relaxação de $(\text{Box}P)$ e que uma solução (\bar{x}, y_-, y_+) de $(\overline{\text{Box}P})$ é, também, uma solução de $(\text{Box}P)$ sempre que $y_- = y_+ = 0$. Portanto, o algoritmo de geração de colunas estabilizado termina somente quando o *pricing* retorna uma coluna de custo reduzido positivo e $y_- = y_+ = 0$ (ou seja, ou (i) $\varepsilon_- = \varepsilon_+ = 0$ ou (ii) $d_- < \pi < d_+$). Para garantir a convergência do algoritmo, deve-se escolher estratégias adequadas para a atualização dos parâmetros d_-, d_+, ε_- e ε_+ .

Du Merle *et al* sugerem as seguintes estratégias de atualização:

- Diminua o valor de ε_- e ε_+ sempre que o *pricing* devolver uma coluna de custo reduzido positivo ou o valor dos duais for a melhor estimativa para os duais ótimos. Nos demais casos, aumenta-se o valor de ε_- e ε_+ . Uma medida da qualidade do valor dos duais pode ser obtida através do cálculo dos limitantes duais de Lasdon.
- Atualize o valor de d_- e d_+ para os duais correntes sempre que o valor dos duais for a melhor estimativa para os duais ótimos (ou seja, faça $\text{Box}_c = \pi$, mantendo-se $\xi > 0$). Quando o *pricing* devolver uma coluna de custo reduzido positivo, aumente o valor de ξ , mantendo o centro do box.

Ambas estratégias de atualização devem juntas garantir a convergência do método. Desde que $\text{Box}P$ seja limitado, em alguma iteração do processo de geração de colunas, o *pricing* retornará uma coluna de custo reduzido positivo e o valor de ε_- e ε_+ será decrementado. Portanto, em um número finito de iterações, a condição (i) ocorrerá e o algoritmo termina. Seja π^k o valor dos duais na iteração k , na qual o *pricing* retornou uma coluna de custo reduzido positivo. Note que $\text{box}D$ é uma relaxação de $\overline{\text{box}D}$. Portanto, π^k é viável para $\text{box}D$ e $v(\overline{\text{box}D}^k) \leq v(\text{box}D)$. Se $v(\overline{\text{box}D}^k) = v(\text{box}D)$ então $\varepsilon_- = \varepsilon_+ = 0$ ou $w_- = w_+ = 0$. No primeiro caso, a condição (i) foi atingida na iteração k e o algoritmo termina. No último caso, após atualizar ξ (aumentar $\xi > 0$), a condição (ii) é atingida e o algoritmo termina. Caso $v(\overline{\text{box}D}^k) < v(\text{box}D)$, em alguma iteração posterior t , $v(\overline{\text{box}D}^k) < v(\overline{\text{box}D}^t)$, uma vez que o valor de ξ foi aumentado na iteração k . E, portanto, após um número finito de iterações l , $v(\overline{\text{box}D}^{k+l}) = v(\text{box}D)$.

A dificuldade na utilização desta estratégia está na determinação das estratégias de atualização e no valor dos parâmetros, que podem apresentar desempenhos diferentes quando aplicados a problemas distintos.

Usando a estratégia de Du Merle *et al* no VSP, o problema mestre restrito estabilizado do VSP consiste em:

$$(PMRs) \quad \max \sum_{j \in Q} w_j x_j + \sum_{i \in M} (d_i^- y_i^- - d_i^+ y_i^+) \quad (4.10)$$

$$\text{sujeito a} \quad \sum_{j \in Q} q_{ij} x_j + y_i^- - y_i^+ \leq 2, \forall i \in M \quad (4.11)$$

$$y_i^- \leq \varepsilon_i^-, \forall i \in M \quad (4.12)$$

$$y_i^+ \leq \varepsilon_i^+, \forall i \in M \quad (4.13)$$

$$x_j \in \{0, 1\}, \forall j \in Q.$$

A utilização deste método de estabilização no VSP possibilitou uma pequena melhora no tempo de processamento, por outro lado reduziu o problema de degenerescência. Box_c foi inicializado com o valor dos duais iniciais do PMR e com largura $\xi = 0.0001$. Inicialmente, considerou-se $\varepsilon_- = 0.01$ e $\varepsilon_+ = 2$. Ambos os valores foram aumentados em 10% e diminuídos em 50%, de acordo com a estratégia de atualização de Du Merle *et al* discutido nesta subseção. As Tabelas 8 e 9 resumem os resultados computacionais da geração de colunas clássica, respectivamente, sem e com a utilização do método de estabilização de Du Merle *et al*.

4.9.1 Interior point stabilization

Rousseau *et al* propuseram um método diferente, chamado *Interior Point Stabilization*, para estabilização do algoritmo de geração de colunas [RGF02]. O método consiste em gerar vários pontos extremos no poliedro dual (ao invés de um só) e tomar a combinação convexa deles para obter um ponto interior do poliedro. A idéia para obter a base dual ótima consiste em identificar os conjuntos S e R^* , definidos a seguir, e usar complementariedade das folgas. Seja x^* a solução ótima de P e seja $S = \{i \in M \mid \sum_{j \in N} a_{ij} x_j^* > 1\}$. Por complementariedade das folgas tem-se que $\pi_i = 0$, para todo $i \in S$, na base dual ótima. Por outro lado, seja $R^* = \{j \in N \mid x_j^* > 0\}$, ou seja, relativo às variáveis básicas não-nulas. Por complementariedade das folgas, segue que $\sum_{i \in M} a_{ij} \pi_i = c_j$, para todo $j \in R^*$, na base dual ótima.

Logo, do seguinte programa linear $D(u)$ pode-se obter o valor de variáveis duais na base dual ótima a partir de diferentes valores de u :

$$(D(u)) \quad \max \sum_{i \in M} u_i \pi_i$$

$$\text{sujeito a} \quad \sum_{i \in M} a_{ij} \pi_i = c_j, \forall j \in R^*$$

$$\sum_{i \in M} a_{ij} \pi_i \leq c_j, \forall j \in N \setminus R^*$$

$$\pi_i = 0, \forall i \in S$$

$$\pi_i \geq 0, \forall i \in M \setminus S.$$

Instancia	b	PMR	UB	tPMR	tPric	Cols	iter	LB	itLB	Deg
ash219	56	79.10	81.78	1800.12	969.41	6831	2136	57.00	1	1341(1341)
dwt72	48	68.00	68.82	305.37	215.16	2736	983	68.00	787	526(528)
can62	41	57.00	57.98	426.30	293.00	3281	1337	42.00	2	1046(1046)
dwt66	44	62.00	62.97	563.43	404.49	3270	1364	45.00	1	1261(1261)
bcpwr02	32	44.24	44.93	57.74	52.91	799	281	33.00	5	88(88)
dwt59	39	52.61	52.72	227.54	208.29	1351	422	40.00	6	172(172)
bcpwr01	26	36.00	36.84	13.29	12.20	374	155	36.00	73	94(94)

Tabela 8: Dantzig-Wolfe sem estabilização.

Instancia	b	PMR	UB	tPMR	tPric	Cols	iter	LB	itLB	Deg
ash219	56	79.33	81.70	1800.37	995.00	7221	2230	57.00	1	0(0)
dwt72	48	68.00	68.78	204.42	160.34	1885	683	2.00	0	0(0)
can62	41	57.00	57.75	369.88	277.15	2974	1094	42.00	1	0(0)
dwt66	44	62.00	63.00	407.78	341.33	2480	789	45.00	1	0(0)
bcpwr02	32	44.22	44.96	64.99	58.06	944	367	33.00	1	0(0)
dwt59	39	52.61	52.90	183.20	170.20	1113	350	2.00	0	0(0)
bcpwr01	26	36.00	36.77	12.22	11.40	359	141	27.00	1	0(0)

Tabela 9: Dantzig-Wolfe com box stabilization.

Após a obtenção de algumas variáveis duais ótimas, toma-se uma combinação convexa delas (por exemplo, uma média delas) para se obter o valor de uma variável dual no interior da base dual ótima.

Um programa linear alternativo para se obter os duais ótimos consiste no dual de $D(u)$, que é bastante similar à formulação original de P :

$$\begin{aligned}
 (\text{P}(u)) \quad & \min \sum_{j \in N} c_j x_j \\
 \text{sujeito a} \quad & \sum_{j \in N} a_{ij} x_j \geq u_i, \forall i \in M \setminus S \\
 & \sum_{j \in N} a_{ij} x_j \geq -\infty, \forall i \in S
 \end{aligned} \tag{4.14}$$

$$\begin{aligned}
 & x_j \geq 0, \forall j \in N \setminus R^* \\
 & x_j \text{ qualquer}, \forall j \in R^*.
 \end{aligned} \tag{4.15}$$

A restrição (4.14) faz com que $\pi_i = 0$, para todo $i \in S$ e a restrição (4.15) faz com que a restrição correspondente no dual seja satisfeito na igualdade.

Inicialmente, tome $u_i = 1$, para todo $i \in M$, $S = \emptyset$ e $R^* = \emptyset$ para obter os conjuntos S e R^* . Depois, seta-se diferentes valores para u_i , $i \in M \setminus S$ e resolve-se $P(u)$ repetidas vezes e calcula-se uma combinação convexa dos duais ótimos obtidos com a solução de cada $P(u)$.

A seguinte formulação parametrizada, com o valor da perturbação u , foi utilizada para obter diferentes duais no VSP:

$$\begin{aligned}
 (\text{VSP}(u)) \quad & \max \sum_{j \in N} c_j x_j \\
 \text{sujeito a} \quad & \sum_{j \in N} a_{ij} x_j \leq 2u_i, \forall i \in M \setminus S \\
 & \sum_{j \in N} a_{ij} x_j \leq +\infty, \forall i \in S
 \end{aligned} \tag{4.16}$$

$$\begin{aligned}
 & x_j \geq 0, \forall j \in N \setminus R^* \\
 & x_j \text{ qualquer}, \forall j \in R^*.
 \end{aligned} \tag{4.17}$$

O gráfico da Figura 4 mostra a oscilação dos duais com e sem a utilização do método interior point stabilization.

4.9.2 Outros métodos para estabilização

Um método que reduz o problema da oscilação dos duais foi proposto por Ben Amor, Desrosiers e Valério de Carvalho [ADdC], que consiste na utilização de desigualdades válidas, chamadas **desigualdades duais-ótimas**, para as soluções ótimas do dual do problema mestre restrito, que em P representa a inclusão de colunas adicionais. A ideia deste método

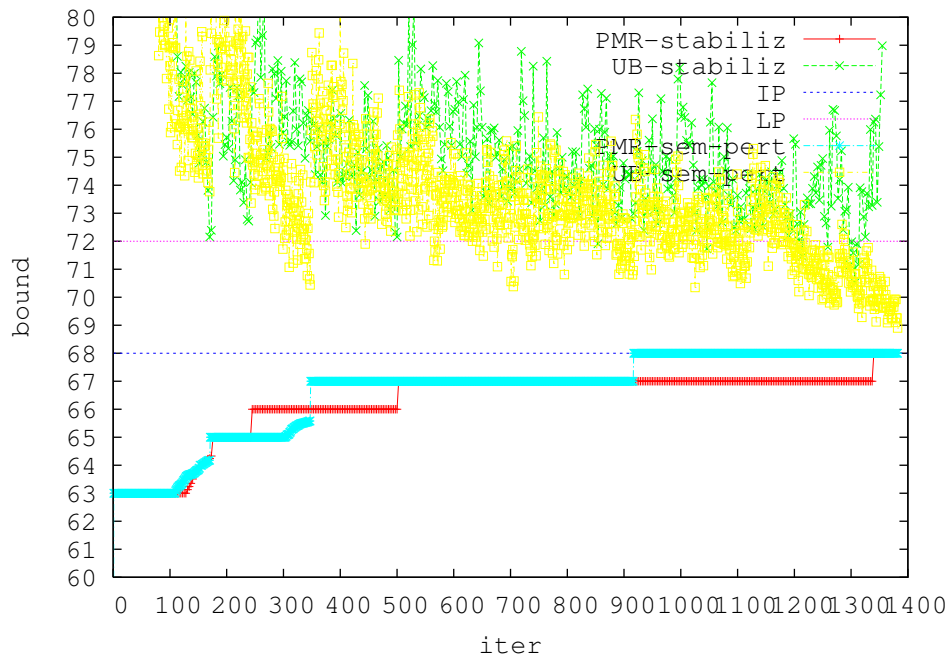


Figura 4: Geração de colunas clássica, com heurística primal, com e sem a utilização do método interior point stabilization.

é que a inclusão destas colunas, restringe os valores dos multiplicadores duais e assim, reduz a oscilação deles no espaço dual, acelerando a convergência em direção aos multiplicadores duais ótimos. Os resultados computacionais da aplicação deste método no problema de corte e empacotamento mostrou a efetividade do método, que reduziu significativamente o tempo e o número de iterações para a convergência do algoritmo de geração de colunas. A utilização deste método é dependente da aplicação uma vez que desigualdades duais-ótimas devem ser encontradas.

Gondzio e Sarkissian [GS96] mostram como usar método de ponto interior na geração de colunas. A idéia é utilizar o método primal-dual inviável para resolver a relaxação linear do problema mestre restrito, abrindo-se mão da otimalidade deste. A vantagem deste método é a rapidez na obtenção dos multiplicadores duais e na qualidade deles para a geração de colunas.

4.10 Degenerescência

A formulação do problema mestre restrito produz muitas soluções degeneradas, como mostra a Tabela 10. Nela, a coluna **Deg** representa o número de soluções degeneradas geradas pelo algoritmo de geração de colunas clássico (o valor entre parênteses representa o número de iterações em que o problema mestre restrito não teve progresso). A coluna **it** representa o número de iterações realizadas no processo de geração de colunas. Soluções degeneradas causam iterações inúteis no algoritmo de geração de colunas, ou seja, que não produzem

melhorias na solução do problema e, assim, pioram a convergência do algoritmo. Uma das técnicas para diminuir a degenerescência em problemas de programação linear consiste em perturbar o lado direito (*rhs*) das restrições do problema. Uma escolha adequada para o valor da perturbação é aquela que não interfere na solução ótima do problema. Para o VSP, uma escolha adequada é adicionar uma perturbação $\epsilon_i = 1/(nrows + i)$ para cada linha i da matriz de restrições (*nrows* é o número de restrições). Pelas Tabelas 10 e 11 pode ser notado que o número de soluções degeneradas diminui consideravelmente em relação à versão não perturbada. Nos testes realizados, pode-se notar um grande número de soluções degeneradas em instâncias do VSP.

As Tabelas 12 e 13 resumem os resultados dos testes realizados sobre as instâncias MM-I na geração de colunas clássica com pricing parcial para avaliar o impacto da perturbação do *rhs*. Os testes mostram a efetividade na diminuição do tempo de convergência tanto na versão com e sem pricing parcial.

5 Conclusões

O método da geração de colunas foi utilizado para resolver o VSP. As decomposições de Dantzig-Wolfe e de mestre explícito foram utilizadas para obter uma formulação extensiva para o problema. Os limitantes obtidos com a resolução do problema mestre foram bem mais apertados que aqueles dados pela relaxação linear da formulação compacta. Isso decorre da decomposição do problema que foi escolhida. O problema de pricing para o VSP gera colunas que representam um *shore* e sua vizinhança no separador. Ao contrário dos limitantes gerados, o tempo gasto pelo algoritmo de geração de colunas não foi satisfatório, considerando-se os resultados recentes conhecidos para as instâncias analisadas.

Embora o método da geração de colunas, utilizando-se a decomposição escolhida neste estudo, não se mostrou adequado ao VSP, testes foram realizados sobre instâncias fáceis com o intuito de avaliar as diferentes técnicas para acelerar o processo de geração de colunas. A utilização de um pool de colunas mostrou-se bastante efetivo, reduzindo, na média, em 50% o tempo de processamento. Uma redução de cerca de 20%, em média, no tempo de processamento foi obtido com a utilização de limitantes superiores inteiros. A heurística GRASP para o pricing conseguiu ter um ganho de até 5 vezes no desempenho em relação ao pricing exato. Um fato interessante é que a heurística falhou um número muito pequeno de vezes e os limitantes duais obtidos com o pricing exato foram apertados o suficiente para encerrar cedo o processo de geração de colunas. O uso de uma heurística primal melhorou muito pouco o desempenho do algoritmo de geração de colunas, no entanto, os resultados mostram que as colunas geradas pela heurística primal foram utilizadas na maioria das soluções ótimas do problema mestre.

Um dos problemas que ocorrem em algoritmos de geração de colunas refere-se à grande oscilação dos duais durante a geração de colunas, que pode dificultar a convergência do método. Para resolver este problema, dois métodos de estabilização foram analisados. Os resultados da utilização destes métodos não foram satisfatórios. A dificuldade na utilização destes métodos está na escolha adequada dos parâmetros e das estratégias de atualização dos mesmos. O método *box stabilization* reduziu relativamente pouco o tempo de proces-

Instancia	b	PMR	UB	tPMR	tPric	Cols	iter	LB	itLB	Deg
ash219	56	79.10	81.78	1801.20	974.73	6714	2103	57.00	1	1308(1308)
dwt72	48	68.00	68.82	320.52	217.85	2736	983	68.00	787	526(528)
can62	41	57.00	57.98	451.08	299.01	3281	1337	42.00	2	1046(1046)
dwt66	44	62.00	62.97	591.12	427.32	3270	1364	45.00	1	1261(1261)
bcpwr02	32	44.24	44.93	59.09	53.96	799	281	33.00	5	88(88)
dwt59	39	52.61	52.72	242.60	222.24	1351	422	40.00	6	172(172)
bcpwr01	26	36.00	36.84	13.70	12.73	374	155	36.00	73	94(94)
ash85	56	73.50	78.45	1806.68	1747.30	2111	349	57.00	1	194(195)

Tabela 10: Geração de colunas clássica sem perturbação do *rhs* nas instâncias MM-I.

Instancia	b	PMR	UB	tPMR	tPric	Cols	iter	LB	itLB	Deg
ash219	56	79.81	79.83	811.74	587.67	3658	1037	-	-1	1(1)
dwt72	48	68.31	68.84	169.58	130.45	1563	553	2.00	0	0(0)
can62	41	57.30	57.89	116.59	98.97	1131	445	2.00	0	0(0)
dwt66	44	62.10	62.96	148.20	133.05	1111	348	-	-1	0(0)
bcpwr02	32	44.53	45.00	49.42	44.52	737	290	-	-1	0(0)
dwt59	39	52.91	52.72	155.99	146.00	983	287	-	-1	0(0)
bcpwr01	26	36.27	36.50	7.25	6.71	218	106	2.00	0	0(0)
ash85	56	73.77	77.19	1806.62	1743.95	2076	325	-	-1	0(0)

Tabela 11: Geração de colunas clássica com perturbação do *rhs* nas instâncias MM-I.

Instancia	b	PMR	UB	tPMR	tPric	tGrasp	totGrasp	Cols	iter	LB	itLB	Deg
ash219	56	79.50	79.85	740.90	363.31	2.81	1719	4792	2103	*	-1	1658(1658)
dwt72	48	68.00	68.53	10.23	3.60	0.22	263	641	270	49.00	2	42(42)
can62	41	57.00	57.68	30.71	19.88	0.37	425	930	478	2.00	1	251(251)
dwt66	44	62.00	62.88	10.61	7.45	0.30	219	470	243	44.00	2	129(129)
bcpwr02	32	44.17	44.83	7.23	4.77	0.16	210	475	230	33.00	2	73(73)
dwt59	39	52.61	52.83	24.94	22.10	0.23	192	452	214	39.00	2	25(25)
bcpwr01	26	36.00	36.89	1.53	0.93	0.09	122	252	131	2.00	1	54(54)
ash85	56	74.11	76.04	1799.13	1681.25	1.49	683	2741	935	*	-1	571(573)

Tabela 12: Geração de colunas clássica com Grasp sem perturbação do *rhs*.

Instancia	b	PMR	UB	tPMR	tPric	tGrasp	totGrasp	Cols	iter	LB	itLB	Deg
ash219	56	79.81	79.89	131.00	76.74	0.96	618	1609	695	*	-1	29(35)
dwt72	48	68.32	68.64	19.79	8.75	0.31	334	816	356	*	-1	12(16)
can62	41	57.30	57.94	24.03	16.08	0.38	310	742	362	*	-1	0(13)
dwt66	44	62.30	62.90	23.73	21.23	0.26	178	437	207	*	-1	2(10)
bcpwr02	32	44.54	44.99	16.12	12.41	0.21	259	598	302	*	-1	0(4)
dwt59	39	52.91	52.78	15.12	12.50	0.16	180	416	195	*	-1	0(0)
bcpwr01	26	36.28	36.64	1.94	1.12	0.04	135	294	144	*	-1	0(0)
ash85	56	74.44	75.14	1799.27	1713.99	1.22	573	2116	801	*	-1	44(73)

Tabela 13: Geração de colunas clássica com Grasp com perturbação do *rhs*.

samento, mas conseguiu resolver o problema de degenerescência que ocorre nos problemas mestres. Outra técnica utilizada para reduzir a degenerescência foi a perturbação do *rhs* das restrições. Esta técnica mostrou efetiva na remoção da degenerescência.

Diferentes combinações de estratégias foram implementadas para avaliar a melhor implementação para o algoritmo de geração de colunas no VSP. A Tabela 14 resume a média de tempo (**mTime**), tempos mínimos (**minTime**) e máximos (**maxTime**), média de *gap* (**mGap**), *gap* mínimo (**minGap**) e máximo (**maxGap**) nas diferentes combinações de implementação. Nesta tabela, a implementação de geração de colunas clássica é denotada por DW e a versão com pricing parcial por Grasp. O sufixo **-h** é usado para indicar a utilização de heurística primal, enquanto **-p** para indicar a versão com perturbação no *rhs*. O sufixo **-hsp** indica que tanto a heurística, quanto o método de estabilização e perturbação no *rhs* foram utilizados. Pode-se notar que a versão que utiliza heurística primal, pricing parcial, método de estabilização e perturbação do *rhs* consegue reduzir em até 70% o tempo de processamento em relação ao algoritmo de geração de colunas clássico.

codigo	mTime	minTime	maxTime	mGap	minGap	maxGap
DW	649.90	13.29	1805.44	0.01	0.00	0.06
DW-h	644.56	17.16	1800.09	0.01	0.00	0.06
DW-p	369.71	14.08	1799.73	0.00	0.00	0.03
DW-hp	342.38	6.68	1799.97	0.00	0.00	0.03
DW-hsp	338.43	7.64	1799.52	0.01	0.00	0.04
Grasp	328.16	1.53	1799.13	0.00	0.00	0.02
Grasp-h	302.75	0.85	1799.56	0.00	0.00	0.01
Grasp-p	253.88	1.94	1799.27	0.00	0.00	0.01
Grasp-hp	255.79	0.54	1799.80	0.00	0.00	0.01
Grasp-hsp	184.37	0.50	1239.59	0.00	0.00	0.00

Tabela 14: Resumo comparativo das implementações nas instâncias MM-I.

Embora o método da geração de colunas não tenha apresentado resultados satisfatórios para o VSP, as técnicas discutidas neste relatório mostram como elas podem ser combinadas para melhorar o desempenho de algoritmos de geração de colunas.

Referências

- [ADdC] H.B. Amor, J. Desrosiers, e J.M. V. de Carvalho. Dual-optimal inequalities for stabilizes column generation. To appear in *Operations Research*, 2005.
- [BdS05a] E. Balas e C. C. de Souza. The vertex separator problem: a polyhedral investigation. *Mathematical Programming*, 103(3):583–608, 2005.
- [BdS05b] E. Balas e C. C. de Souza. The vertex separator problem: algorithms and computations. *Mathematical Programming*, 103(3):609–631, 2005.

- [BJN⁺98] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, e P.H. Vance. Branch-and-price: Column generation for huge integer programs. *Operations Research*, 46:316–329, 1998.
- [dAU03] M. Poggi de Aragão e E. Uchoa. Integer program reformulation for robust branch-and-cut-and-price, August 2003. Working paper. Presented at ISMP'03. Copenhagen.
- [dMVDH99] O. du Merle, D. Villeneuve, J. Desrosiers, e P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194:229–237, 1999.
- [DSD84] Desrosiers, F. Soumis, e M. Desrochers. Routing with time windows by column generation. *networks*, 14:545–565, 1984.
- [DW60] G. Dantzig e P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.
- [Far90] A. A. Farley. A note on bounding a class of linear programming problems, including cutting stock problems. *Operations Research*, 38:992–993, 1990.
- [FLL⁺06] R. Fukasawa, H. Longo, J. Lysgaard, M. P. de Aragão, M. Reis, E. Uchoa, e R.F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511, July 2006.
- [GS96] J. Gondzio e R. Sarkissian. Column generation with a primal-dual method. Relatório técnico, University of Geneva, 102 Bd Carl Vogt, CH-1211 Geneva 4, Switzerland, 1996.
- [Las70] L.S. Lasdon. *Optimization Theory for Large Systems*. MacMillan, 1970.
- [LD] M.E. Lübbecke e J. Desrosiers. Selected topics in column generation. Working paper, Cahier du Gerad.
- [RG] T.K. Ralphs e M. Galati. Decomposition and dynamic cut generation in integer programming. To appear in *Mathematical Programming (2005)* (Working paper version). <http://www.lehigh.edu/~tkr2/research/pubs.html>.
- [RGF02] L. M. Rousseau, M. Gendreau, e D. Feillet. Interior point stabilization for column generation, 2002. Working paper, Cahier du Gerad.
- [Sav97] M.W.P. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45:831–841, 1997.
- [VW96] F. Vanderbeck e L. A. Wolsey. An exact algorithm for ip column generation. *Operations Research Letters*, 19:151–159, 1996.