Um novo limite inferior baseado num modelo de Programação por Restrições para o Problema de Minimização de Padrões

Cláudio Alves*, Rita Macedo, José Valério de Carvalho

Centro de Investigação Algoritmi da Universidade do Minho,
Escola de Engenharia, Universidade do Minho, 4710-057 Braga, Portugal
{claudio,rita,vc}@dps.uminho.pt

15 de Setembro de 2008

Resumo

O Problema de Minimização de Padrões é um problema de optimização combinatória da família dos problemas de corte e empacotamento que consiste em determinar o plano de corte com o menor número de padrões diferentes. Esse problema tem sido essencialmente abordado através de procedimentos heurísticos, sendo muito poucas as contribuições descritas na literatura relativas a abordagens de resolução exacta.

Neste artigo, descrevemos um novo limite inferior que é calculado em tempos computacionais que superam outras abordagens ao nível do estado-da-arte. Consideramos o problema a uma dimensão, e em particular o caso em que o número de rolos a usar é conhecido. Esse valor é obtido resolvendo o problema de corte standard correspondente. Para calcular o limite, usamos um modelo de Programação por Restrições que incorpora de forma eficiente as restrições mais complexas do Problema de Minimização de Padrões. Introduzimos também um conjunto de novas restrições válidas que contribuem para o reforço do nosso modelo. Os resultados que obtivemos em instâncias da literatura mostram que o limite inferior é obtido em tempos computacionais muito reduzidos. Em alguns casos, é mesmo mais forte que o limite contínuo obtido através do melhor modelo de geração de colunas descrito até agora.

1 Introdução

Neste artigo, abordamos o Problema de Minimização de Padrões (PMP) unidimensional, que segundo a tipologia proposta por Wäscher *et al.* [20] para problemas de corte e empacotamento, pode ser caracterizado como um problema 1D-SSSCSP (*one-dimensional single stock size cutting stock problem*). O objectivo do problema é reduzir o número de padrões de corte diferentes usados num determinado plano. O problema é definido através de um conjunto de m itens de tamanhos w_i e procuras b_i, $i \in \{1, \ldots, m\}$. Os itens devem ser cortados a partir de rolos de comprimento W. Sempre que um novo padrão de corte

*Corresponding author. Tel.: +351 253 604765; fax: +351 253 604741.
E-mail address: claudio@dps.uminho.pt (C. Alves).

é iniciado, é necessário acertar as facas da máquina de corte numa operação dita de *setup*. Essa operação leva tempo, e pode gerar desperdícios por ser necessário testar o posicionamento das facas realizando experiências com a matéria-prima disponível. Em contexto industrial, a redução do número de padrões diferentes é relevante no sentido em que contribui para a minimização quer dos tempos de operação quer dos custos relativos a materiais.

O PMP é um problema NP-difícil [15]. A maioria dos métodos de resolução descritos na literatura baseia-se em heurísticas, sendo poucas as publicações que apresentam resultados de métodos de resolução exacta ou limites inferiores. Todas as abordagens apresentadas, tanto heurísticas como exactas, assentam numa de duas possíveis abordagens. Numa delas, o problema de minimização de padrões é resolvido num único estágio em conjunto com o problema de corte. Nesses casos, procura-se o melhor equilíbrio entre o desperdício e o número de padrões diferentes. Outra abordagem consiste em assumir que não é possível utilizar mais do que um dado número de rolos (geralmente o número de rolos correspondente à solução óptima do problema de corte standard), e a partir daí encontrar a solução com o número mínimo de padrões diferentes. Neste artigo, consideramos esta última abordagem.

Em [11], Haessler apresenta uma heurística sequencial onde são favorecidos os padrões com desperdícios pequenos e elevados níveis de utilização. A heurística procura soluções onde os níveis de desperdício e o número de padrões distintos sejam equilibrados.

Em [17], Teghem *et al.* analisam um problema real da indústria editorial no qual se considera a produção de capas de livros. Os autores formulam o problema como um PMP, e resolvem-no usando um método de arrefecimento simulado. Chen *et al.* [4] propõem também um algoritmo de arrefecimento simulado para a resolução do problema de corte standard cuja função objectivo considera custos de materiais e de *setups*.

Em [6], Diegel *et al.* apresentam um método heurístico que combina dois padrões diferentes num único padrão, mantendo o número de rolos usados. Este conceito foi generalizado por Foerster e Wäscher [9] para combinações de p padrões diferentes em q padrões diferentes, considerando apenas combinações de p para $(p-1)$ e limitando p a 4. Foerster e Wäscher propõem uma classe de métodos designada por KOMBI. Cada uma dessas classes define um modo e uma sequência para combinar os diferentes padrões. Os autores apresentam testes computacionais para as classes KOMBI23 (combinações de 2 para 1 e 3 para 2) e KOMBI234 (combinações de 2 para 1, 3 para 2 e 4 para 3).

Em [18], Umetani *et al.* descrevem uma meta-heurística para o PMP. Os autores tentam encontrar o menor número possível de padrões distintos fixando iterativamente esse valor e tentando encontrar uma solução para o problema de corte standard correspondente. Para esse efeito, os autores recorrem a métodos de pesquisa local, algoritmos de geração de padrões adaptativos e a uma heurística baseada no método não-linear de Gauss-Seidel.

Yanasse e Limeira [21] propõem um método híbrido para o PMP que é independente da sua dimensão. O método pode ser decomposto em três fases. Numa primeira fase, recorre-se à técnica *RPET* (*Repeated Pattern Exhaustion Technique*) para encontrar um conjunto conveniente de padrões de corte. Numa segunda fase, resolve-se um problema de corte residual a partir do conjunto dos itens que não foram considerados na primeira fase. Finalmente, na terceira fase, são aplicadas ao conjunto de padrões gerados nas duas primeiras fases, técnicas de redução descritas na literatura.

A primeira abordagem exacta para o PMP foi proposta por Vanderbeck [19]. Nesse artigo, o autor descreve um algoritmo de partição e geração de colunas com cortes baseado num modelo de geração de

colunas em que as variáveis estão associadas a padrões com uma determinada multiplicidade. A geração dinâmica de colunas implica a resolução de subproblemas de programação inteira quadrática. O autor contorna essas não-linearidades resolvendo um conjunto de problemas de mochila com limites, um para cada valor possível de multiplicidade. Para reforçar o modelo, o autor recorre a uma família de planos de corte baseada em funções superaditivas. Num estudo posterior, Clautiaux *et al.* [5] provaram que algumas das funções de Vanderbeck não são maximais, podendo ser dominadas por outras funções superaditivas.

A abordagem proposta por Vanderbeck foi melhorada recentemente por Alves e Carvalho ([2]). Nesse artigo, os autores propõem um novo algoritmo de partição e geração de colunas com cortes para o PMP. A regra de partição do algoritmo baseia-se nas variáveis de um modelo de fluxos em arcos, equivalente ao modelo original, que permite eliminar a simetria na árvore de pesquisa. Os autores descrevem também várias formas de melhorar o modelo de geração de colunas proposto por Vanderbeck [19]. Os autores apresentam diversos resultados ligados ao uso de funções duais válidas para a geração de cortes. As funções que usam foram descritas por Fekete e Schepers em [8]. Os autores fazem também a prova de que os cortes gerados por estas funções são equivalentes ou dominam a função superaditiva usada por Vanderbeck.

Em [3], Belov descreve um algoritmo de partição e geração de colunas para o PMP baseado numa extensão do modelo de Gilmore e Gomory [10] para o problema de corte, onde se considera uma função objectivo com custos de *setup* e de materiais. Para lidar com o enorme número de restrições, o autor simplifica o modelo, transformando-o num modelo não-linear, que lineariza por aproximação. Com esse método, Belov resolve apenas 7 das 16 instâncias usadas por Vanderbeck, mas obtém, em média, melhores resultados que o KOMBI234 de Foerster e Waschër [9], em testes com 12 classes de instâncias.

Aloisio *et al.* [1] abordam o PMP para o caso especial em que no máximo dois itens podem ser cortados a partir do mesmo rolo. Os autores apresentam duas formulações para o problema, e derivam diferentes resultados ligados à existência de soluções.

Com este artigo, pretendemos contribuir para a resolução do PMP com uma nova família de limites inferiores. Esse limite é obtido resolvendo uma sequência de problemas de satisfação de restrições. Adicionalmente, apresentamos novas restrições válidas para o problema. As experiências computacionais que foram conduzidas baseiam-se em instâncias da literatura usadas em todas as publicações que descrevem abordagens exactas de resolução. Os resultados que obtivemos mostram que esse limite pode ser mais forte que o limite contínuo do modelo de geração de colunas de Vanderbeck [19], e é obtido em tempos computacionais muito reduzidos.

O paradigma da Programação por Restrições tem sido usado com sucesso para resolver alguns problemas de optimização combinatória [12, 7]. Os modelos de Programação por Restrições são mais expressivos que os modelos de Programação Matemática. Permitem representar mais facilmente restrições difíceis do tipo não-linear, ou restrições lógicas, por exemplo. A forma como tira partido das restrições é outra das grandes forças dessa técnica. As restrições são usadas de forma activa num processo de dedução que conduz a reduzir o domínio das variáveis e detectar inconsistências. Os problemas associados são problemas de satisfação de restrições nos quais se procura determinar se existe pelo menos uma solução que satisfaça todas as restrições do modelo. Recentemente, muitos esforços têm sido feitos no sentido de aproveitar a complementaridade que existe entre o paradigma de Programação por Restrições e a Programação Inteira [16].

A estrutura do artigo é a seguinte. Dado que comparamos os nossos resultados com contribuições

recentes baseadas em modelos de Programação Inteira, começamos na Secção 2 por introduzir alguns desses modelos, nomeadamente um modelo não-linear compacto e um modelo de geração de colunas. Na Secção 3, apresentamos o modelo de Programação por Restrições no qual se baseia o cálculo dos limites inferiores para o PMP, e introduzimos também novas famílias de restrições válidas. O procedimento seguido para o cálculo dos limites inferiores é introduzido na Secção 3.3. Na Secção 4, descrevemos os resultados computacionais obtidos em instâncias reais da literatura. As conclusões finais são apresentadas na Secção 5.

2 Modelos de Programação Inteira

Neste artigo, abordamos o PMP a uma dimensão. Assumimos que apenas z_{CSP} rolos podem ser usados, sendo esse valor igual ao número mínimo de rolos necessários para cortar todos os itens. Esse valor pode ser calculado resolvendo o problema de corte standard correspondente. As procuras de cada item devem ser satisfeitas exactamente. Consideramos ainda que os itens estão ordenados por ordem decrescente dos seus tamanhos.

Sejam $\underline{z}$ e $\overline{z}$ um limite inferior e superior para o número de padrões diferentes, respectivamente. O primeiro desses limites pode ser obtido resolvendo um problema de empacotamento com os mesmos itens que o PMP e todas as procuras iguais a 1. O limite superior pode ser obtido através da solução óptima do problema de corte standard correspondente.

2.1 Modelo compacto não-linear

Em [19], Vanderbeck define pela primeira vez uma formulação compacta e não-linear para o PMP. O modelo é definido através de váriaveis de afectação dos itens aos rolos, e de variáveis que determinam o número de vezes que um dado padrão é usado. Além de ser não-linear, o modelo tem também um alto grau de simetria. A sua definição é a que segue.

$$\min \sum_{k=1}^{z_{CSP}} y_k \tag{1}$$

$$\text{s.t.} \sum_{k=1}^{z_{CSP}} z_k x_{ik} = b_i,\ i = 1, \ldots, m, \tag{2}$$

$$\sum_{k=1}^{z_{CSP}} z_k \leq z_{CSP}, \tag{3}$$

$$z_k \leq z_{CSP} y_k,\ k = 1, \ldots, z_{CSP}, \tag{4}$$

$$\sum_{i=1}^{m} w_i x_{ik} \leq W y_k,\ k = 1, \ldots, z_{CSP}, \tag{5}$$

$$x_{ik} \in \mathbb{N},\ i = 1, \ldots, m,\ k = 1, \ldots, z_{CSP}, \tag{6}$$

$$y_k \in \{0, 1\},\ k = 1, \ldots, z_{CSP}, \tag{7}$$

$$z_k \in \mathbb{N},\ k = 1, \ldots, z_{CSP}. \tag{8}$$

As variáveis x_{ik} representam o número de vezes que um item i é considerado no padrão k. O número de vezes que um padrão k é repetido é representado pela variável z_k, enquanto as variáveis binárias

y_k traduzem o facto desse padrão estar ou não presente na solução. A função objectivo (1) consiste na minimização da soma dessas últimas variáveis, *i.e.* do número de padrões diferentes que são efectivamente usados. As restrições (2) garantem que a procura de cada item é satisfeita exactamente. Essas restrições são não-lineares. A restrição (3) define um limite superior para o número de rolos que podem ser usados. As restrições (5) garantem que o tamanho total dos itens não excede o tamanho W do rolo a partir do qual irão ser cortados. Essas restrições são restrições de mochila.

2.2 Modelo de geração de colunas de Vanderbeck [19]

Em [19], Vanderbeck descreve um modelo de geração de colunas que resulta de uma decomposição de Dantzig-Wolfe do modelo compacto (1)-(8), no qual se dualizam as restrições (2) e (3). No problema mestre (9)-(12), cada coluna representa um padrão de corte admissível, associado a uma possível multiplicidade $n \in \{1, \ldots, n^{max}\}$, com $n^{max} = \min\{z_{CSP} - \underline{z} + 1, \max_i\ b_i\}$.

$$\min \sum_{k \in K} \sum_{n=1}^{u_k} \lambda_{kn} \tag{9}$$

$$\text{s.t. } \sum_{k \in K} \sum_{n=1}^{u_k} n a_{ik} \lambda_{kn} = b_i,\ i = 1, \ldots, m, \tag{10}$$

$$\sum_{k \in K} \sum_{n=1}^{u_k} n \lambda_{kn} \leq z_{CSP}, \tag{11}$$

$$\lambda_{kn} \in \{0, 1\},\ k \in K.\ n = 1, \ldots, u_k. \tag{12}$$

As variáveis binárias λ_{kn} tomam o valor 1 se um padrão k com multiplicidade n é usado e 0, caso contrário. Os coeficientes a_{ik} representam o número de itens i presentes no padrão k. O parâmetro $u_k = \min\limits_{i=1,\ldots,m} \left\lfloor \frac{b_i}{a_{ik}} \right\rfloor$ representa um limite superior para o valor da multiplicidade do padrão k. Uma melhoria para este limite é proposta em [2]. Designando por l_{CSP} o desperdício total associado à solução óptima do problema de corte correspondente, os autores propuseram o seguinte valor para u_k:

$$u_k = \min\left\{ \min_{i=1,\ldots,m} \left\lfloor \frac{b_i}{a_{ik}} \right\rfloor, \left\lfloor \frac{l_{CSP}}{W - \sum_{i=1}^{m} w_i a_{ik}}. \right\rfloor \right\}$$

Como o modelo implica a enumeração de um número exponencial de colunas (variáveis binárias associadas aos padrões), a melhor forma de o resolver passará necessariamente por uma enumeração implícita e uma geração dinâmica dos subconjunto de colunas necessárias. O subproblema de geração de colunas que resulta da decomposição é um problema de mochila quadrático. O modelo (13)-(16) representa esse subproblema. As variáveis duais associadas à restrições (10) e (11) são representadas respectivamente por π e ρ.

$$\max n \left(\sum_{i=1}^{m} \pi_i x_i + \rho \right) \tag{13}$$

$$\text{s.t.} \sum_{i=1}^{m} w_i x_i \leq W, \tag{14}$$

$$nx_i \leq b_i, \ i = 1, \ldots, m, \tag{15}$$

$$n \in \{1, \ldots, n^{max}\}, \ x_i \geq 0 \ \text{ and integer}, \ i = 1, \ldots, m. \tag{16}$$

Para contornar a não-linearidade do subproblema, Vanderbeck resolve uma sequência de problemas lineares que são obtidos fixando o valor da multiplicidade. No pior caso, por cada iteração do método de geração de colunas, é resolvido um problema de mochila por cada valor possível de n. Em [19], Vanderbeck mostra que em muitos casos a solução óptima do subproblema permanece óptimo para vários valores sucessivos de n, o que permite que sejam resolvidos menos problemas. Em [2], Alves e Carvalho mostram que a adição de uma restrição ao limite total de desperdício introduzida no subproblema permite reduzir ainda mais o número de subproblemas que têm de ser resolvidos. Além disso, a restrição permite também aumentar o valor do limite inferior obtido com a relaxação linear do modelo.

3 Um modelo de Programação por Restrições

Nesta Secção, descrevemos o modelo de Programação por Restrições no qual se baseia o novo limite inferior. O princípio subjacente ao nosso modelo assenta numa perspectiva diferente do problema. A questão que se coloca está em determinar o número mínimo de multiplicidades de padrões que garante que todas as procuras de itens possam ser satisfeitas exactamente. Dito de outra forma, quaisquer que sejam os padrões escolhidos, as procuras dos itens deverão poder ser expressas como combinações lineares das respectivas multiplicidades. Sendo as multiplicidades dos padrões incógnitas do problema, essas condições são expressas na forma de restrições não-lineares. É o que acontece por exemplo no modelo (1)-(8) com as restrições (2). Nesta Secção, descrevemos formas de reduzir a dimensão do modelo. Introduzimos também uma nova família de restrições válidas para o problema.

3.1 Formulação

O modelo de Programação por Restrições que iremos descrever nesta Secção pode ser visto como uma relaxação do modelo (1)-(8) com restrições adicionais, no qual se consideram de forma particular as restrições (2) e (3). O facto de se considerarem isoladamente essas restrições permite efectuar um conjunto de simplificações que contribuem para a redução da dimensão do modelo. A Programação por Restrições permite-nos aqui lidar com um conjunto de restrições complexas. Em [19], Vanderbeck contorna as não-linearidades enumerando todos os possíveis valores das multiplicidades. Essa estratégia tem implicações directas na dimensão do modelo, e no tempo necessário para resolver, por exemplo, o subproblema de geração de colunas.

Como vimos atrás, a multiplicidade de um padrão representa o número de vezes que esse padrão é repetido. Seja X um vector de variáveis que representam multiplicidades, sendo X_j, $j = 1, \ldots, \overline{z}$, o valor da j-ésima multiplicidade em X. O vector X representa o conjunto de multiplicidades usadas numa solução do PMP. As variáveis de X poderão ter valores iguais. Impomos que $|X| = \overline{z}$. Consideramos ainda que as multiplicidades que pertencem a X estão ordenadas por ordem crescente à excepção das últimas que poderão ter o valor 0, se menos do que $\overline{z}$ multiplicidades forem efectivamente usadas. Com

este esquema, sabemos que X_1 será a multiplicidade de menor valor da solução, X_2 a segunda menor, e assim em diante. Por outro lado, se para um determinado valor de k se verifica $X_k \neq 0$ e $X_{k+1} = 0$, então a solução representada por X usará exactamente k multiplicidades (padrões diferentes), sendo X_k a multiplicidade de maior valor. Temos assim que:

$$X_j \leq X_{j+1} \vee X_{j+1} = 0,\ j = 1, \ldots, \overline{z} - 1,$$
$$X_j = 0 \Rightarrow X_{j+1} = 0,\ j = 1, \ldots, \overline{z} - 1.$$

As restrições seguintes determinam que a combinação X de multiplicidades deve ser tal que a procura de cada item possa ser expressa como uma combinação linear dos X_j correspondentes, $j = 1, \ldots, \overline{z}$:

$$\sum_{j=1}^{\overline{z}} A_{ij} X_j = b_i,\ i = 1, \ldots, m. \tag{17}$$

As variáveis A_{ij} são variáveis inteiras gerais que representam o número de vezes que uma multiplicidade j deve ser usada de forma a poder ser recuperado o valor da procura b_i. Um limite superior válido para as variáveis A_{ij} é dado por $\left\lfloor \frac{W}{w_i} \right\rfloor$, $\forall j$. As restrições (17) podem ser associadas às restrições de procura (2), enquanto as variáveis A_{ij} podem ser interpretadas como sendo o número de vezes que um item i é considerado num padrão j de multiplicidade X_j. De notar contudo que nós relaxamos as restrições de mochila do modelo (1)-(8), que são as restrições que condicionam mais significativamente o modo como podem ser afectados os itens aos rolos. Isso permite-nos concentrar na relação entre as procuras dos itens e as multiplicidades consideradas numa solução do PMP, e assim aplicar diferentes procedimentos para reduzir o número de restrições (17) e variáveis A_{ij}.

Em primeiro lugar, em situações em que haja dois itens com igual procura, só precisamos de considerar em (17) o maior item entre todos aqueles que têm a mesma procura. Assumindo que dois itens s e t com $w_s \geq w_t$ têm o mesmo valor de procura ($b_s = b_t$), as restrições (17) para esses dois itens podem ser expressas como segue.

$$A_{s1}X_1 + A_{s2}X_2 + A_{s3}X_3 + \ldots + A_{s\overline{z}}X_{\overline{z}} = b_s \tag{18}$$
$$A_{t1}X_1 + A_{t2}X_2 + A_{t3}X_3 + \ldots + A_{t\overline{z}}X_{\overline{z}} = b_t \tag{19}$$

Claramente, dado que $A_{sj} \leq \left\lfloor \frac{W}{w_s} \right\rfloor \leq \left\lfloor \frac{W}{w_t} \right\rfloor$, se (18) é satisfeita, o mesmo acontecerá com (19).

Além disso, em segundo lugar, as procuras de itens que são iguais à soma de duas (ou mais) outras procuras podem também ser excluídas das restrições (17) se algumas condições forem cumpridas. Por exemplo, considerem-se três itens r, s e t tais que $b_r = b_s + b_t$. Para esses itens, as restrições (17) consistem no seguinte:

$$A_{r1}X_1 + A_{r2}X_2 + A_{r3}X_3 + \ldots + A_{r\overline{z}}X_{\overline{z}} = b_r, \tag{20}$$
$$A_{s1}X_1 + A_{s2}X_2 + A_{s3}X_3 + \ldots + A_{s\overline{z}}X_{\overline{z}} = b_s, \tag{21}$$
$$A_{t1}X_1 + A_{t2}X_2 + A_{t3}X_3 + \ldots + A_{t\overline{z}}X_{\overline{z}} = b_t. \tag{22}$$

Se $\left\lfloor \frac{W}{w_r} \right\rfloor \geq \left\lfloor \frac{W}{w_s} \right\rfloor + \left\lfloor \frac{W}{w_t} \right\rfloor$, então qualquer conjunto X que satisfaça (21) e (22) irá também satisfazer (20). O valor das variáveis A_{rj} pode ser obtido somando A_{sj} com A_{tj}, $\forall j$. Consequentemente, a restrição (20)

e as correspondentes variáveis A_{rj} poderão ser removidas do modelo.

A restrição (3) é facilmente expressa no nosso modelo da forma seguinte

$$\sum_{j=1}^{\overline{z}} X_j \leq z_{CSP}.$$

3.2 Novas famílias de cortes

Dado que as procuras devem ser satisfeitas exactamente, e portanto que a sobreprodução de itens não é permitida, um item de procura b_i só poderá ser considerado em padrões cujas multiplicidades sejam menores ou iguais a b_i. Sejam r_i e p_i limites inferiores para o número de rolos e padrões diferentes necessários para cortar todos os itens de procura inferior ou igual a b_i, respectivamente. Seja ainda m' o número de procuras diferentes. Um limite inferior para o valor de p_i é dado por $\left\lceil \frac{r_i}{b_i} \right\rceil$. Por definição, os itens de procura menor ou igual a b_i têm de ser cortados a partir de pelo menos r_i rolos. Dado que esses itens só podem ser considerados em padrões de multiplicidade menor ou igual a b_i, serão precisos pelo menos $\left\lceil \frac{r_i}{b_i} \right\rceil$ padrões diferentes para os cortar.

Por definição, o número de rolos e padrões diferentes usados no corte de itens de procura menor ou igual a b_i terão de ser maiores que os respectivos limites r_i e p_i. Essas restrições são expressas no nosso modelo da forma que segue:

$$\sum_{n=1}^{\min\{b_i, n^{max}\}} n\ count(X, n) \geq r_i,\ i = 1, \ldots, m', \tag{23}$$

$$\sum_{n=1}^{\min\{b_i, n^{max}\}} count(X, n) \geq p_i,\ i = 1, \ldots, m. \tag{24}$$

A expressão $count(X, n)$ representa o número de elementos de X que são iguais a n. De notar que, se essas restrições são fáceis de adicionar ao nosso modelo de Programação por Restrições, o mesmo já não acontece quando as tentamos adicionar ao modelo (1)-(8). Considerar essas restrições nesse modelo implicaria passar para um modelo com um número pseudo-polinomial de variáveis e restrições. Em particular, para formular as restrições (23) em (1)-(8), seria necessário adicionar variáveis binárias y_{kn} (uma por cada multiplicidade n até n^{max}) que tomariam o valor 1 se o padrão k fosse usado n vezes. As seguintes restrições completariam o modelo:

$$\begin{aligned}
& z_k = n y_{kn},\ k = 1, \ldots, z_{CSP},\ n = 1, \ldots, n^{max}, \\
& \sum_{n=1}^{b_i} n y_{kn} \geq r_i,\ i = 1, \ldots, m, \\
& y_{kn} \in \{0, 1\},\ k = 1, \ldots, z_{CSP},\ n = 1, \ldots, n^{max}.
\end{aligned}$$

Essas restrições permitem-nos ainda reduzir o domínio das variáveis $X_j = \{1, \ldots, n^{max}\}$. Por definição, o domínio das variáveis X_j é tal que $X_j \in [1, n^{max}],\ \forall j \in \{1, \ldots, \underline{z}\}$ e $X_j \in [0, n^{max}],\ \forall j \in \{\underline{z}+1, \ldots, \overline{z}\}$. Por exemplo, dado que os itens com a procura mais baixa só poderão ser considerados em padrões com multiplicidade menor ou igual que essa procura, a primeira multiplicidade X_1 nunca será maior que essa procura mais baixa, e assim, temos que $X_1 \leq \min_{i=1,\ldots,m}\{b_i\}$. Podemos generalizar esse

resultado aos valores de procuras diferentes do seguinte modo:

$$X_j \leq b_i,\ i = 1, \ldots, m',\ j = 1, \ldots, p_i,$$

sendo m' o número de procuras com valores diferentes ($m' \leq m$). Os itens de procura menor ou igual a b_i têm de ser cortados a partir de pelo menos p_i padrões diferentes. Isso implica que as primeiras p_i multiplicidades de X, $i \in \{1, \ldots, m'\}$, tenham de ser sempre menores ou iguais a b_i.

3.3 Cálculo do novo limite inferior para o PMP

Como já foi referido anteriormente, um limite inferior para o PMP pode ser calculado através da resolução de um problema de empacotamento em que todas as procuras são iguais a 1, e em que todos os itens e rolos correspondem aos do problema original. Isso acontece na medida em que essa solução corresponde também ao menor número de padrões diferentes necessários para cortar aquele conjunto de m itens uma vez que existe apenas um exemplar de cada item. Seja lb o valor desse limite.

O valor desse limite inferior pode ser melhorado recorrendo ao modelo de Programação por Restrições descrito na Secção anterior. Para esse efeito, é resolvida uma sequência de problemas de satisfação de restrições conforme passamos a descrever. O primeiro passo consiste em resolver um problema de satisfação de restrições definido a partir do nosso modelo, e ao qual se adiciona a restrição $X_{lb+1} = 0$. Resolver esse problema corresponde a perguntar se é possível satisfazer todas as restrições do modelo de Programação por Restrições com um máximo de lb multiplicidades (elementos positivos em X). Se o problema não tiver solução, o limite inferior pode ser incrementado de uma unidade, *i.e.* lb passará a ter o valor $lb + 1$. O processo é repetido até que o problema de satisfação de restrições tenha solução. Nesse caso, o procedimento termina sendo o novo limite inferior igual ao último valor de lb.

4 Resultados computacionais

Para avaliar a qualidade do limite inferior, foram conduzidas uma série de experiências computacionais num conjunto de instâncias reais da literatura. Usamos em particular as instâncias que Vanderbeck usou em [19]. Essas instâncias foram também usadas noutras publicações que descrevem abordagens exactas de resolução [2, 3].

Para resolver o nosso modelo de Programação por Restrições, usámos o ILOG CP Optimizer 1.0 [14]. Para resolver o modelo de geração de colunas (9)-(12), recorremos a algumas rotinas de optimização do CPLEX 10.2 [13]. Os testes foram todos conduzidos num PC com um processador Intel Core Duo de 2.20 GHz e 2GB de RAM.

Na Tabela 1, comparamos resultados obtidos usando um modelo de Programação por Restrições com os resultados obtidos com o modelo de geração de colunas (9)-(12) de Vanderbeck. As entradas da tabela representam a seguinte informação:

$Inst.$	:	nome da instância;
m	:	número de itens diferentes;
lb	:	valor do limite inferior calculado com base no problema de empacotamento correspondente;
z_{GC}	:	limite inferior contínuo dado pela relaxação linear de (9)-(12);
t_{GC}	:	tempo necessário à resolução da relaxação linear de (9)-(12) (em segundos);
lb_{PR}	:	limite inferior obtido através do modelo de Programação por Restrições;
t_{PR}	:	tempo necessário ao cálculo do limite inferior baseado no modelo de Programação por Restrições (em segundos);
$melhor$	:	um * nessa coluna identifica uma instância para a qual o limite obtido com o modelo de Programação de Restrições é maior do que aquele que é obtido com o modelo de geração de colunas;
$igual$	:	as entradas assinaladas com * nessa coluna correspondem às instâncias para as quais o limite obtido com o modelo de Programação de Restrições é ígual ao do modelo de geração de colunas.

	Inst.	m	lb	z_{GC}	t_{GC}	lb_{PR}	t_{PR}	$melhor$	$igual$
1	kT03	7	3	4,77	0,06	5	0,47		*
2	kT05	10	4	5,65	0,61	5	0,19		
3	kT01	5	1	2,1	0,05	2	0,01		
4	kT02	24	13	15,93	0,12	14	0,19		
5	kT04	16	6	6,74	1,08	7	0,01		*
6	d16p6	16	6	6,74	1,19	7	0,02		*
7	7p18	7	2	3,74	0,69	4	0,45		*
8	d33p20	23	5	6,18	2,85	6	0,11		
9	12p19	12	2	2,89	4,54	4	0,06	*	
10	d43p21	32	7	7,86	39,39	8	0,23		*
11	kT06	9	1	1,75	18,38	3	0,17	*	
12	kT07	11	2	2,86	11,43	3	0,45		*
13	14p12	14	2	3,75	8,2	4	0,37		*
14	kT09	14	2	3,65	47,18	4	0,87		*
15	11p4	11	1	2,48	21,63	4	3,78	*	
16	30p0	26	4	5,51	120,21	6	26,66		*
médias					17,35		2,13		

Tabela 1: Resultados computacionais para instâncias reais [19]

O tempo médio necessário para calcular o limite inferior através de um modelo de Programação por Restrições é cerca de 8 vezes inferior quando comparado com o modelo de geração de colunas de Vanderbeck. Na grande maioria das instâncias, o tempo de computação é significativamente menor

quando se usa um modelo de Programação por Restrições.

Em termos da qualidade do limite inferior, o modelo de Programação por Restrições fornece um limite que é igual ou superior ao limite do modelo de geração de colunas em 12 das 16 instâncias. O limite é superior ao do modelo de geração de colunas em 3 casos. Em todos esses casos, o tempo de computação é substancialmente reduzido. Em 4 instâncias, o limite inferior piora, mas continua a ser calculado em tempo muito reduzido.

5 Conclusões

O Problema de Minimização de Padrões é um problema relevante no domínio dos problemas de corte e empacotamento. O número de publicações que lhe foram dedicadas atesta essa realidade. No entanto, muitas das abordagens propostas centram-se em procedimentos heurísticos sendo poucos os resultados de abordagens de resolução exacta descritos na literatura. Neste artigo, contribuimos com um novo limite inferior que supera os resultados obtidos com um modelo de geração de colunas ao nível do estado-da-arte.

O modelo de Programação por Restrições que descrevemos permite tirar partido de um conjunto de restrições complexas do problema. Nos modelos de geração de colunas, essas restrições são formuladas através da enumeração de todos os valores possíveis para as multiplicidades dos padrões. De acordo com os testes que realizámos, uma abordagem baseada em Programação por Restrições parece adaptar-se muito bem a esse problema. Por um lado, a qualidade dos limites é melhorada em muitos casos. Em todos eles, o tempo necessário para calcular o limite é claramente inferior ao tempo necessário para calcular o limite contínuo do modelo de geração de colunas.

Agradecimentos

Este trabalho foi parcialmente financiado pela Fundação para a Ciência e a Tecnologia através da Bolsa de Doutoramento SFRH/BD/39607/2007 de Rita Macedo. Foi desenvolvido no Grupo de Engenharia de Sistemas, Optimização e Investigação Operacional do Centro de Investigação Algoritmi da Universidade do Minho.

Referências

[1] A. Aloisio, C. Arbib, and F. Marinelli. Cutting stock with no three parts per pattern: Work-in-process and pattern minimization. Technical report, Dipartimento di Informatica, Università degli Studi dell'Aquila, via Vetoio, I-67010 Coppito, L'Aquila, Italy, 2008.

[2] C. Alves and J. M. Valério de Carvalho. A branch-and-price-and-cut algorithm for the pattern minimization problem. *RAIRO Operations Research (in press)*, 2008.

[3] G. Belov. *Problems, models and algorithms in one- and two- dimensional cutting.* PhD thesis, Dresden University, 2003.

[4] C. Chen, S. Hart, and W. Tham. A simulated annealing heuristic for the one-dimensional cutting stock problem. *European Journal of Operational Research*, 93:522–535, 1996.

[5] F. Clautiaux, C. Alves, and J. M. Valério de Carvalho. A survey of dual-feasible and superadditive functions. *Annals of Operations Research (accepted)*, 2008.

[6] A. Diegel, M. Chetty, S. Van Schalkwyck, and S. Naidoo. Setup combining in the trim loss problem - 3-to-2 & 2-to-1. Working Paper, Business Administration, University of Natal, Durban, First Draft, 1993.

[7] T. Fahle, U. Junker, S. Karisch, N. Kohl, M. Sellmann, and B. Vaaben. Constraint programming based column generation for crew assignment. *Journal of Heuristics*, 8(1):59–81, 2002.

[8] S. Fekete and J. Schepers. New classes of fast lower bounds for bin packing problems. *Mathematical Programming*, 91:11–31, 2001.

[9] H. Foerster and G. Wäscher. Pattern reduction in one-dimensional cutting stock problems. *International Journal of Production Research*, 38(7):1657–1676, 2000.

[10] P. Gilmore and R. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859, 1961.

[11] R. Haessler. Controlling cutting pattern changes in one-dimensional trim problems. *Operations Research*, 23(3):483–493, 1975.

[12] J. Hooker. An integrated method for planning and scheduling to minimize tardiness. *Constraints*, 11(2-3):139–157, 2006.

[13] Ilog. *Ilog CPLEX 10.0 Reference Manual*. 9, rue de Verdun, BP 85, F-92453, Gentilly, France, 2006.

[14] Ilog. *Ilog CP Optimizer Reference Manual*. 9, rue de Verdun, BP 85, F-92453, Gentilly, France, 2007.

[15] C. McDiarmid. Pattern minimisation in cutting stock problems. *Discrete Applied Mathematics*, 98:121–130, 1999.

[16] M. Milano. *Constraint and integer programming: toward a unified methodology*. Kluwer Academic Publishers, 2004.

[17] J. Teghem, M. Pirlot, and C. Antoniadis. Embedding of linear programming in a simulated annealing algorithm for solving a mixed integer production planning problem. *Journal of Computational and Applied Mathematics*, 64:91–102, 1995.

[18] S. Umetani, M. Yagiura, and T. Ibaraki. One-dimensional cutting stock problem to minimize the number of different patterns. *European Journal of Operational Research*, 146:388–402, 2003.

[19] F. Vanderbeck. Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem. *Operations Research*, 48(6):915–926, 2000.

[20] G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183:1109–1130, 2007.

[21] H. Yanasse and M. Limeira. A hybrid heuristic to reduce the number of different patterns in cutting stock problems. *Computers and Operations Research*, 33:2744–2756, 2006.