# GERAÇÃO DE COLUNAS

**José Valério de Carvalho,**
Departamento de Produção e Sistemas,
Escola de Engenharia, Universidade do Minho

## Introdução

A Programação Inteira (PI) é uma técnica da Investigação Operacional largamente usada na modelação e resolução de problemas. É hoje possível abordar problemas muito complexos, considerando as principais restrições e questões suscitadas em problemas reais. A investigação nesta área tem tido um enorme impacto económico em áreas como a logística e a distribuição, as telecomunicações ou os transportes aéreos, urbanos e de massas.

Parte deste sucesso é devido ao uso de modelos de geração de colunas, que permitem abordar grandes instâncias de problemas de programação inteira, usando modelos reformulados através do método de decomposição de Dantzig-Wolfe. O método de geração de colunas é conhecido desde os finais dos anos 50 [3 ,4], mas esta técnica ganhou um novo impulso quando surgiram ideias sobre a forma de compatibilizar geração de colunas com o método de partição e avaliação, para obter soluções óptimas inteiras para os modelos, dando origem ao método de partição e geração de colunas, designado, na literatura anglo-saxónica, por branch-and-price.

Embora os dois métodos fossem conhecidos houvera bastante tempo, só nos anos 80 é que se implementou um algoritmo que encontrava soluções óptimas inteiras para um problema de encaminhamento de veículos [10]. Novas ideias e um enorme incremento da investigação nesta área tornaram possível resolver problemas de maior dimensão e abordar problemas reais complexos. Em problemas reais, mesmo nos casos em que não é possível obter uma solução óptima inteira, é tipicamente possível obter, no processo de resolução, soluções de muito boa qualidade, com custos que comprovadamente não diferem mais do que uma pequena percentagem do valor óptimo.

O que contribuiu para o longo hiato entre os primórdios da investigação nesta área e o trabalho desenvolvido após os anos 90 foi a competição entre a geração de colunas e outros métodos. O livro de Lasdon, intitulado Optimization Theory for Large Systems, publicado em 1970 [7], é essencialmente sobre geração de colunas. Ironicamente, o sucesso dos trabalhos de Held e Karp [5,6] na resolução do problema do caixeiro viajante, datados de 1970 e 1971, motivou toda a comunidade de optimização para o uso da relaxação lagrangeana, uma técnica alternativa que tem relações muito estreitas com o método de decomposição de Dantzig-Wolfe [9]. Durante duas décadas, a relaxação lagrangeana foi um campo fecundo de investigação, tendo praticamente substituído a utilização da geração de colunas na grande maioria das aplicações.

Foi o trabalho de Savelsbergh no problema de afectação generalizado [11] que mostrou que a geração de colunas poderia ter um melhor desempenho que a relaxação lagrangeana, apesar de teoricamente serem equivalentes do ponto de vista de limites inferiores. A explicação avançada é que a informação dual mais completa fornecida pelo Método Simplex ajuda a uma melhor convergência, ao contrário do que acontece no método do subgradiente que, apesar da prova teórica de que converge, tem por vezes na prática dificuldade em encontrar o caminho para a solução óptima em tempo razoável.

Uma questão essencial que concorreu para assegurar a competitividade dos algoritmos de geração de colunas foi o aumento da eficiência dos packages de programação linear. Se tal não acontecesse, o método de geração de colunas, apesar de robusto, não seria porventura competitivo com a relaxação lagrangeana, que usa pouca informação e tem uma carga computacional muito menos pesada. Nos dias de hoje, a geração de colunas compete com packages que incorporam grandes desenvolvimentos teóricos, com a grande desvantagem de requerer geralmente a implementação de código computacional adaptado a problemas específicos.

Este artigo apresenta alguns dos principais conceitos e das ideias condutoras da área de geração de colunas em 17500 caracteres sem espaços, e destina-se a um público que tem conhecimentos básicos sobre programação linear e dualidade. Esperamos que ele possa suscitar o interesse noutras leituras [1,8].

## Decomposição de Dantzig-Wolfe

Os poliedros podem ser representados de duas formas, que se pode mostrar que são equivalentes através do Teorema de Minkowski [15]. A primeira é através da intersecção de um conjunto de semi-espaços, cada um deles definido por uma restrição, e a segunda é através de um conjunto de pontos, os vértices do poliedro. A primeira forma é a normalmente adoptada para definir a região de soluções admissíveis de um problema de programação linear, enquanto a segunda está na base dos modelos que resultam da decomposição de Dantzig-Wolfe.

A metodologia e os resultados que vamos abordar podem ser aplicados no caso geral, em que os poliedros são abertos, i.e., contêm raios (ou seja, semi-rectas) e não podem ser colocados dentro de uma caixa do respectivo espaço dimensional. Aqui, vamos apenas analisar a situação mais simples, em que a região de soluções admissíveis do modelo de programação linear é uma região fechada. Doravante, passaremos a designar esta região admissível por poliedro.

A Decomposição de Dantzig-Wolfe (DW) é aplicada a um modelo de programação linear (PL), que vamos designar por modelo original, cujas restrições podem ser divididas em dois conjuntos: um conjunto de restrições gerais e um conjunto designado por com uma estrutura especial:

$$\min\ z_{PL} := c^T x \quad (1)$$

$$\text{suj. a } A x \geq b \quad (2)$$

$$x \in X \quad (3)$$

$$x \geq 0 \quad (4)$$

em que $A \in R^{m \times n}$, $b \in R^m$, $c \in R^n$, e $x \in R^n$ é um vector de variáveis de decisão. A região de soluções admissíveis deste modelo é $X_{PL} = \{x \in R^n_+ : A x \geq b, x \in X\}$.

É comum os modelos serem compostos por vários conjuntos de restrições. A título de exemplo, o problema de corte de rolos pode ser modelado com um conjunto de restrições que determinam que os itens cortados de um rolo não usam mais espaço do que a própria largura do rolo e um outro conjunto de restrições destinadas a garantir que o número de itens cortados são suficientes para satisfazer os pedidos dos clientes. Uma possível decomposição seria o primeiro conjunto de restrições gerais ser o conjunto de restrições de satisfação dos clientes, e o conjunto $X$ ser o conjunto de combinações admissíveis de itens em rolos, que normalmente são em número que é exponencialmente grande [2].

Conforme referido, a segunda forma de representar um poliedro é através dos seus vértices (pontos extremos). Seja $Q = \{Q_1, \ldots, Q_i, \ldots, Q_{|Q|}\}$ o conjunto de vértices do poliedro $X$. A envolvente convexa de um conjunto de pontos é o menor conjunto convexo que contém todos os pontos do conjunto. O poliedro (fechado) $X$ coincide então com a envolvente convexa dos seus vértices:

$$X = \text{Conv}\{Q_1, \ldots, Q_i, \ldots, Q_{|Q|}\},$$

e qualquer ponto $x$ do poliedro $X$ pode ser expresso como uma combinação convexa dos vértices de $X$. O poliedro $X$ e os seus pontos $x$ podem portanto ser definidos do seguinte modo:

$$X = \left\{ x \in R^n_+ : x = \sum_{i=1,\ldots,|Q|} \lambda_i Q_i,\ \sum_{i=1,\ldots,|Q|} \lambda_i = 1, \lambda_i \geq 0, i = 1, \ldots, |Q| \right\}. \quad (5)$$

Se substituirmos o valor de $x$ no modelo original, a rearranjando os termos, obtemos o seguinte modelo reformulado, também designado por modelo-DW:

$$\min\ z_{DW} := \sum_{i=1,\ldots,|Q|} (c^T Q_i) \lambda_i \quad (6)$$

$$\text{suj. a } \sum_{i=1,\ldots,|Q|} (A Q_i) \lambda_i \geq b \quad (7)$$

$$\sum_{i \in Q} \lambda_i = 1 \quad (8)$$

$$\lambda_i \geq 0, i = 1,\ldots,|Q| \quad (9)$$

As variáveis de decisão do modelo reformulado são as variáveis $\lambda_i$, cada uma delas correspondendo a um ponto extremo do poliedro $X$. Os elementos $c^T Q_i, i = 1,\ldots,|Q|$, são os coeficientes da função objectivo, e as colunas , $AQ_i, i = 1,\ldots,|Q|$ definem as restrições do modelo reformulado. A restrição da soma dos valores das variáveis $\lambda_i$ é normalmente designada por restrição de convexidade. O modelo-DW (6)-(9) é equivalente ao modelo original (1)-(4), dado que a definição de $X$ fornece a correspondência entre uma qualquer solução admissível do modelo-DW e uma solução admissível do modelo original.

### Geração de colunas

Uma vez que o número de vértices do conjunto $X$ é tipicamente grande, não sendo portanto praticável enumerar à partida todas as variáveis de decisão do problema reformulado (6) - (9), usa-se um procedimento de geração (diferida) de colunas, que pode ser sucintamente descrito da seguinte forma.

O algoritmo de geração de colunas começa com um modelo com um conjunto restrito de variáveis, designado por problema mestre restrito, que é optimizado. A informação dual do problema mestre restrito é utilizada num (ou em vários) problema(s) de optimização auxiliar(es), designado(s) por subproblema(s), que se destina(m) a encontrar a variável de decisão (representada por uma coluna) mais atractiva para inserir no problema mestre restrito, que é novamente reoptimizado. Estas operações são repetidas até que mais nenhuma coluna atractiva seja encontrada, obtendo-se então uma solução que é comprovadamente óptima.

No Método Simplex, o que determina se uma variável não-básica $j$ é atractiva é o seu custo reduzido, $c_j' = c_B B^{-1} A_j - c_j$, calculado com base no custo original da variável, $c_j$, na solução dual corrente, $c_B B^{-1}$, e na coluna $A_j$ (ver, por exemplo, [9]). No método de geração de colunas, é possível articular o problema mestre e o subproblema, porque se pode exprimir o custo reduzido (para o problema mestre) de um qualquer ponto $x$ do conjunto $X$ em termos das variáveis do modelo original, ou seja, através de coeficientes de custo das variáveis do modelo original. O custo reduzido é calculado com base na informação dual do problema mestre restrito, na coluna, $AQ_i$, e no custo da coluna, $c^T Q_i$. Assim, o subproblema, que é um problema no espaço das variáveis do modelo original, pode determinar qual o vértice do conjunto $X$ que corresponde à variável mais atractiva para inserir no problema mestre restrito.

A Figura 1 representa graficamente um problema no espaço das variáveis de decisão do modelo original. O conjunto $X = \text{Conv}\{Q_1, Q_2, Q_3\}$, e a região de soluções admissíveis, representada a sombreado, é o subconjunto de pontos não-negativos de $X$ que obedece às restrições gerais (neste caso, à esquerda do segmento $\overline{AB}$).

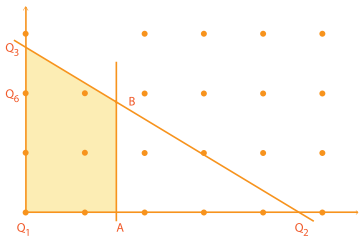

Figura 1: Região admissível do modelo original

A resolução do modelo reformulado pode ser acompanhada através da representação gráfica no espaço das variáveis de decisão do modelo original. Vamos supor que $\lambda_1$, correspondente ao vértice $Q_1$, é a única variável de decisão do problema mestre restrito. Podemos aplicar o Método Simplex, porque existe um vértice inicial admissível; se assim não fosse, teríamos de considerar variáveis artificiais no modelo reformulado, como se faz no Método do Grande M. A solução óptima do modelo mestre restrito é $\lambda_1 = 1$, porque a solução deve obedecer à restrição de convexidade. Nesta iteração, o espaço de soluções do modelo reformulado é um conjunto singular, o vértice $Q_1$.

Vamos supor que, com base na informação dual, a solução óptima do subproblema, que procura a solução mais atractiva no poliedro $X = \text{Conv}\{Q_1, Q_2, Q_3\}$, é o vértice $Q_2$. Vamos supor também que a variável de decisão correspondente, $\lambda_2$, representada pela coluna $AQ_2$ e com custo $c^T Q_2$, é atractiva para o modelo reformulado, traduzindo o facto de que, se for adicionada ao modelo reformulado, pode potencialmente melhorar o valor da sua função objectivo.

O problema mestre restrito é reoptimizado, agora com duas variáveis de decisão, uma correspondente ao vértice $Q_1$ e outra ao vértice $Q_2$. Como já vimos, o modelo-DW e o modelo original são equivalentes. A definição da região admissível é clara: são os pontos $x \in X$ que obedecem também a $Ax \geq b$ e $x \geq 0$. Nesta iteração, no modelo mestre restrito, não temos o conjunto $X$ mas apenas o segmento de recta que liga o vértice $Q_1$ ao vértice $Q_2$, pelo que a solução óptima do problema mestre restrito é o ponto $A$. As variáveis $\lambda_1$ e $\lambda_2$ tomam os valores fraccionários que traduzem a combinação convexa que dá o ponto $A$.

A resolução prosseguiria até se obter a solução óptima. Se o ponto óptimo fosse o ponto $B$ no espaço original, a solução óptima do modelo reformulado apareceria como uma combinação convexa das variáveis de decisão $\lambda_2$ e $\lambda_3$.

Neste exemplo, todos os vértices do conjunto $X$ estão envolvidos no algoritmo de geração de colunas. Em problemas de grande dimensão, para obter a solução óptima, tipicamente apenas é necessário incluir no problema mestre uma fracção muito pequena de todas as variáveis possíveis.

O método de Dantzig-Wolfe tem uma interpretação económica, que fornece uma perspectiva do funcionamento de processos de decisão descentralizados. É um método de decomposição, em que o problema inicial é decomposto em subproblemas de menor dimensão, que passam a ser coordenados pelo problema mestre. As restrições do problema principal traduzem o modo como as colunas propostas pelos subproblemas utilizam os recursos comuns disponíveis. Os subproblemas, de uma forma descentralizada e independente, competem por esses recursos comuns, usando apenas a informação dada pelas variáveis duais do problema principal, que são uma medida dos ganhos e perdas relativos à utilização desses recursos comuns. De certo modo, as variáveis duais correspondem aos preços de utilização dos recursos. No final, o sistema aceita um conjunto de propostas independentes que globalmente traduzem a máxima eficiência [3].

### Decomposição de Dantzig-Wolfe em Programação Inteira

O problema de programação inteira é representado como:

$$\min\ z_{PI} := c^T x \quad (10)$$

$$\text{suj. a } Ax \geq b \quad (11)$$

$$x \in X \quad (12)$$

$$x \geq 0 \text{ e inteiro} \quad (13)$$

devendo agora as variaveis de decisão ser inteiras, $x \in Z_+^n$. O conjunto de soluções admissíveis é o conjunto discreto de ponto $X_{PI} = \{x \in Z_+^n : Ax \geq b, x \in X\}$.

Quem lida com Programação Inteira sabe que é crucial obter modelos que, quando se relaxam as condições que obrigam as variáveis a serem inteiras, descrevam da forma mais próxima possível o espaço de soluções inteiras. Esses modelos são designados por modelos fortes. Usar modelos fortes permite geralmente reduzir o número de nós pesquisados na árvore do método de partição e avaliação, e encontrar a solução óptima mais rapidamente.

A envolvente convexa do conjunto $X_{PI}$ designado por $\text{Conv}\{X_{PI}\}$, é o modelo mais forte para um problema de PI. Usando PL, a solução óptima nunca seria fraccionária, porque os seus pontos extremos são todos inteiros. A questão é que pode ser muito difícil conhecer todas as restrições que são necessárias para definir o conjunto $\text{Conv}\{X_{PI}\}$. Torna-se necessário recorrer a outras alternativas, mais fracas do que $\text{Conv}\{X_{PI}\}$, mas que podem ser mais fortes do que a relaxação de PL, $X_{PL}$.

Uma das motivações principais para usar a decomposição-DW em PI é obter um modelo mais forte do que $X_{PL}$, o que é possível quando o conjunto $X$ tem características especiais. A decomposição-DW fornece um modelo forte se o poliedro $X$ não tiver a propriedade da integralidade, o que significa que seus pontos extremos não são todos inteiros.

Para obter um modelo forte, usa-se $\mathcal{X} \in \text{Conv}\{x \in X \text{ e inteiro}\} \subseteq X$, em vez de usar $x \in X$. Isso corresponde a impor as restrições de integralidade nas variáveis de decisão apenas no segundo conjunto de restrições do modelo original. O conjunto de soluções admissíveis do modelo reformulado, expresso em termos de variá-

veis $x$ do modelo original, é $X_{DWI}$ $\{x \in R_+^n : A\,x \geq b, x \in$ Conv $\{x \in X$ e inteiro$\}\}$. O modelo-DWI correspondente é o seguinte:

$$\min\ z_{DWI} := c^T x \quad (14)$$

$$\text{suj. a } A\,x \geq b \quad (15)$$

$$x \in \text{Conv}\{x \in X \text{ e inteiro}\} \quad (16)$$

$$x \geq 0 \quad (17)$$

Dado que Conv $\{X_{IP}\} \subseteq X_{DWI} \subseteq X_{PL}$ resulta que $z_{IP} \leq z_{DWI} \leq z_{PL}$ no caso de um problema de minimização. Note-se que, se o poliedro $X$ tiver a propriedade de integralidade, então $X$ = Conv $\{x \in X$ e inteiro$\}$, e $X_{DWI} = \{x \in X : A\,x \geq b, x \geq 0\}$. Neste caso, o modelo-DWI é tão forte como a relaxação de PL, e $z_{PI} \leq z_{DWI} = z_{PL}$.

A Figura 2 mostra como a decomposição-DWI pode ajudar a obter um modelo mais forte do que a relaxação de PL. O conjunto de soluções do problema inteiro é o conjunto discreto de pontos representados pelos círculos maiores. A envolvente convexa desses pontos é o conjunto Conv $\{X_{PI}\}$. O conjunto $X$ = Conv $\{Q_1, Q_2, Q_3\}$ , mas, se impusermos a restrição de as variáveis serem inteiras no subproblema, Conv $\{x \in X$ e inteiro$\}$ = Conv $\{Q_1, Q_4, Q_5, Q_6\} \subset X$. Da mesma forma, a região de soluções admissíveis do modelo de decomposição-DWI é a representada a sombreado, $X_{DWI}$ = Conv $\{Q_1, A, C, Q_5, Q_6\} \subset X_{PL}$ = Conv $\{Q_1, A, B, Q_3\}$, sendo que neste caso $X_{PI} \subset X_{DWI} \subset X_{PL}$.

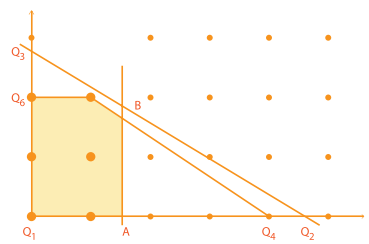


Figura 2: Região admissível do modelo de decomposição Dantzig-Wolfe Inteira

Resolver problemas de PI é mais difícil do que resolver problemas de PL, mas impor as restrições de integralidade apenas no subproblema pode não constituir um aumento muito significativo da carga computacional. No caso do problema de corte de rolos, por exemplo, consiste em resolver um problema de saco de mochila, que é, na prática, um problema de programação inteira relativamente fácil, mas os ganhos que se obtêm em termos de qualidade de modelos são muito significativos.

A escolha de uma decomposição de um modelo envolve uma partição das restrições, seleccionando as do conjunto das restrições gerais e as que definem o conjunto . Colocar mais restrições no conjunto dá origem tipicamente a um modelo mais forte, mas aumenta o esforço computacional de resolução do subproblema. Uma solução de compromisso deve balancear a qualidade do modelo mantendo o subproblema de programação inteira tratável.

## Método de Partição e Geração de Colunas

Mesmo com modelos mais fortes, a solução óptima da relaxação linear do modelo reformulado pode não ser inteira. Daí a necessidade de combinar o método de partição e avaliação com o método de geração de colunas.

Os algoritmos de partição e avaliação começam pela resolução da relaxação linear do modelo, que corresponde à raiz da árvore de pesquisa. Depois, são introduzidas restrições de partição que dão origem, em cada nó da árvore, a uma região admissível que é um subconjunto da região admissível na relaxação linear. Em cada nó da árvore de pesquisa, é resolvido um problema de PL para avaliar qual a melhor solução que se pode encontrar nesse subconjunto.

A principal dificuldade que se coloca na concepção de algoritmos de partição e geração de colunas é também de articulação: o subproblema deve continuar a identificar correctamente as colunas que são atractivas, e as que não são, depois de introduzir restrições de partição no problema mestre restrito. Tipicamente, em qualquer nó da árvore de pesquisa, é necessário gerar novas colunas para obter a solução óptima desse nó.

Os investigadores cedo descobriram que as restrições de partição não devem incidir directamente sobre as variáveis do modelo reformulado. De facto, pode acontecer que uma dada coluna colocada a zero por uma restrição de partição no problema mestre venha a revelar-se imediatamente a seguir, no subproblema, a coluna mais atractiva para ser colocada no problema mestre, tendo em vista vir a assumir um valor positivo, quando se torna básica. Este fenómeno é designado por regeneração de variáveis, e leva a um impasse.

Uma forma de criar esquemas de partição bem sucedidos é usar restrições de partição que possam ser expressas em termos das variáveis de decisão do modelo original. Face ao mostrado acima, isso não deveria ser surpresa; as restrições de partição inseridas no problema mestre relativas a um dado nó da árvore de pesquisa estabelecem o subconjunto do conjunto em que subproblema deve procurar os vértices atractivos, permitindo uma identificação correcta dos pontos extremos atractivos. Um exemplo é o modelo de fluxos em arcos para o problema de corte de rolos, que é um modelo com variáveis originais que fornece um esquema de partição que pode ser usado para resolver o modelo de Gilmore-Gomory de geração de colunas [2,12].

Mas nem todos os modelos originais fornecem de uma forma directa esquemas de partição que assegurem uma fácil articulação entre o problema mestre e o subproblema. Embora se possa mostrar que é sempre possível fazer a articulação [13], pode ser necessário considerar variáveis binárias adicionais no subproblema, alterando a sua estrutura, e eventualmente tornando-o um problema de programação inteira de muito difícil resolução.

Há muitos modelos de optimização com um número exponencial de colunas, semelhantes aos modelos reformulados, que são obtidos, não através de uma decomposição de Dantzig-Wolfe, mas a partir de colunas com uma estrutura definida. Levanta-se a questão de saber se há modelos originais que possam ajudar a construir esquemas de partição. Foi mostrado que, sob premissas relativamente suaves, é possível, a partir de modelos reformulados, construir modelos originais compactos, com um número polinomial de variáveis e de restrições [14].

## Outras questões e perspectivas

Esforços de investigação recentes são dedicados à obtenção de algoritmos mais rápidos, que ajudem a ultrapassar questões como a degenerescência e a estabilizar os valores das soluções duais, ajudando à convergência [1,8]. A área de geração de colunas continua a ser objecto de intensa investigação teórica e aplicada.

**Referências**

[1 ] C. Barnhart, E. Johnson, G. Nemhauser, M. Savelsbergh, P. Vance, Branch-and-Price: column generation for solving huge integer programs, Operations Research 46, 316-329, (1998).
[2] H. Ben Amor and J. Valério de Carvalho, "Cutting Stock Problems", in "Column Generation" (eds. G. Desaulniers, J. Desrosiers, and M. Solomon, GERAD), Springer (2005).
[3] G. Dantzig and P. Wolfe. Decomposition principle for linear programs. Operations Research 8, 101-111 (1960).
[4] L. Ford and D. Fulkerson, A suggested computation for maximal multicommodity network flows, Management Science 5, 97-101 (1958).
[5] M. Held and R. M. Karp, The traveling-salesman problem and minimum spanning trees, Operations Research 18, 1138-1167 (1970).
[6] M. Held and R. M. Karp, The traveling-salesman problem and minimum spanning trees: Part II, Mathematical Programming 1, 6-25 (1971).
[7] L. Lasdon, "Optimization Theory for Large Systems". Case Western Reserve University. The Macmillan Company. Collier-Macmillan Limited, London (1970).
[8] M. Lübbecke and J. Desrosiers, Selected topics in column generation, Operations Research 53, 1007-1023 (2005).
[9] G. L. Nemhauser, L. A. Wolsey, "Integer and Combinatorial Optimization", John Wiley and Sons (1999).
[10] J. Desrosiers, F. Soumis, M. Desrochers, Routing with time windows by column generation, Networks 14, 545-565 (1984).
[11] M. Savelsbergh, A branch-and-price algorithm for the generalized assignment problem, Operations Research 45, 831-841 (1997).
[12] J. Valério de Carvalho, Exact solution of bin-packing problems using column generation and branch-and-bound. Annals of Operations Research 86, 629-659 (1999).
[13] F. Vanderbeck and L. A. Wolsey, An exact algorithm for IP column generation, Operations Research Letters 19, 151-159 (1996).
[14] D. Villeneuve; J. Desrosiers, M. Lubbecke and F. Soumis, On compact formulations for integer programs solved by column generation, Annals of Operations Research, 139, 375-388 (2005).
[15] G. Ziegler, "Lectures on Polytopes", Graduate Texts in Mathematics 152, Springer-Verlag, New York, Inc. (1995).