

# Uma abordagem primal para a geração de planos cortantes disjuntivos mais separadores \*

Ana Ramires<sup>1</sup>, João Soares<sup>2</sup>

<sup>1</sup> Departamento de Matematica, Universidade Portucalense, 4200 Porto, Portugal.

<sup>2</sup> Departamento de Matematica, Universidade de Coimbra, 3000 Coimbra, Portugal. e-mail: [jsoares@mat.uc.pt](mailto:jsoares@mat.uc.pt) (corresponding author).

13 de Maio de 2004

**Abstract** In this article, we propose a generalization of Ph. Wolfe's algorithm (Wolfe, 1976) to finding the nearest point in a polytope according to any metric. The algorithm we propose is composed of an external cycle (Major cycle) and an internal cycle (Minor cycle), stops after a finite number of iterations and generates the same iterates as Wolfe's algorithm for the Euclidean metric. The algorithm was implemented as a deepest disjunctive cut generator for integer programming problems and numerical results were obtained for a selection of test problems from MIPLIB and ORLIB.

**Keywords:** Optimization, Projection, Separation, Nearest point, Column generation

---

**Resumo** Neste artigo propomos uma generalização do algoritmo de Ph. Wolfe (Wolfe, 1976) para obter o ponto pertencente a um polítopo que está mais próximo da origem, de acordo com qualquer métrica. O algoritmo que propomos é composto por um ciclo externo (Ciclo maior) e um ciclo interno (Ciclo menor), termina ao fim de um número finito de iterações e gera a mesma sucessão de pontos que o algoritmo de Ph. Wolfe quando a métrica é a Euclidiana. O algoritmo foi implementado como gerador de cortes disjuntivos mais separadores em problemas de programação inteira, e foram obtidos resultados numéricos para uma selecção de problemas teste da MIPLIB e da ORLIB.

**Keywords:** Optimização, Projecção, Separação, Ponto mais próximo, Geração de colunas

---

## 1 Introdução

Conhecido um ponto  $\bar{x}$  de  $\mathbb{R}^n$  e um conjunto finito de polítopos  $K^i \subset \mathbb{R}^n$ ,  $i \in I$ , consideramos o problema de obter o hiperplano *mais separador* entre  $\bar{x}$  e o polítopo  $K$  definido implicitamente através de

$$K \equiv \text{conv} \left( \bigcup_{i \in I} K^i \right), \quad (1)$$

onde  $\text{conv}(\cdot)$  denota invólucro convexo. O hiperplano *mais separador* entre  $\bar{x}$  e  $K$  é caracterizado pela solução dual óptima no seguinte par de problemas

$$\left. \begin{array}{l} \min \|x - \bar{x}\| \\ \text{s.a } x \in K \end{array} \right\} = \left\{ \begin{array}{l} \max \beta - \alpha \bar{x} \\ \text{s.a } (\alpha, \beta) \in \text{polar}(K), \|\alpha\|_* \leq 1, \end{array} \right. \quad (2)$$

---

\* Research partly supported by Centro de Matemática da Universidade de Coimbra and Fundação para a Ciência e Tecnologia (Projecto POCTI/MAT/14243/1998).

onde  $\|\cdot\|$  e  $\|\cdot\|_*$  designam normas vectoriais mutuamente conjugadas, e o conjunto polar( $K$ )  $\subseteq \mathbf{R}^{n+1}$  designa o conjunto dos  $(n+1)$ -uplos  $(\alpha, \beta)$  tais que “ $\alpha x \geq \beta$ ” é uma desigualdade válida para  $K$ . A igualdade (2) traduz a dualidade entre determinar a projecção de  $\bar{x}$  sobre  $K$  (que vamos designar por problema primal) e determinar o hiperplano mais separador entre  $\bar{x}$  e  $K$  (o problema dual).

O interesse na resolução desta classe de problemas de optimização é motivado pela geração de hiperplanos mais separadores em programação inteira. Por exemplo, quando na resolução de problemas de programação inteira (ou inteira mista)  $\bar{x}$  é minimizante de uma função linear  $cx$  sobre uma relaxação  $\bar{K}$  de um conjunto  $K$ , e interessa obter uma desigualdade válida para  $K$  e inválida em  $\bar{x}$ . Em geral, o problema de minimização (2) é tão difícil quanto o próprio problema de programação inteira que se pretende resolver. Por isso, na prática, a separação deve ocorrer entre o ponto  $\bar{x}$  e uma outra relaxação de  $K$  definida com base em argumentos disjuntivos (por exemplo,  $x_j = 0$  ou  $x_j = 1$ ) que não contenha  $\bar{x}$ .

Neste contexto, duas questões científicas decorrem naturalmente. Que tipo e número de disjunções se devem considerar em (2)? Um maior número de disjunções pode favorecer a obtenção de cortes mais profundos. Contudo, a caracterização algébrica explícita de  $K$  pode ser impeditiva em termos de espaço de memória e/ou de velocidade de convergência. Que norma vectorial se deve considerar em (2)? Existe alguma evidência teórica, ainda não observável na prática, de que normas como a Euclidiana, por exemplo, poderão proporcionar convergência mais rápida ao algoritmo de planos cortantes, veja-se (Soares, 1998), por exemplo. Mais, a resolução de (2) pode obrigar à resolução de problemas não lineares e/ou não diferenciáveis, quadráticos no caso da norma Euclidiana, e para os quais não é óbvia a escolha dos melhores algoritmos que aproveitem a estrutura específica do problema em causa. Por outro lado, tem-se observado experimentalmente que a norma escolhida em (2) influencia a esparsidade do corte assim caracterizado, o que também parece ser um aspecto relevante na eficácia do algoritmo de planos cortantes. Nesse aspecto, os cortes mais separadores baseados na norma Euclidiana tendem ser menos esparsos do que cortes baseados em normas poliedrais como as normas  $l_1$  e  $l_\infty$ .

A dualidade (2) sugere a utilização de algoritmos que, com o propósito de determinar um hiperplano separador, possam utilizar de forma eficiente a estrutura combinatorial de  $K$ . Por exemplo, quando a norma é a Euclidiana, o algoritmo proposto em (Wolfe, 1976) resolve (2) através da resolução sucessiva de problemas lineares sobre  $K$ . Sendo  $K$  o invólucro convexo de conjuntos polidricos, a optimização de uma função linear sobre  $K$  pode ser massivamente paralelizável ou eficientemente resolvida por algoritmos do tipo paramétrico.

Neste artigo, propomos uma generalização do algoritmo de Ph. Wolfe para a resolução do problema (2) com qualquer norma, diferenciável ou não, motivados pelo interesse dessa abordagem na geração de planos cortantes baseados em argumentos disjuntivos. Resultados computacionais obtidos com uma implementação combinada de Matlab e Cplex são também apresentados.

Na próxima secção, é apresentada a notação usual, e recordamos alguns conceitos e resultados básicos da teoria polidral. Os conceitos topológicos que usamos são os presentes na literatura da análise convexa, veja-se, por exemplo, (Hiriart-Urruty and Lemaréchal, 1993a,b). Na Secção 3, propomos uma generalização do algoritmo de Ph. Wolfe para a resolução do problema de obter o hiperplano mais separador entre um ponto conhecido e o invólucro convexo de conjuntos polidrais definido com base em argumentos disjuntivos, com qualquer norma, diferenciável ou não. Apresentamos uma prova da sua convergência finita e analisamos a álgebra linear numérica, inerente à implementação do algoritmo proposto, distintamente para as normas  $l_2$ ,  $l_1$  e  $l_\infty$ . Na Secção 4, apresentamos alguns resultados computacionais da implementação do algoritmo proposto para a resolução do problema de determinar o ponto pertencente ao invólucro convexo de um conjunto finito de pontos, com diferentes normas -  $l_1$ ,  $l_2$  e  $l_\infty$ , e, como gerador de cortes, para a resolução de problemas de optimização discreta com diversos tipos de disjunções e a norma Euclideana.

## 2 Fundamentos

Seja  $Q = \{q_1, q_2, \dots, q_m\}$  um conjunto finito de pontos de  $\mathbf{R}^n$  ou uma matriz com aquelas colunas, devendo a distinção ser inferida do contexto. O conjunto  $Q$  diz-se *independente afim* se, para todo  $q \in Q$ ,  $q \notin \text{aff}(Q \setminus \{q\})$ , com  $\text{aff}(\cdot)$  a denotar invólucro afim. Se  $Q$  é independente afim, então todo o  $x \in \text{aff}(Q)$  possui uma representação única  $x = Qw$  com  $w$ , designado por vector das coordenadas baricênticas de  $x$  relativamente a  $Q$ , um vector tal que  $ew \equiv \sum_{i=1}^m w_i = 1$ . O invólucro convexo de  $Q$ , denotado por  $\text{conv}(Q)$ , é o conjunto dos pontos  $x$  tais que, para algum  $w \geq 0$ , se tenha  $x = Qw$  e  $ew = 1$ . Sendo  $Q$  independente afim, o interior relativo de  $\text{conv}(Q)$ , denotado por  $\text{ri}(\text{conv}(Q))$ , é o conjunto dos pontos  $x$  tais que, para algum  $w > 0$ , se tenha  $x = Qw$  e  $ew = 1$ .

Um poliedro  $K$  diz-se um *polítopo* se existe um conjunto finito  $Q$  tal que  $K = \text{conv}(Q)$ . Se  $K = \text{conv}(Q)$ , então  $Q$  é minimal se e só se  $Q$  é o conjunto dos pontos extremos de  $K$ . A dimensão de um poliedro  $K$ , denotada por  $\text{dim}(K)$ , é a cardinalidade do maior conjunto independente afim contido em  $K$  menos um. O conjunto polar( $K$ ) é o conjunto de pontos  $(\alpha, \beta) \in \mathbf{R}^{n+1}$  que definem desigualdades válidas do tipo " $\alpha x \geq \beta$ " para o poliedro  $K$ . Uma face de  $K$  é um poliedro  $F$  tal que  $F = \{x \in K : \alpha x = \beta\}$  para algum  $(\alpha, \beta) \in \text{polar}(K)$ . Uma faceta de  $K$  é uma face de dimensão  $\text{dim}(K) - 1$ . Um ponto extremo ou vértice de  $K$  é uma face de dimensão zero. Se  $K = \text{conv}(Q)$  então, para qualquer subconjunto independente afim  $S \subseteq Q$ , o conjunto  $\text{conv}(S)$  é uma face de  $K$  de dimensão  $|S| - 1$ .

Seja  $f$  uma função convexa de  $\mathbf{R}^n$  em  $\mathbf{R}$ . O subdiferencial de  $f$  em  $x \in \mathbf{R}^n$ , denota-se  $\partial f(x)$ , é o conjunto de todos os subgradientes de  $f$  em  $x$  (são vectores linha!). Se, adicionalmente,  $f$  é contínua em  $x$ , então o conjunto  $\partial f(x)$  é não vazio e compacto; se  $f$  é diferenciável em  $x$ , então  $\partial f(x)$  possui um único elemento denominado o gradiente de  $f$  em  $x$  e denota-se por  $\nabla f(x)$ .

Seja  $\|\cdot\|$  uma qualquer norma em  $\mathbf{R}^n$  e  $\bar{x} \notin K = \text{conv}(Q)$ , com  $Q$  finito. Então, existe pelo menos um ponto  $\hat{x} \in \text{conv}(Q)$  que está à distância mínima de  $\bar{x}$ . A unicidade só é garantida se a norma for continuamente diferenciável. Qualquer um daqueles pontos  $\hat{x}$  é caracterizado por: para  $f(x) = \|x - \bar{x}\|$ , existe um subgradiente  $\alpha \in \partial f(\hat{x})$  e um escalar  $\beta = f(\hat{x})$ , tal que  $\alpha x \leq \beta$  para todo o  $x \in K$ . Recorde-se que  $\partial f(x) = \{\xi \in \mathbf{R}^n : \xi(x - \bar{x}) = \|x - \bar{x}\|, \|\xi\|_* \leq 1\}$  é um conjunto que, na maioria dos casos de interesse ( $l_1, l_\infty, l_p$ , Frobenius, etc.), é facilmente caracterizável em termos algébricos. A norma  $\|\cdot\|_*$  é denotada por norma conjugada (ou dual) da norma  $\|\cdot\|$  e é definida por  $\|\alpha\|_* \equiv \max\{\alpha x : \|x\| \leq 1\}$ . Exemplos de normas vectoriais em  $\mathbf{R}^n$  são: a norma  $l_2$  ou Euclidiana,  $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ , a norma  $l_1$ ,  $\|x\|_1 = \sum_{i=1}^n |x_i|$ , e a norma  $l_\infty$ ,  $\|x\|_\infty = \max_{i=1, \dots, n} |x_i|$ . A norma  $l_2$  é dual de si própria, a norma  $l_1$  é dual da norma  $l_\infty$  e *vice-versa*.

## 3 Uma generalização do algoritmo de Ph. Wolfe

Ph. Wolfe propôs, em (Wolfe, 1976), um algoritmo para determinar o ponto pertencente ao invólucro convexo de um conjunto finito de pontos  $K^0 \equiv \{p_1, p_2, \dots, p_m\} \subset \mathbf{R}^n$ , aqui denotado por  $K \equiv \text{conv}(K^0)$ , que está mais próximo da origem no sentido da norma Euclidiana. Algebricamente, o algoritmo resolve o seguinte problema de optimização, onde  $\|\cdot\|$  denota norma Euclidiana,

$$\begin{aligned} & \min \|x\| \\ \text{s.a. } & x = \sum_{k=1}^m w_k p_k, \quad \sum_{k=1}^m w_k = 1, \quad w_k \geq 0, \quad k = 1, 2, \dots, m. \end{aligned} \quad (3)$$

O algoritmo de Ph. Wolfe não requer o conhecimento explícito e antecipado de  $K^0$ , antes manipulando subconjuntos de  $K^0$  gerados iterativamente - neste aspecto o algoritmo pode ser considerado um algoritmo de geração de colunas. A cardinalidade desses subconjuntos nunca ultrapassa  $\text{dim}(K) + 1$ , o que torna a abordagem atractiva quando  $|K^0|$  é elevado.

Propomos, na Figura 1, uma generalização do algoritmo de Ph. Wolfe para a resolução do problema (3) com qualquer norma. Tal como aquele, o algoritmo que propomos é composto por um ciclo externo (Ciclo maior) e um ciclo interno (Ciclo menor) com invariantes semelhantes, terminando ao fim de um número finito de passos e gerando a mesma sucessão de pontos quando a norma é a Euclidiana.

Cada iteração do Ciclo maior inicia-se com o conhecimento de um par de subconjuntos  $P_1, P_2 \subseteq K^0$ , com  $P_1$  independente afim, e um ponto  $X \in \text{ri conv}(P_1)$  tal que  $X \in \arg \min\{\|x\|: x \in \text{aff}(P_1)\}$ . Como veremos, o subconjunto  $P_2$  é necessário para lidar com a averiguação de optimalidade. Se  $X$  verifica a condição necessária e suficiente de optimalidade sobre  $K$  (conforme Secção 2) com um dado subgradiente, então o algoritmo termina. Caso contrário, é identificado um novo ponto  $\hat{p} \in K^0$  que deve ser acrescentado a  $P \equiv P_1 \cup P_2$ . Quando a norma é não diferenciável, existe a dificuldade adicional da identificação do subgradiente que valida a optimalidade de  $X$ .

Se a inserção de  $\hat{p}$  em  $P$  permite identificar uma direcção de descida, então  $\hat{p}$  é inserido em  $P_1$  (possivelmente com outros pontos de  $P_2$ ) e  $P_2$  fica vazio. Prossegue a execução do Ciclo menor que consiste na eliminação sucessiva de pontos de  $P_1$  até que seja alcançado  $X \in \text{ri conv}(P_1)$  tal que  $X \in \arg \min\{\|x\|: x \in \text{aff}(P_1)\}$ . Se a inserção de  $\hat{p}$  não favorece a identificação de uma direcção de descida, então  $\hat{p}$  é inserido em  $P_2$ , e é imediatamente iniciada uma nova iteração do Ciclo maior sem qualquer redefinição de  $X$ , mas com um novo subgradiente. Esta última situação, que favorecerá o aparecimento do tal subgradiente que assegura a optimalidade de  $X$  ao fim de um número finito de passos, só acontece quando a norma não é continuamente diferenciável.

Designamos uma dada iteração do Ciclo maior por *séria* se  $\|X\|$  decresce nessa iteração, caso contrário designamos essa iteração por *nula*. Esta notação também é adoptada na descrição de alguns algoritmos para optimização convexa não diferenciável, descritos em (Hiriart-Urruty and Lemaréchal, 1993b), por exemplo. Iterações nulas de um Ciclo maior são constituídas por nenhuma execução do Ciclo menor, enquanto que iterações sérias são constituídas por pelo menos uma execução do Ciclo menor.

A próxima sequência de lemas destina-se a demonstrar a convergência finita do algoritmo proposto. É consequência imediata de resultados clássicos que  $X \in \arg \min\{\|x\|: x \in \text{aff}(P_1)\}$  se e só se existe  $\alpha \in \partial\|\cdot\|(X)$  tal que  $\alpha X = \alpha p$  para todo  $p \in P_1$ . Em particular, se  $P_1 = \{X\}$  então isso é verdade com qualquer  $\alpha \in \partial\|\cdot\|(X)$ . O próximo lema assegura a veracidade, para algum  $\alpha$ , da afirmação entre parêntesis no início de cada uma das iterações do Ciclo maior.

**Lema 1** *Seja  $\hat{x} = \sum_{p^i \in P} \hat{w}_i p^i$  admissível para (4). Então,  $\hat{x}$  é solução óptima para (4) se e só se existe  $\hat{\alpha} \in \partial\|\cdot\|(\hat{x})$  tal que*

$$\begin{aligned} \hat{\alpha}(p - \hat{x}) &= 0 \quad \text{para todo } p \in P_1 \cup \{p^i \in P_2: \hat{w}_i > 0\}, \\ \hat{\alpha}(p - \hat{x}) &\geq 0 \quad \text{para todo } p \in \{p^i \in P_2: \hat{w}_i = 0\}. \end{aligned}$$

*Mais,*

1. se  $\|\hat{x}\| < \|X\|$ , então  $\hat{w}_0 > 0$  para  $p^0 = \hat{p} \in P_2$ ;
2. se  $\|\hat{x}\| = \|X\|$ , então  $\hat{\alpha} \in \partial\|\cdot\|(X)$ .

*Demonstração.* Seja  $C \subseteq \mathbf{R}^n$  o conjunto dos pontos  $x$  admissíveis para (4). Sendo  $C$  um conjunto convexo fechado,  $\hat{x}$  é óptimo para  $\min\{\|x\|: x \in C\}$  se e só se  $0 \in \partial\|\cdot\|(\hat{x}) + N_C(\hat{x})$ , com  $N_C(\hat{x})$  a denotar o cone normal de  $C$  em  $\hat{x}$ , veja-se, por exemplo, (Hiriart-Urruty and Lemaréchal, 1993a). Como

$$N_C(\hat{x}) = \left\{ s \in \mathbf{R}^n: \begin{array}{l} s(p - \hat{x}) = 0, p \in P_1 \cup \{p^i \in P_2: \hat{w}_i > 0\}, \\ s(p - \hat{x}) \leq 0, p \in \{p^i \in P_2: \hat{w}_i = 0\} \end{array} \right\}$$

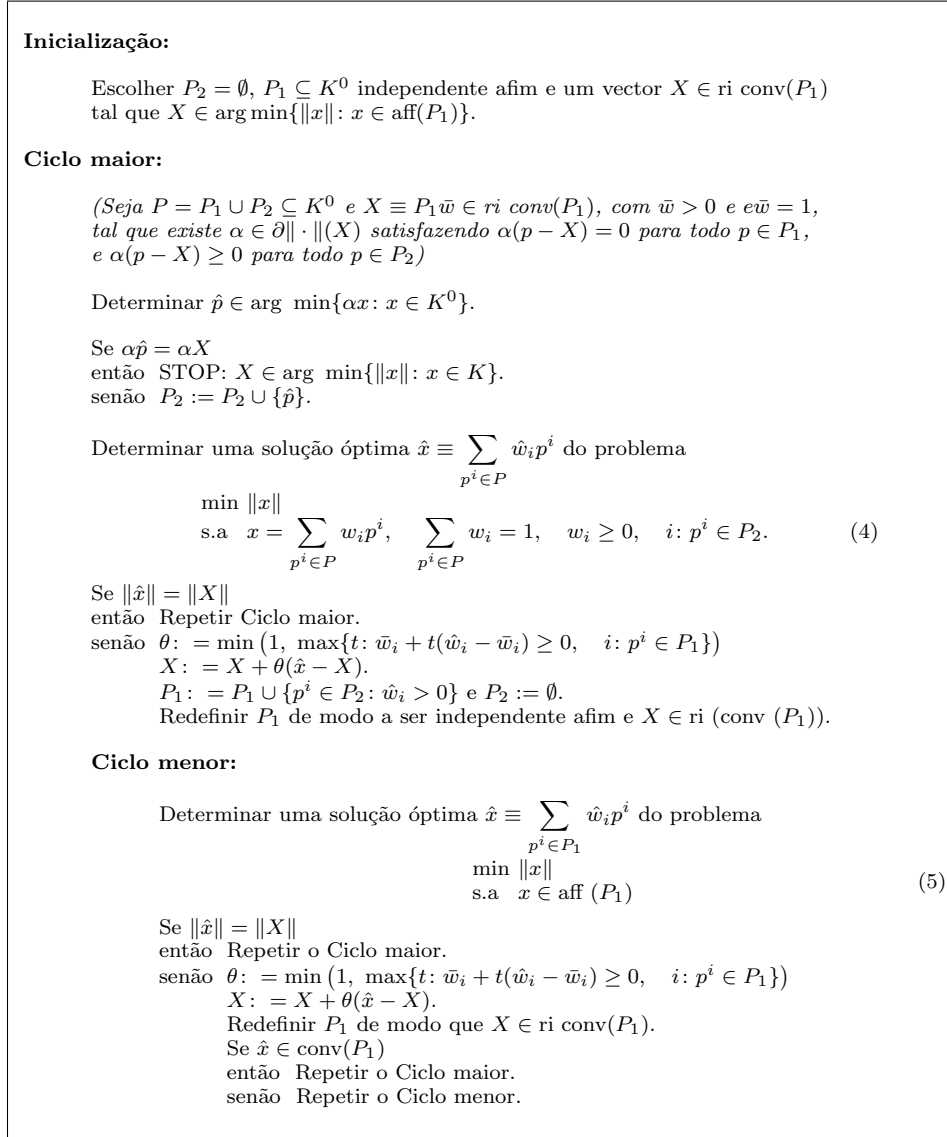
concluimos a primeira parte do lema. Agora, suponhamos que  $\|\hat{x}\| < \|X\|$ . Como  $\alpha \in \partial\|\cdot\|(X)$  e  $X$  é admissível para (4),

$$\|\hat{x}\| \geq \|X\| + \alpha(\hat{x} - X) > \|\hat{x}\| + \alpha(\hat{x} - X), \quad (6)$$

pelo que  $\alpha(\hat{x} - X) < 0$ . Por outro lado,

$$\alpha(\hat{x} - X) = \sum_{p^i \in P_1} \hat{w}_i \alpha(p^i - X) + \sum_{p^i \in P_2 \setminus \{\hat{p}\}} \hat{w}_i \alpha(p^i - X) + \hat{w}_0 \alpha(\hat{p} - X) \geq \hat{w}_0 \alpha(\hat{p} - X),$$

devido à propriedade de  $\alpha$  no início do Ciclo maior. Por isso, também  $\hat{w}_0 \alpha(\hat{p} - X) < 0$ . Como  $\alpha \hat{p} < \alpha X$ , por definição de  $\hat{p}$ , então  $\hat{w}_0 > 0$ . Finalmente, se  $\|\hat{x}\| = \|X\|$ , então  $\hat{\alpha} \in \partial\|\cdot\|(X)$  é consequência de (Burke and Ferris, 1991, Lema 2(a)), o que conclui a demonstração.



**Figura 1.** Algoritmo modificado de Ph. Wolfe.

No caso de  $f(x) \equiv \|x\|$  ser uma norma continuamente diferenciável,  $\alpha = \nabla f(x)$  e, portanto,  $\alpha(\hat{p} - X) < 0$  implica  $\|\hat{x}\| < \|X\|$ , pelo que não existirão iterações nulas. Em particular, no problema (4),  $P_2 = \{\hat{p}\}$  sempre. Como, pelo Lema 1,  $\hat{w}_0 > 0$ , então o problema (4) é essencialmente um problema sem restrições e, portanto, equivalente a um adequado sistema de equações lineares que, como é possível demonstrar, tem solução única.

No caso de  $f(x) \equiv \|x\|$  ser uma norma  $l_1$  ou  $l_\infty$ , o problema (4) é equivalente a um adequado problema linear. Mais adiante, explicaremos como obter  $\hat{\alpha}$  a partir de informação dual básica. Por 2. do Lema 1, caso  $\|\hat{x}\| = \|X\|$ , este vector  $\hat{\alpha}$  servirá para iniciar uma nova iteração do Ciclo maior.

**Lema 2** *Se, após a resolução de (4),  $\|\hat{x}\| \equiv \sum_{p^i \in P} \hat{w}_i p^i < \|X\| \equiv \sum_{p^i \in P_1} \bar{w}_i p^i$ , então:*

1.  $\theta \equiv \min(1, \max\{t: \bar{w}_i + t(\hat{w}_i - \bar{w}_i) \geq 0 \text{ para todo } i: p^i \in P_1\}) \in (0, 1]$ ;
2.  $X + \theta(\hat{x} - X) \in \text{conv}(P'_1 \equiv P_1 \cup \{p^i \in P_2: \hat{w}_i > 0\})$ , e
3.  $\|X + \theta(\hat{x} - X)\| < \|X\|$ .

*Demonstração.* O facto de  $\theta \in (0, 1]$  decorre de  $\bar{w}_i > 0$  para todo  $i: p^i \in P_1$ . Portanto, para todo o escalar  $t \geq 0$ ,

$$X + t(\hat{x} - X) = \sum_{p^i \in P_1} (\bar{w}_i + t(\hat{w}_i - \bar{w}_i)) p^i + \sum_{p^i \in P_2} (t\hat{w}_i) p^i.$$

Como  $\hat{w}_i \geq 0$  para  $i: p^i \in P_2$ , e  $\bar{w}_i > 0$  para  $i: p^i \in P_1$ , tem-se  $X + t(\hat{x} - X) \in \text{conv}(P'_1)$  para todo  $t \in [0, \theta]$ . Finalmente,  $\|X + \theta(\hat{x} - X)\| = \|(1 - \theta)X + \theta\hat{x}\| \leq (1 - \theta)\|X\| + \theta\|\hat{x}\| < (1 - \theta)\|X\| + \theta\|X\| = \|X\|$ .

Após a resolução de (4) e no caso de  $\|\hat{x}\| < \|X\|$ , ocorre uma redefinição de  $X$  e de  $P_1$ , de modo a que a definição de  $\theta$  assegure  $X \in \text{conv}(P_1)$ , e, atendendo a 1. do Lema 1, pelo menos um novo vector é inserido em  $P_1$ . O próximo resultado diz respeito à independência afim de  $P_1$ .

**Lema 3** *Suponhamos que  $\hat{x} = \sum_{p^i \in P} \hat{w}_i p^i$  para algum  $\hat{w}$  tal que  $\hat{w}_i \geq 0$  para todo  $i: p^i \in P_2$ , e que  $P_1$  é independente afim. Se  $\hat{w}$  é ponto extremo do poliedro*

$$W = \{w: \hat{x} = Pw, w_i \geq 0, \quad i: p^i \in P_2\},$$

*então  $P_1 \cup \{p^i \in P_2: \hat{w}_i > 0\}$  é independente afim.*

*Demonstração.* Suponhamos que  $P_1 \cup \{p^i \in P_2: \hat{w}_i > 0\}$  não é independente afim. Então, existe  $\bar{w} \neq 0$  tal que  $\bar{w}_i = 0$  para todo  $i: p^i \in P_2$  e  $\hat{w}_i = 0$ , e

$$\sum_{p^i \in P} \bar{w}_i p^i = 0, \quad \sum_{p^i \in P} \bar{w}_i = 0.$$

Para  $t > 0$  e suficientemente pequeno,  $\hat{w} + t\bar{w}$  e  $\hat{w} - t\bar{w}$  pertencem a  $W$ . Chegamos a um absurdo pois  $\hat{w}$  é combinação convexa desses dois pontos de  $W$ .

Portanto, se a solução óptima de (4) estiver caracterizada adequadamente, então o conjunto  $P_1$  mantém-se independente afim, sem que isso seja imposto artificialmente. Note-se que, em geral, se  $X \in \text{conv}(P_1)$ , então um procedimento análogo ao que está implícito na demonstração do Lema 3 permite encontrar  $P'_1 \subseteq P_1$  independente afim e tal que  $X \in \text{ri conv}(P'_1)$ .

Na primeira iteração de uma passagem pelo Ciclo menor, a solução óptima  $\hat{x}$  do problema (4) permanece óptima em (5) nessa iteração. A demonstração do próximo lema é análoga à do Lema 2.

**Lema 4** *Se, após a resolução de (5),  $\|\hat{x}\| < \|X\|$  e  $\hat{x} \notin \text{conv}(P_1)$ , então*

$$\theta \equiv \min(1, \max\{t: X + t(\hat{x} - X) \in \text{conv}(P_1)\}) \in (0, 1), \quad \|X + \theta(\hat{x} - X)\| < \|X\|,$$

*e  $P_1$  fica com menos um elemento, pelo menos.*

Estamos agora em condições de demonstrar a convergência finita do algoritmo proposto.

**Teorema 1** *O algoritmo da Figura 1 termina num número finito de iterações dos Ciclos maior e menor.*

*Demonstração.* Pelo Lema 4, em cada iteração do Ciclo maior, ocorre um número finito de iterações do Ciclo menor. Em cada iteração (com a possível excepção da primeira e da última), o conjunto  $P$  fica com menos um elemento. Por isso, basta provar que o algoritmo termina num número finito de iterações do Ciclo maior.

Primeiro, provamos que existe um número finito de iterações nulas consecutivas. Entre duas iterações nulas consecutivas,  $P_1$  mantém-se constante, e  $P_2$  tem mais um elemento. Como  $K^0$  é finito, esta sequência não pode decorrer indefinidamente. Segundo, na iteração seguinte a uma iteração séria,  $\|X\| = \min\{\|x\|: x \in \text{aff}(P_1)\}$  e  $X \in \text{ri conv}(P_1)$ . Entre duas iterações sérias consecutivas,  $\|X\|$  decresce estritamente. Portanto, de iteração séria em iteração séria, os conjuntos  $P_1$  não podem repetir-se. Como existem em número finito, concluímos que existe um número finito de iterações sérias e também de todo o algoritmo.

### 3.1 Álgebra linear numérica

Nesta secção, explica-se como se devem organizar os cálculos algébricos envolvidos na implementação do algoritmo descrito na Figura 1 na resolução do problema de minimização em (2). Duas características desse problema não são capturadas em (3). Nomeadamente, a função objetivo é  $\|x - \bar{x}\|$  e  $K^0$ , o conjunto dos pontos extremos de  $K$ , não é conhecido explicitamente. Mas como

$$\begin{aligned} \min \|x - \bar{x}\| &= \min \|y\| \\ \text{s.a } x \in K &= \text{s.a } y \in K - \bar{x} \end{aligned} \quad (7)$$

e  $(K - \bar{x})^0 = K^0 - \bar{x}$ , então o algoritmo proposto também resolve o problema de minimização em (2) com devidos ajustamentos.

Começamos por analisar a primeira escolha para o conjunto  $P_1$  que dá início ao algoritmo. Em geral, uma escolha possível é  $P_1 := \{p - \bar{x}\}$ , com  $p$  um qualquer ponto extremo de  $K$ . No entanto, quando já é conhecido um conjunto  $Q_1 = \{q^i\} \subseteq K^0$  proveniente de, por exemplo, uma prévia resolução de (7) com um  $\bar{x}$  diferente, podemos inicializar  $P_1$  de modo a acelerar o algoritmo. Primeiro, obtemos o vector  $X = \sum_i w_i q^i \in \arg \min\{\|x - \bar{x}\| : x \in \text{conv}(Q_1)\}$ , usando um algoritmo adequado, após o que se define  $P_1 = \{q^i \in Q_1 : w_i > 0\} - \bar{x}$  de modo a que  $P_1$  seja independente afim e  $X - \bar{x} \in \text{ri conv}(P_1)$ .

Outro aspecto a ter em conta na implementação do algoritmo proposto é a determinação do novo ponto  $\hat{p} \in K^0 - \bar{x}$ . Se  $K$  é da forma (1), então

$$\begin{aligned} \min \alpha y \\ \text{s.a } y \in K^0 - \bar{x} \end{aligned} = \min_{i \in I} \left( \begin{array}{l} \min \alpha x \\ \text{s.a } x \in K^i \end{array} \right) - \alpha \bar{x}, \quad (8)$$

pelo que  $\hat{p} = \hat{q}^k - \bar{x}$ , com  $\hat{q}^k$  ponto extremo de um dos polítopos  $K^k, k \in I$ . Nas próximas subsecções, analisamos a álgebra linear numérica, inerente à implementação do algoritmo proposto, distintamente para as normas  $l_2, l_1$  e  $l_\infty$ .

### 3.2 Norma Euclidiana

Dado um conjunto  $Q_1 \subseteq K^0$  tal que  $|Q_1|$  não é elevado, o problema  $\min\{\|x - \bar{x}\| : x \in \text{conv}(Q_1)\}$  pode ser resolvido pelo algoritmo NNLS proposto em (Lawson and Hanson, 1995) para a resolução de problemas de mínimos quadrados com restrições de não negatividade. O algoritmo NNLS está disponível como função interna do Matlab e também está disponível em FORTRAN em <http://www.netlib.org/lawson-hanson/all>. O algoritmo NNLS é um método de restrições activas baseado na decomposição  $QR$ .

O Lema 5 abaixo mostra que (9) abaixo está relacionado com o problema de mínimos quadrados com restrições de não negatividade (10) e justifica a utilização da rotina NNLS,

$$\begin{aligned} \min \|x - \bar{x}\|^2 &= \min \|Q_1 v - \bar{x}\|^2 = \min \|(Q_1 - \bar{x}e)v\|^2 \\ \text{s.a } x \in \text{conv}(Q_1) & \quad \text{s.a } ev = 1, v \geq 0 \quad \text{s.a } ev = 1, v \geq 0, \end{aligned} \quad (9)$$

$$\begin{aligned} \min \left\| \begin{bmatrix} Q_1 - \bar{x}e \\ e \end{bmatrix} u - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\|^2 &= \min \|(Q_1 - \bar{x}e)u\|^2 + (eu - 1)^2 \\ \text{s.a } u \geq 0 & \quad \text{s.a } u \geq 0, \end{aligned} \quad (10)$$

em que a matriz  $Q_1 - \bar{x}e$  é obtida subtraindo  $\bar{x}$  a cada coluna da matriz  $Q_1$ . Resultados similares podem ser deduzidos na ausência, total ou parcial, de restrições de não-negatividade. Esse lema é mencionado sem demonstração em (Wolfe, 1976). Incluímos aqui uma possível demonstração.

**Lema 5** *Se  $\bar{u}$  é solução óptima de (10), então o valor óptimo é  $1 - e\bar{u}$ ,  $\bar{v} = \bar{u}/(e\bar{u})$  é solução óptima de (9), e  $(1 - e\bar{u})/(e\bar{u})$  o respectivo valor óptimo.*

*Demonstração.* Das condições necessárias e suficientes de optimalidade de (10), e usando  $Q \equiv Q_1 - \bar{x}e$ , tem-se

$$\bar{u} \geq 0, \quad (Q^T Q + e^T e) \bar{u} - e^T \geq 0, \quad \bar{u}^T [(Q^T Q + e^T e) \bar{u} - e^T] = 0.$$

Então,  $\bar{u}^T Q^T Q \bar{u} + (e\bar{u})^2 = e\bar{u}$  ou, de modo equivalente,  $\|Q\bar{u}\|^2 + (1 - e\bar{u})^2 = 1 - e\bar{u}$ . Note-se que  $e\bar{u} > 0$ , porque se  $e\bar{u} = 0$ , então  $\bar{u} = 0$ , o que é absurdo. Observe-se que  $\bar{v} \equiv \bar{u}/(e\bar{u}) \geq 0$  e  $e\bar{v} = 1$ . Se  $\lambda = (1 - e\bar{u})/(e\bar{u})$ , então  $Q^T Q \bar{v} + \lambda e^T = [(Q^T Q + e^T e) \bar{u} - e^T]/(e\bar{u}) \geq 0$  e

$$\bar{v}^T (Q^T Q \bar{v} + \lambda e^T) = \frac{1}{(e\bar{u})^2} \bar{u}^T [(Q^T Q + e^T e) \bar{u} - e^T] = \frac{1}{(e\bar{u})^2} [\bar{u}^T (Q^T Q + e^T e) \bar{u} - e\bar{u}] = 0.$$

Por isso,  $\bar{v}$  é óptimo para (9), e o valor óptimo é

$$\|Q\bar{v}\|^2 = \frac{1}{(e\bar{u})^2} \|Q\bar{u}\|^2 = \frac{e\bar{u}(1 - e\bar{u})}{(e\bar{u})^2} = \frac{1 - e\bar{u}}{e\bar{u}}.$$

No final do algoritmo NNLS, as componentes positivas de  $\bar{u}$  estão associadas a um subconjunto  $P_1 - \bar{x}e$  de  $Q_1 - \bar{x}e$  que é independente afim. Então, como  $P_1$  também é independente afim e  $X = Q_1 \bar{v} \in \text{ri conv}(P_1)$ ,  $X$  é única solução óptima de problema de optimização

$$\begin{aligned} \min \|x - \bar{x}\|^2 &= \min \|P_1 v - \bar{x}\|^2 \\ \text{s.a } x \in \text{aff}(P_1) &\quad \text{s.a } ev = 1 \end{aligned} \quad (11)$$

que, por sua vez, está relacionado com o problema de mínimos quadrados sem restrições

$$\min \left\| \begin{bmatrix} P_1 - \bar{x}e \\ e \end{bmatrix} u - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\|^2 = \min \|(P_1 - \bar{x}e)u\|_2 + (eu - 1)^2, \quad (12)$$

conforme está implícito no Lema 5.

Seguidamente, explicamos a álgebra linear numérica inerente à introdução de novas colunas e remoção de colunas de  $P_1$ . Se  $P_1$  é um conjunto independente afim, então existem uma matriz  $(n+1) \times (n+1)$  ortogonal  $Q$ , uma matriz  $m \times m$  triangular superior e não singular  $R$  e um escalar não nulo  $\bar{b}_2$  tais que

$$\tilde{P}_1 \equiv \begin{bmatrix} P_1 - \bar{x}e & 0 \\ e & 1 \end{bmatrix} = Q \begin{bmatrix} R \bar{b}_1 \\ 0 \bar{b}_2 \\ 0 & 0 \end{bmatrix}. \quad (13)$$

Como é simples de verificar, a solução óptima do problema (12) é a solução  $\bar{u}$  do sistema de equações lineares  $Ru = \bar{b}_1$ , e o respectivo valor óptimo é  $\bar{b}_2^2 = 1 - e\bar{u}$ . Consequentemente, por um resultado análogo ao do Lema 5,  $\bar{v} = \bar{u}/(e\bar{u})$  é solução óptima de (11). Em seguida, explica-se como actualizar a matriz triangular superior em (13) após a inserção e remoção de um elemento em  $P_1$ . Consideremos a inserção de um elemento ao conjunto  $P_1$ .

**Lema 6** *Seja  $P_1 \cup \{\hat{p}\}$  um conjunto independente afim, e suponha-se conhecida a decomposição (13) de  $\tilde{P}_1$ . Então, existe uma matriz  $(n+1) \times (n+1)$  ortogonal  $\bar{Q}$  tal que*

$$\bar{P}_1 \equiv \begin{bmatrix} P_1 - \bar{x}e & \hat{p} - \bar{x} & 0 \\ e & 1 & 1 \end{bmatrix} = \bar{Q} \begin{bmatrix} R a \bar{b}_1 \\ 0 \rho f \\ 0 0 g \\ 0 0 0 \end{bmatrix}, \quad (14)$$

onde, para  $s = (P_1 - \bar{x}e)^T(\hat{p} - \bar{x}) + e^T$ ,

$$\begin{aligned} a &= R^{-T} s, & \rho &= \sqrt{(\hat{p} - \bar{x})^T(\hat{p} - \bar{x}) + 1 - a^T a}, \\ f &= (1 - a^T \bar{b}_1)/\rho, & g &= \sqrt{\bar{b}_2^2 - f^2}. \end{aligned}$$



*Demonstração.* Primeiro, observe-se que

$$\tilde{P}_1^T \tilde{P}_1 = \begin{bmatrix} (P_1 - \bar{x}e)^T(P_1 - \bar{x}e) + e^T e & e^T \\ e & 1 \end{bmatrix} = \begin{bmatrix} R & \bar{b}_1 \\ 0 & \bar{b}_2 \end{bmatrix}^T \begin{bmatrix} R & \bar{b}_1 \\ 0 & \bar{b}_2 \end{bmatrix} = \begin{bmatrix} R^T R & R^T \bar{b}_1 \\ \bar{b}_1^T R & \bar{b}_1^T \bar{b}_1 + \bar{b}_2^2 \end{bmatrix}.$$

Seja  $\bar{R}$  a matriz  $(m+1) \times (m+1)$  triangular superior em (14). Então,

$$\begin{aligned} \bar{R}^T \bar{R} &= \begin{bmatrix} R & a & \bar{b}_1 \\ 0 & \rho & f \\ 0 & 0 & g \end{bmatrix}^T \begin{bmatrix} R & a & \bar{b}_1 \\ 0 & \rho & f \\ 0 & 0 & g \end{bmatrix} = \begin{bmatrix} R^T R & R^T a & R^T \bar{b}_1 \\ a^T R & a^T a + \rho^2 & a^T \bar{b}_1 + \rho f \\ \bar{b}_1^T R & \bar{b}_1^T a + \rho f & \bar{b}_1^T \bar{b}_1 + f^2 + g^2 \end{bmatrix} \\ &= \begin{bmatrix} (P_1 - \bar{x}e)^T(P_1 - \bar{x}e) + e^T e & (P_1 - \bar{x}e)^T(\hat{p} - \bar{x}) + e^T & e^T \\ (\hat{p} - \bar{x})^T(P_1 - \bar{x}e) + e & (\hat{p} - \bar{x})^T(\hat{p} - \bar{x}) + 1 & 1 \\ e & 1 & 1 \end{bmatrix} = \bar{P}_1^T \bar{P}_1. \end{aligned}$$

Este lema permite caracterizar a solução óptima  $(\hat{u}, \hat{u}_0)$  de

$$\min \left\| \begin{bmatrix} P_1 - \bar{x}e & \hat{p} - \bar{x} \\ e & 1 \end{bmatrix} \begin{bmatrix} u \\ u_0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\|^2 \quad (15)$$

como sendo a solução do sistema de equações lineares

$$\begin{bmatrix} R & a \\ 0 & \rho \end{bmatrix} \begin{bmatrix} u \\ u_0 \end{bmatrix} = \begin{bmatrix} \bar{b}_1 \\ f \end{bmatrix} \iff \begin{bmatrix} \hat{u} \\ \hat{u}_0 \end{bmatrix} = \begin{bmatrix} \bar{u} - (f/\rho)a \\ f/\rho \end{bmatrix}.$$

Agora, consideremos a remoção de uma das primeiras  $m$  colunas de  $P_1$ , denotada por  $p_j$ . Considere-se a matriz

$$\bar{P}_1 \equiv \begin{bmatrix} p_1 - \bar{x} & \dots & p_{j-1} - \bar{x} & p_{j+1} - \bar{x} & \dots & p_m - \bar{x} & 0 \\ 1 & \dots & 1 & 1 & \dots & 1 & 1 \end{bmatrix} = Q \begin{bmatrix} \tilde{R} & \bar{b}_1 \\ 0 & \bar{b}_2 \\ 0 & 0 \end{bmatrix},$$

onde a matriz  $\tilde{R}$  só não é triangular superior porque os elementos nas posições  $(j, j+1), \dots, (m, m+1)$  são não nulos. Esta matriz pode ser conduzida à forma triangular superior, efectuando uma multiplicação à esquerda por  $m-j+1$  matrizes de Givens, que são matrizes ortogonais. Deste modo se obtém a decomposição  $QR$  da matriz  $\bar{P}_1$ .

### 3.3 Normas $l_1$ e $l_\infty$

No caso da norma  $l_1$ , o problema (4) pode ser formulado como um problema linear cujo par primal-dual apresentamos de seguida:

$$\left. \begin{array}{l} \min e\sigma \\ \text{s.a. } Pw + \sigma \geq \bar{x} \\ \quad -Pw + \sigma \geq -\bar{x} \\ ew = 1 \\ w_i \geq 0 \ (i : p^i \in P_2) \end{array} \right\} = \left\{ \begin{array}{l} \max (\gamma_1 - \gamma_2)\bar{x} + \rho \\ \text{s.a. } (\gamma_1 - \gamma_2)P_1 + \rho e = 0 \\ (\gamma_1 - \gamma_2)P_2 + \rho e \leq 0 \\ \gamma_1 + \gamma_2 = e \\ \gamma_1, \gamma_2 \geq 0 \end{array} \right. \quad (16)$$

No caso da norma  $l_\infty$ , o problema (4) pode ser formulado como um problema linear cujo par primal-dual apresentamos de seguida:

$$\left. \begin{array}{l} \min \sigma \\ \text{s.a. } Pw + e\sigma \geq \bar{x} \\ \quad -Pw + e\sigma \geq -\bar{x} \\ ew = 1 \\ w_i \geq 0 \ (i : p^i \in P_2) \end{array} \right\} = \left\{ \begin{array}{l} \max (\gamma_1 - \gamma_2)\bar{x} + \rho \\ \text{s.a. } (\gamma_1 - \gamma_2)P_1 + \rho e = 0 \\ (\gamma_1 - \gamma_2)P_2 + \rho e \leq 0 \\ (\gamma_1 + \gamma_2)e_n = 1 \\ \gamma_1, \gamma_2 \geq 0 \end{array} \right. \quad (17)$$

Em qualquer dos casos, se  $(\hat{\rho}, \hat{\gamma}_1, \hat{\gamma}_2)$  denota uma qualquer solução dual óptima, então o vector  $\hat{a}$  a que se refere o Lema 1 é caracterizado por  $\hat{a} = \hat{\gamma}_2 - \hat{\gamma}_1$ . Mais, se  $(\hat{\sigma}, \hat{w})$  denota uma qualquer solução primal básica admissível que seja óptima, então, pelo Lema 3, o novo conjunto  $P_1$  é independente afim sem que isso seja imposto à parte.

## 4 Experiência computacional

Implementámos o algoritmo da Figura 1 e efectuámos diversos testes computacionais. Começámos por testar a implementação com base nos problemas analisados em (Wolfe, 1976) que são, essencialmente, pontos uniformemente distribuídos sobre o cubo  $n$ -dimensional. Os resultados são apresentados na Subsecção 4.1.

Testámos também o algoritmo da Figura 1 como gerador de cortes para problemas de optimização discreta, definidos apenas pela norma Euclidiana. Os resultados obtidos com uma colecção de problemas teste proveniente da MIPLIB (Bixby et al., 1993, 1998), disponível electronicamente em <http://miplib.zib.de/miplib3/miplib.html>, são apresentados na Subsecção 4.2. Os resultados obtidos com uma colecção de problemas teste proveniente da OR LIB, disponível electronicamente em <http://mscmga.ms.ic.ac.uk/jeb/orlib/sppinfo.html>, são apresentados na Subsecção 4.3. O primeiro conjunto contém essencialmente problemas lineares com variáveis binárias, enquanto que o segundo conjunto contém problemas de partição (do inglês, *set partitioning problems*).

A implementação é constituída por um conjunto de rotinas escritas em Matlab, versão 6.0 (Mat, 1999). Todos os testes computacionais foram obtidos num PC com processador Intel Pentium II a 350MHz, com 128Mb de RAM e 256K de cache.

### 4.1 Experiência computacional com diferentes normas

Testámos a implementação com a resolução do problema (3) para as normas  $l_1$ ,  $l_2$  e  $l_\infty$ . Conforme sugerido em (Wolfe, 1976), o conjunto de base é um conjunto  $K^0$  de  $m$  pontos uniformemente distribuídos sobre o hipercubo  $n$ -dimensional  $[0, 2]^n$ . Cada problema é caracterizado por:

*Tipo 0:*  $K^0$ ;

*Tipo 1:*  $X^0 + K^0$ , sendo  $X^0$  um ponto escolhido aleatoriamente no conjunto  $2K^0$ ;

*Tipo 2:*  $\{(1 + 10^{-3}x_1, x_2, \dots, x_n) : (x_1, x_2, \dots, x_n) \in K^0\}$ , e

*Tipo 3:*  $\{10^{-2} + 10^{-3}x_1, x_2, \dots, x_n) : (x_1, x_2, \dots, x_n) \in K^0\}$ .

As Tabelas 1, 2, 3 e 4 apresentam o comportamento médio do algoritmo para problemas do Tipo 0, Tipo 1, Tipo 2 e Tipo 3, respectivamente, em termos do número de ciclos maiores e menores, número de pontos de  $K$  necessários para determinar a solução e tempo de CPU, utilizando as normas  $l_1$ ,  $l_2$  e  $l_\infty$ . Os resultados referem-se a valores médios de 10 amostras de várias dimensões. Quando  $m > n$ , o número de pontos de  $K$  necessários para determinar a solução em problemas do Tipo 0 é  $n + 1$ , do Tipo 1 é, em geral, pequeno (entre 2 e 8), do Tipo 2 e 3 é aproximadamente  $n$ , com a excepção no Tipo 3 e norma  $l_\infty$  que é 1.

### 4.2 Experiência computacional com problemas lineares 0-1

Testámos o algoritmo da Figura 1 como gerador de cortes em problemas lineares 0-1. Considerámos apenas a norma Euclideana, pelo que o Figura 1 se torna o algoritmo original de Ph. Wolfe, e diversos tipos de disjunções. A Tabela 5 descreve os problemas teste seleccionados de acordo com a seguinte estrutura: as primeiras três colunas contêm o nome, o número de restrições e o número de variáveis; as quarta e quinta colunas contêm o valor óptimo e o número de nodos da árvore do *branch-and-bound* requeridos para otimizar o problema através do Cplex 8.0 (ILO, 2002), com definições *default* dos parâmetros; e, as duas últimas colunas contêm, relativamente à relaxação linear, o valor óptimo e o número de componentes fraccionárias na solução óptima encontrada.

Passemos à descrição das implementações efectuadas do algoritmo de Ph. Wolfe, como gerador de cortes, em cada uma das aplicações *Bundle.sc*, *Bundle.1d*, *Bundle.2d* e *Bundle.c*. Na aplicação *Bundle.sc*, o objectivo foi testar a qualidade de cortes mais profundos que separam o invólucro convexo inteiro da solução óptima da relaxação linear. Nas aplicações *Bundle.1d* e *Bundle.2d*, o objectivo foi testar a qualidade de cortes disjuntivos baseados nas restrições de integralidade das

Tipo 0								
Norma $l_1$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	12,9	11,7	11,7	11,1	11,1	11,2	11,0	11,2
	14,0	11,4	11,4	10,2	10,2	10,4	10,0	10,4
	10,8	11,0	11,0	11,0	11,0	11,0	11,0	11,0
	0,290	0,234	0,257	0,289	0,276	0,303	0,320	0,355
n=20	16,5	26,7	22,4	21,8				
	17,9	31,9	22,8	21,6				
	14,1	20,5	21,0	21,0				
	0,382	0,816	0,638	0,648				
n=30	17,7	30,3						
	19,1	35,7						
	15,3	23,9						
	0,506	1,187						
Norma $l_2$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	11,9	11,3	11,0	11,0	11,0	11,0	11,0	11,0
	12,1	10,6	10,0	10,0	10,0	10,0	10,0	10,0
	11,0	11,0	11,0	11,0	11,0	11,0	11,0	11,0
	0,141	0,165	0,210	0,336	0,310	0,344	0,394	0,438
n=20	15,8	23,6	21,2	21,9				
	15,8	25,7	20,4	21,8				
	14,8	20,5	21,0	21,0				
	0,334	0,513	0,522	0,651				
n=30	17,4	27,3						
	17,0	29,6						
	16,8	24,0						
	0,547	0,901						
Norma $l_\infty$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	13,4	11,7	11,5	11,3	11,6	11,3	11,7	11,3
	15,0	11,4	11,0	10,6	11,2	10,6	11,4	10,6
	10,8	11,0	11,0	11,0	11,0	11,0	11,0	11,0
	0,252	0,205	0,233	0,263	0,290	0,292	0,333	0,340
n=20	16,0	25,0	22,5	22,8				
	17,4	28,9	23,0	23,6				
	13,6	20,1	21,0	21,0				
	0,305	0,628	0,545	0,617				
n=30	17,6	31,5						
	19,6	38,9						
	14,6	23,1						
	0,388	0,986						

**Tabela 1.** Problemas do Tipo 0: número de ciclos maiores e menores, número de pontos e tempo de CPU.

variáveis. Na aplicação *Bundle\_c*, o objectivo foi testar a qualidade de cortes mais profundos com base no invólucro inteiro de cada uma das restrições do problema, separadamente.

As Tabelas 6, 7, 8 e 9 resumem os resultados numéricos obtidos. Para o número de cortes gerados pelo algoritmo de Ph. Wolfe, que se apresenta na coluna CUTS, a coluna W-IT indica o número acumulado de iterações do Ciclo maior do algoritmo efectuadas, e a coluna W-LP, o número total de iterações feitas pelo Cplex ao resolver os problemas lineares envolvidos na geração desses cortes. A coluna ND indica o número de nodos utilizados pelo Cplex para otimizar o problema original após a adição do número de cortes indicado na coluna CUTS, ou A-CUTS. O valor apresentado na coluna IMP corresponde à melhoria obtida, relativa ao número de nodos da árvore do *branch-and-bound* requerido para otimizar o problema através do Cplex com definições *default* dos parâmetros. Seguidamente descrevemos aspectos mais específicos de cada aplicação.

*Bundle\_sc*: A aplicação inicia-se com a leitura e optimização da relaxação linear. O seguinte procedimento é repetido até que o número de nodos na resolução do problema original seja zero ou o número de cortes gerados seja igual a dez. Gerar um corte que separa  $\bar{x}$ , uma solução óptima da relaxação linear  $\bar{P}$ , do invólucro inteiro, acrescentar esse corte a  $\bar{P}$  e optimizar o problema linear 0-1 resultante. Após cada corte introduzido, registar também o número de nodos (em ND) que seriam necessários para resolver o problema original. A Tabela 6 resume os resultados obtidos.

Tipo 1								
Norma $l_1$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	3,1	4,6	4,4	4,0	5,4	5,2	5,3	5,1
	2,4	4,8	4,3	3,6	6,1	5,1	6,0	6,1
	2,8	3,4	3,5	3,4	3,7	4,3	3,6	3,1
	0,041	0,085	0,086	0,084	0,143	0,140	0,164	0,176
n=20	3,4	4,4	5,6	5,4				
	2,7	4,1	5,8	5,2				
	3,1	3,7	4,4	4,6				
	0,050	0,089	0,131	0,131				
n=30	4,5	4,9						
	3,7	4,1						
	4,3	4,7						
	0,079	0,100						
Norma $l_2$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	3,9	4,6	4,6	3,9	4,7	4,4	3,9	4,8
	2,9	3,8	3,8	2,9	3,9	3,5	2,9	4,1
	3,9	4,4	4,4	3,9	4,5	4,3	3,9	4,5
	0,047	0,101	0,096	0,096	0,141	0,149	0,149	0,201
n=20	4,2	5,1	5,2	5,5				
	3,2	4,1	4,2	4,6				
	4,2	5,1	5,2	5,4				
	0,098	0,140	0,163	0,197				
n=30	5,2	7,0						
	4,2	6,0						
	5,2	7,0						
	0,173	0,265						
Norma $l_\infty$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	5,7	6,6	7,0	7,2	8,1	7,6	9,5	7,7
	6,1	8,0	8,7	9,1	10,0	9,6	12,6	9,9
	4,3	4,2	4,3	4,3	5,1	4,6	5,4	4,5
	0,091	0,126	0,150	0,177	0,217	0,218	0,302	0,259
n=20	7,7	10,1	10,2	11,6				
	8,9	13,3	12,6	15,0				
	5,5	5,9	6,8	7,2				
	0,136	0,228	0,247	0,312				
n=30	9,4	9,6						
	11,0	11,0						
	6,8	7,2						
	0,182	0,210						

**Tabela 2.** Problemas do Tipo 1: número de ciclos maiores e menores, número de pontos e tempo de CPU.

*Bundle\_1d:* A aplicação inicia-se com a leitura e optimização da relaxação linear  $\bar{P}$ . O seguinte procedimento é repetido para cada componente fraccionária  $\bar{x}_j$  na solução óptima  $\bar{x}$  da relaxação linear  $\bar{P}$ . Gerar o corte mais profundo entre  $\bar{x}$  e o invólucro convexo de  $P \cap \{x: x_j \in \{0, 1\}\}$ . Os coeficientes das variáveis nesse corte são fortalecidos, usando os procedimentos descritos em (Balas and Jeroslow, 1980). Do conjunto de todos os cortes gerados, selecciona-se apenas o mais profundo no sentido da distância Euclidiana<sup>1</sup>. O termo independente desse corte é fortalecido, por aplicação do procedimento de “strong branching” (ver (ILO, 2002, pág. 252)), o corte assim obtido é acrescentado à relaxação linear  $\bar{P}$  e o problema linear resultante é optimizado. Seja  $\bar{x}_k$  a componente que definiu o corte adicionado. Se esta componente permanece fraccionária, então ocorre o seguinte procedimento: é gerado apenas mais um corte com base na disjunção  $x_k = 0 \vee x_k = 1$ . Este corte é fortalecido e acrescentado à relaxação linear  $\bar{P}$ . O problema linear 0-1 resultante é optimizado. A coluna A-CUTS indica se este procedimento ocorreu (2 cortes acrescentados) ou não (1 só corte acrescentado). A Tabela 7 resume os resultados obtidos.

*Bundle\_2d:* A aplicação inicia-se com a leitura e optimização da relaxação linear  $\bar{P}$ . Obter uma solução óptima  $\bar{x}$  da relaxação linear e efectuar o seguinte procedimento. Identificar o par  $(\bar{x}_j, \bar{x}_k)$

<sup>1</sup> a distância Euclidiana de  $\bar{x}$  ao hiperplano  $\alpha x = \beta$  é de valor igual  $(\beta - \alpha \bar{x}) / \|\alpha\|_2$ .

Tipo 2								
Norma $l_1$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	13,3	20,8	23,5	30,7	29,5	30,9	31,3	36,6
	16,5	30,6	36,0	47,1	48,0	47,5	49,8	56,8
	9,1	10,0	10,0	13,3	10,0	10,0	11,8	15,4
	0,277	0,550	0,693	0,981	1,055	1,129	1,230	1,505
n=20	15,6	34,2	48,5	52,4				
	16,8	44,1	72,0	80,0				
	13,4	23,3	24,0	23,8				
	0,359	1,118	1,953	2,282				
n=30	16,6	31,6						
	17,3	38,7						
	14,9	23,5						
	0,441	1,283						
Norma $l_2$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	14,2	19,7	23,3	26,1	25,2	28,3	27,9	28,7
	18,0	28,4	35,6	41,2	39,5	45,6	44,8	46,4
	9,4	10,0	10,0	10,0	9,9	10,0	10,0	10,0
	0,175	0,327	0,500	0,664	0,745	0,960	1,069	1,212
n=20	15,5	29,2	42,1	45,1				
	15,4	38,0	63,2	69,2				
	14,6	19,4	20,0	20,0				
	0,319	0,668	1,125	1,396				
n=30	15,7	28,6						
	15,2	31,4						
	15,2	24,8						
	0,507	0,938						
Norma $l_\infty$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0
	0,004	0,006	0,009	0,011	0,013	0,016	0,018	0,021
n=20	1,0	1,1	1,0	1,0				
	0,0	0,1	0,0	0,0				
	1,0	1,1	1,0	1,0				
	0,004	0,006	0,009	0,011				
n=30	1,0	1,0						
	0,0	0,0						
	1,0	1,0						
	0,004	0,007						

**Tabela 3.** Problemas do Tipo 2: número de ciclos maiores e menores, número de pontos e tempo de CPU.

de componentes mais fracionárias e gerar o corte mais profundo entre  $\bar{x}$  e o invólucro convexo de  $P \cap \{x: x_j, x_k \in \{0, 1\}\}$ . Este corte é acrescentado à relaxação linear. Se ambas as componentes se mantêm fracionárias, então é gerado um novo corte mais profundo usando a mesma disjunção, à semelhança da aplicação anterior. A Tabela 8 resume os resultados obtidos.

*Bundle-c:* A aplicação inicia-se com a leitura e otimização da relaxação linear  $\bar{P}$ . O seguinte procedimento é repetido para cada restrição do problema  $a_i x \leq b_i$ . Gerar o corte mais profundo entre  $\bar{x}$ , uma solução ótima da relaxação linear  $\bar{P}$ , e o invólucro convexo de  $\{x: a_i x \leq b_i, x \in \{0, 1\}^n\}$ . Do conjunto de todos os cortes gerados, selecciona-se apenas o mais profundo, no sentido da distância Euclidiana, que é acrescentado à relaxação linear  $\bar{P}$ . O problema linear 0-1 resultante é otimizado. A Tabela 9 resume os resultados obtidos.

### 4.3 Experiência computacional com problemas de partição

Testámos o algoritmo de Ph. Wolfe como gerador de cortes em problemas de partição. Na resolução de problemas de partição através do método *Branch-and-bound*, Ryan e Foster sugeriram uma regra específica de ramificação em (Ryan and Foster, 1981). Ceria e Pataki usaram o mesmo

Tipo 3								
Norma $l_1$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	13,4	22,2	23,2	25,6	29,2	34,8	28,8	30,6
	16,3	33,4	35,4	40,2	47,4	51,3	46,6	50,2
	9,5	10,0	10,0	10,0	10,0	17,3	10,0	10,0
	0,272	0,754	0,895	0,853	1,043	1,241	1,140	1,297
n=20	16,4	31,1	44,8	50,0				
	18,9	41,7	68,6	79,0				
	12,9	19,5	20,0	20,0				
	0,403	1,054	1,852	2,245				
n=30	16,5	30,6						
	17,2	38,0						
	14,8	22,2						
	0,449	1,236						
Norma $l_2$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	13,4	20,6	23,8	26,5	27,6	28,7	29,8	28,2
	16,4	30,2	36,6	42,0	44,2	46,4	48,6	45,4
	9,4	10,0	10,0	10,0	10,0	10,0	10,0	10,0
	0,160	0,495	0,621	0,670	0,822	0,977	1,136	1,192
n=20	15,0	29,1	40,0	47,1				
	15,3	37,7	59,0	73,2				
	13,7	19,5	20,0	20,0				
	0,319	0,655	1,058	1,459				
n=30	15,5	25,3						
	15,0	26,9						
	15,0	22,7						
	0,509	0,853						
Norma $l_\infty$	m=20	m=40	m=60	m=80	m=100	m=120	m=140	m=160
n=10	14,6	19,2	21,4	24,1	25,2	25,4	26,2	25,9
	18,8	27,7	32,1	37,6	39,8	40,2	41,9	41,1
	9,4	9,7	9,7	9,6	9,6	9,6	9,5	9,7
	0,284	0,552	0,613	0,738	0,835	0,901	0,993	1,036
n=20	16,1	29,3	37,8	41,3				
	17,9	38,7	55,3	62,4				
	13,3	18,9	19,3	19,2				
	0,309	0,846	1,345	1,595				
n=30	16,9	31,4						
	18,5	38,5						
	14,3	23,3						
	0,366	0,970						

Tabela 4. Problemas do Tipo 3: número de ciclos maiores e menores, número de pontos e tempo de CPU.

Problema	Linhas	Colunas	Solução inteira	Número de nodos	Solução linear	Número de var. fracc.
bm23	20	27	34	114	20.57	6
enigma	21	100	0	3521	0	6
lseu	28	89	1120	280	834.68	10
p0033	16	33	3089	5	2520.57	7
p0040	13	40	62027	0	61830	2
p0201	133	201	7615	311	6875	40
pipex	25	48	788.263	36	773.8	6
sentoy	30	60	-7772	108	-7839	8
stein9	13	9	5	9	4	6
stein27	118	27	18	3667	13	21
stein45	331	45	30	54359	22	35

Tabela 5. Descrição dos problemas teste lineares 0-1.

princípio para gerar cortes disjuntivos mais separadores em (Ceria and Pataki, 1998). Os problemas

Problema	CUTS	W-IT	W-NC	ND	CUTS	W-IT	W-NC	ND
bm23	3	42	36446	8	4	54	41140	6
	5	67	47728	1	6	79	52592	3
	7	87	69034	0				
enigma	1	4	36328	20	2	6	36706	0
lseu	3	258	201939	148	4	340	270591	175
	5	429	300375	169	6	519	342352	357
	7	625	527176	227	8	727	601491	150
	9	815	665021	144	10	903	692136	147
p0033	3	38	736	6	4	56	2324	4
	5	69	2348	4	6	84	2526	17
	7	97	2796	2	8	113	2852	4
	9	125	3000	2	10	141	3198	15
p0040	1	20	0	0				
p0201	3	329	707715	235	4	434	775454	207
	5	547	869431	260	6	659	962273	220
	7	800	1153337	323	8	943	1290638	295
	9	1074	134543	299	10	1205	1504062	569
pipex	3	49	11385	46	4	68	15300	18
	5	83	17909	24	6	106	20830	7
	7	125	24609	4	8	147	34429	2
	9	169	42363	0				
sentoy	3	134	65118	164	4	187	112926	156
	5	242	145072	114	6	286	166309	152
	7	331	213989	141	8	384	265517	155
	9	430	311669	87	10	490	501656	130
stein9	1	7	14	0				
stein27	1	25	10009	3644	2	60	155270	3
	3	76	155383	0				
stein45	3	139	1019358	44829	4	182	1068410	49814
	5	236	1220510	39416	6	283	1272705	51140
	7	325	1343509	48309	8	368	1399658	446225
	9	414	1454693	48542	10	470	1637195	46587

Tabela 6. Resultados computacionais do *Bundle\_sc*.

Problema	CUTS	W-IT	W-LP	A-CUTS	ND	IMP
bm23	6	160	2535	1	186	-63.158%
enigma	7	128	1727	1	40	+98.864%
lseu	11	1238	15498	2	188	+32.857%
p0033	8	335	2476	2	7	-40.000%
p0040	2	25	88	1	0	0%
p0201	40	6220	168816	1	381	-22.508%
pipex	6	132	1213	1	26	+27.778%
sentoy	8	586	9712	1	190	-75.926%
stein9	7	46	314	2	2	+77.778%
stein27	22	398	8821	2	3819	-4.145%
stein45	36	1570	61239	2	58182	-7.033%

Tabela 7. Resultados computacionais do *Bundle\_1d*.

de partição têm, genericamente, o seguinte aspecto:

$$\begin{aligned}
 \min cx &= \min \sum_{j=1}^n c_j x_j \\
 \text{s.a } Ax = e, & \quad \text{s.a } \sum_{j \in S_i} x_j = 1, \quad i = 1, 2, \dots, m, \\
 x \in \{0, 1\}^n, & \quad x_j \in \{0, 1\}, \quad j = 1, 2, \dots, n,
 \end{aligned} \tag{18}$$

Problema	A-CUTS	W-IT	W-LP	ND	IMP
bm23	1	17	406	190	-66.667%
enigma	2	32	691	219	+93.780%
lseu	1	61	1657	137	+51.071%
p0033	1	21	291	6	-20.000%
p0040	1	12	51	0	0%
p0201	2	282	11105	243	+21.865%
pipex	1	15	269	23	+36.111%
sentoy	2	116	4029	113	-4.630%
stein9	-	-	-	-	-
stein27	1	5	379	3736	-1.882%
stein45	2	46	3143	61505	-13.146%

Tabela 8. Resultados computacionais do *Bundle\_2d*.

Problema	CUTS	W-IT	W-NC	ND	IMP
bm23	20	356	93	180	-57.895%
enigma	21	265	7319097	1029	+70.775%
lseu	28	615	1800	198	+29.286%
p0033	16	237	112	4	+20.000%
p0040	11	60	0	0	0%
p0201	133	2119	9203062	646	-107.717%
pipex	25	363	0	48	-33.333%
sentoy	30	746	5361	81	+25.000%
stein9	13	60	0	9	0%
stein27	118	985	0	3500	+4.554%
stein45	331	2626	0	41365	+23.904%

Tabela 9. Resultados computacionais do *Bundle\_c*.

onde  $A \in \{0, 1\}^{m \times n}$ ,  $e$  denota um vector-coluna de tudo-uns e  $S_i = \{j: a_{ij} = 1\}$  denota o conjunto de suporte da linha  $i$  da matriz  $A$ . A regra de Ryan-Foster baseia-se na observação de que, dada uma solução fraccionária  $\bar{x}$ , ponto extremo de uma relaxação linear  $\bar{P} \subseteq [0, 1]^n$ , é sempre possível determinar um conjunto não singular de restrições que são cobertas fraccionariamente pelas variáveis de dois conjuntos complementares entre si. Isto é, se  $\bar{x}$  é um ponto extremo fraccionário de  $\bar{P}$ , então existe  $I \subseteq \{1, 2, \dots, m\}$  tal que

$$0 < \sum_{k \in T_I} \bar{x}_k < 1, \quad (19)$$

onde  $T_I \equiv \cap_{i \in I} S_i$ . Então,  $\bar{x}$  não pertence ao poliedro definido por

$$\begin{aligned} & \text{conv} \left( \left( \bar{P} \cap \{x: \sum_{k \in T_I} x_k = 0\} \right) \cup \left( \bar{P} \cap \{x: \sum_{k \in T_I} x_k = 1\} \right) \right) \\ &= \text{conv} \left( (\bar{P} \cap \{x: x_k = 0, k \in T_I\}) \cup (\bar{P} \cap \{x: x_k = 0, k \in \bar{T}_I\}) \right), \end{aligned} \quad (20)$$

onde  $\bar{T}_I \equiv (\cup_{i \in I} S_i) \setminus T_I$ . O leitor poderá ver a demonstração da existência de pelo menos um conjunto  $I$  de cardinalidade dois em (Vance, 1993, Proposição 2.2), por exemplo.

Em geral, dada uma solução fraccionária  $\bar{x}$  da relaxação linear do problema (18), existem vários conjuntos  $I$  de restrições que podem ser utilizados para definir a disjunção em (20). A regra de prioridade que estabelecemos consiste em ordenar os conjuntos identificados pelo maior valor da função definida por

$$f(I) = \sum_{k \in T_I} \bar{x}_k < 1,$$

em que  $T_I \equiv \cap_{i \in I} S_i$  para  $I \subseteq \{1, 2, \dots, m\}$ .

A Tabela 10 descreve as características dos problemas teste seleccionados. Seguidamente descrevemos a aplicação.



Problema	Linhas	Colunas	Solução inteira	Número de nodos	Solução linear	Número de var. fracc.
nw21	25	577	7408	1	7380.00	7
nw22	23	619	6984	1	6942.00	7
nw23	19	711	12534	21	12317.00	6
nw24	19	1366	6314	3	5843.00	6
nw25	20	1217	5960	2	5852.00	8
nw26	23	771	6796	1	6743.00	5
nw27	22	1355	9933	1	9877.50	3
nw28	18	1210	8298	2	8169.00	3
nw29	18	2540	4274	32	4185.33	12
nw30	26	2653	3942	7	3726.80	8
nw31	26	2662	8038	9	7980.00	5
nw32	19	294	14877	10	14570.00	4
nw33	23	3068	6678	2	6484.00	6
nw34	20	899	10488	1	10453.50	4
nw35	23	1709	7216	1	7206.00	4
nw36	20	1783	7314	11	7260.00	6
nw37	19	770	10068	2	9961.50	4
nw38	23	1220	5558	1	5552.00	6
nw39	25	677	10080	3	9868.50	3
nw40	19	404	10809	1	10658.25	9
nw41	17	197	11307	2	10972.50	3
nw42	23	1079	7656	3	7485.00	7
nw43	18	1072	8904	1	8897.00	7

Tabela 10. Descrição dos problemas teste de partição.

*Bundle\_RF*: A aplicação inicia-se com a leitura e optimização da relaxação linear  $\bar{P}$ . Gerar o corte disjuntivo mais profundo que separa  $\bar{x}$ , uma solução óptima  $\bar{x}$  da relaxação linear  $\bar{P}$ , do invólucro inteiro, baseado na disjunção de Ryan-Foster. Para a selecção dos dois conjuntos de variáveis que definem a disjunção, as restrições do problema são analisadas de modo a identificar o melhor conjunto de restrições, de cardinalidade dois ou três, que verifica a regra de Ryan-Foster. O corte gerado é acrescentado a  $\bar{P}$  e o problema linear 0-1 resultante é optimizado. A Tabela 11 resume os resultados obtidos. As colunas T e CT indicam a cardinalidade dos dois conjuntos de variáveis que definem a disjunção de Ryan-Foster, a coluna W-IT indica o número de iterações do Ciclo maior do algoritmo de Ph. Wolfe efectuadas, e a coluna W-NC, o número total de nodos utilizados pelo Cplex ao resolver os problemas lineares envolvidos na geração desse corte. A coluna ND indica o número de nodos utilizados pelo Cplex para optimizar o problema linear 0-1 resultante.

## Referências

- E. Balas and R. G. Jeroslow. Strengthening cuts for mixed integer programs. *European Journal of Operations Research*, 4:224–234, 1980.
- R. E. Bixby, E. A. Boyd, S. S. Dadmerhr, and R. Indovina. The MIPLIB mixed integer programming library. *COAL Bulletin*, 22, 1993.
- R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. P. Savelsberg. An updated mixed integer programming library: MIPLIB 3.0. *Optima*, 58:12–15, 1998.
- J. V. Burke and M. C. Ferris. Characterization of solution sets of convex programs. *Oper. Res. Lett.*, 10(1):57–60, 1991.
- S. Ceria and G. Pataki. Solving integer and disjunctive programs by lift and project. In *Integer programming and combinatorial optimization (Houston, TX, 1998)*, volume 1412 of *Lecture Notes in Comput. Sci.*, pages 271–283. Springer, Berlin, 1998.
- J.-B. Hiriart-Urruty and Cl. Lemaréchal. *Convex Analysis and Minimization Algorithms*, volume 1. Springer-Verlag, 1993a.
- J.-B. Hiriart-Urruty and Cl. Lemaréchal. *Convex Analysis and Minimization Algorithms*, volume 2. Springer-Verlag, 1993b.
- CPLEX 8.0*. ILOG Inc., 2002.
- G. L. Lawson and R. J. Hanson. *Solving least squares problems*. 1995.
- Using Matlab*. The Math Works Inc., 1999.
- D. Ryan and B. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer scheduling of public transport : urban passenger vehicle and crew scheduling: papers based on presentations at the international workshop held at the University of Leeds, 16-18 July, 1980*, pages 269–280. North-Holland Pub. Co., 1981.

Problema	2 restrições					3 restrições				
	T	CT	W-IT	W-NC	ND	T	CT	W-IT	W-NC	ND
nw21	180	234	52	835	1	120	253	46	693	1
nw22	119	239	42	664	1	57	381	38	579	1
nw23	276	252	47	510	19	82	456	24	310	24
nw24	409	548	40	572	3	268	877	56	683	3
nw25	356	428	77	1112	2	141	732	59	839	2
nw26	269	292	45	554	2	158	470	56	761	2
nw27	509	637	91	1355	1	388	749	83	1271	1
nw28	537	521	93	1131	3	311	843	133	1639	3
nw29	661	1167	122	3407	21	349	1850	180	5199	21
nw30	524	1516	136	3137	7	409	1657	137	2964	7
nw31	762	988	55	1418	6	374	1358	58	1399	18
nw32	17	117	27	307	19	20	206	22	284	13
nw33	888	1320	77	1925	2	517	1876	100	2167	2
nw34	208	229	37	734	1	161	522	45	759	1
nw35	591	723	37	1017	0	289	981	46	850	0
nw36	369	995	77	981	33	293	1113	46	1051	33
nw37	191	224	49	991	0	94	476	68	1217	0
nw38	507	475	48	813	4	396	634	54	1029	6
nw39	109	308	56	856	3	74	425	48	818	3
nw40	112	210	42	794	2	71	271	45	715	2
nw41	35	91	20	187	2	21	111	23	195	3
nw42	227	523	59	1185	2	189	460	44	1008	2
nw43	229	397	58	1438	1	99	721	61	1353	1

Tabela 11. Resultados computacionais do *Bundle\_RF*- 2 e 3 restrições.

J. Soares. *Disjunctive Convex Optimization*. PhD thesis, Graduate School of Business, Columbia University, New York, USA, June 1998.

P. Vance. *Crew Scheduling, Cutting-stock and Column Generation: solving huge integer programs*. PhD thesis, Georgia Institute of Technology, Georgia, Aug 1993.

Ph. Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 11:128–149, 1976.