Formulações para o Problema da Compra Mínima

Sandro H. C. Uliana[1], **Rafael C. S. Schouery**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
Caixa Postal 1.251 – 13.083-852 – Campinas – SP – Brasil

sandrouliana32@gmail.com, rcss@unicamp.br

Abstract. *We consider the Min-Buying Problem, where a seller wants to sell a set I of items to a set B of buyers. Each buyer b is willing to pay up to v_{ib} for item i. For this, the seller must define prices p_i for each $i \in I$ in order to maximize his profit considering that each buyer b will buy, if any, the most unexpensive item among those that $p_i \leq v_{ib}$. In this paper, we present and compare two integer linear programming formulations for such a problem.*

Resumo. *Consideramos o Problema da Compra Mínima, em que um vendedor deseja vender um conjunto I de itens para um conjunto B de compradores. Cada comprador b está disposto a pagar até v_{ib} pelo item i. O vendedor deve então definir preços p_i para cada $i \in I$ de forma a maximizar o seu lucro considerando que cada comprador b comprará (se existir) o item mais barato dentre aqueles que $p_i \leq v_{ib}$. Neste resumo, apresentamos e comparamos duas formulações de programação linear inteira para tal problema.*

1. Introdução

Um grande problema enfrentado por vendedores é determinar o preço de seus itens de forma que sua receita, ao final, seja máxima. Dentre as várias possíveis formas de modelar esse problema, estamos interessados no *Problema da Compra Mínima* [Rusmevichientong et al. 2006], onde compradores estão interessados em minimizar o valor gasto por eles. Tal problema tem como entrada um conjunto B de *compradores*; um conjunto I de *itens*; e uma matriz de *valorações* v indexada por $I \times B$, em que v_{ib} é o *valor* do item i para o comprador b, isto é, quanto b está disposto a pagar, no máximo, por i. Uma solução do problema é uma *precificação* p, em que p é um vetor indexado por I, que representa o preço dos itens. Neste problema consideramos que os itens possuem *oferta ilimitada*, isto é, o mesmo item pode ser vendido para vários compradores (como se os itens pudessem ser replicados conforme necessário), e que os compradores possuem *demanda unitária*, isto é, que estão dispostos a comprar no máximo um item (pois os itens são substitutos entre si). Assim, dada uma precificação p, cada comprador b recebe seu item viável de valor mínimo, ou nenhum caso não existam itens viáveis para ele. Um item i é viável para b se e somente se $p_i \leq v_{ib}$. O objetivo é encontrar uma precificação p que maximize a receita do vendedor, que é dada pela soma dos preços dos itens recebidos pelos compradores. Briest and Krysta (2007) demonstraram que esse problema não pode ser aproximado em tempo polinomial por $\mathcal{O}(\log^{\epsilon}|B|)$, a não ser que NP $\subseteq$ DTIME($|I|^{\mathcal{O}(\log\log|I|)}$).

Apesar desta ser a primeira vez que o Problema da Compra Mínima é abordado do ponto de vista de formulações de programação linear inteira, outros problemas de precificação já foram explorados previamente na literatura. Aggarwal et al. (2004) apresentaram um modelo de programação linear inteira para o *Problema da Compra Máxima*.

Shioda et al. (2011) abordaram o *Problema da Precificação Livre de Inveja* e apresentaram alguns resultados matemáticos obtidos a partir da modelagem. Heilporn et al. (2010) abordaram tanto a Precificação Livre de Inveja quanto a Precificação de Rede, e apresentaram formulações de programação inteira mista. Fernandes et al. (2016) também abordaram o mesmo problema, comparando diferentes formulações de programação linear inteira mista. Uma delas se refere à desenvolvida por Heilporn et al. (2010) para a Precificação de Rede com rodovias pedagiadas. Outra é uma adaptação da formulação de Myklebust et al. (2016). Fernandes et al. (2016) apresentaram ainda três modelos de instâncias para problemas de precificação com interpretações econômicas distintas.

2. Desenvolvimento

Para o desenvolvimento de formulações para a Compra Mínima, tomou-se como ponto de partida formulações desenvolvidas para outros problemas de precificação. A primeira, apresentada por Aggarwal et al. (2004), foi desenvolvida para o Problema da Compra Máxima. A segunda, apresentada por Heilporn et al. (2010) e estudada por Fernandes et al. (2016), foi desenvolvida para a Precificação Livre de Inveja.

Seja P_i o conjunto de preços possíveis para o item i. Inicialmente consideramos, sem perda de generalidade, que $P_i = \{v_{ib} \mid \forall i \in I, \forall b \in B\}$, isto é, que qualquer valoração pode ser atribuída como preço de um item i. Porém, posteriormente, discutimos como reduzir a cardinalidade deste conjunto. Seja também, para cada comprador b, $P'_b = \bigcup_{i \in I | v_{ib} > 0} P_i$ o conjunto dos preços que o comprador b pode pagar pelo item que porventura ele comprar.

Para a primeira formulação, inspirada na formulação para a Compra Máxima, as variáveis de decisão (todas binárias) são x_{ip}, que indica se o item i recebeu o preço p, e y_{bp}, que indica se o comprador b pagou o preço p pelo item que ele adquiriu. Denotamos tal formulação de (PLI), já que ela conta apenas com variáveis inteiras.

$$
\begin{aligned}
\text{(PLI)} \quad & \text{maximize} \sum_{b \in B} \sum_{p \in P'_b} p \cdot y_{bp} \\
& \text{sujeito a} \quad \sum_{p \in P_i} x_{ip} = 1 && \forall\, i \in I && (1) \\
& \qquad y_{bp} \leq \sum_{i \in I | p \leq v_{ib}} x_{ip} && \forall\, b \in B, \forall\, p \in P'_b && (2) \\
& \qquad \sum_{p \in P'_b} y_{bp} \leq 1 && \forall\, b \in B && (3) \\
& \qquad \sum_{\substack{p' < p \\ p' \leq v_{ib}}} x_{ip'} \leq 1 - y_{bp} && \forall\, b \in B, \forall\, p \in P'_b, \forall\, i \in I && (4) \\
& \qquad y_{bp} \in \{0, 1\} && \forall\, b \in B, \forall\, p \in P'_b && (5) \\
& \qquad x_{ip} \in \{0, 1\} && \forall\, i \in I, \forall\, p \in P_i && (6)
\end{aligned}
$$

A função objetivo é dada pela soma de todos os preços pagos pelos compradores. As restrições (1) garantem que todo item receba exatamente um único preço. Já as restrições (2) fazem com que se um comprador b paga um preço p ($y_{bp} = 1$), então existe pelo menos um item i viável para ele com preço p. As restrições (3) garantem que cada comprador pague no máximo um único preço, adquirindo, assim, um único item. As restrições (4) garantem que cada comprador receba seu item viável, caso exista algum, de menor preço, pois se $y_{bp} = 1$, então não pode existir nenhum item viável para b com preço menor que p, e caso exista algum $x_{ip'} = 1, p' < p$, então $y_{bp} = 0$.

O seguinte teorema, proposto pelos autores, auxilia na redução do número de variáveis do modelo.

Teorema 1. *Sem perda de generalidade, se p é uma solução ótima para o Problema da Compra Mínima, então, para todo item $i \in I$, $p_i \in \{v_{ib} \mid b \in B\} \setminus \{0\}$, isto é, $P_i = \{v_{ib} \mid \forall b \in B\} \setminus \{0\}$.* □

A segunda formulação, inspirada na formulação para a Precificação Livre de Inveja, tem como variáveis de decisão a variável binária x_{ib}, que indica se um item i foi alocado ao comprador b; a variável binária y_{ib}, que indica se o item i é viável para o comprador b; a variável contínua p_i, que representa o preço do item i; e a variável contínua $\hat{p}_{ib}$, que indica o preço que o comprador b pagou pelo item i (p_i caso b tenha recebido i, e 0 caso contrário). Além disso, definimos $R_i = \max\{v_{ib} \mid b \in B\}, \forall i \in I$; $\epsilon = \min\{v_{ib} - v_{i'b'} \mid \forall i, i' \in I, \forall b, b' \in B, v_{ib} \neq v_{i'b'}\}$; e $S_b = \{i \mid v_{ib} > 0\}, \forall b \in B$. Denotamos tal formulação de (PLIM), por se tratar de um modelo linear inteiro misto.

$$
\begin{array}{llll}
\text{(PLIM)} & \text{maximize} \displaystyle\sum_{b \in B} \sum_{i \in S_b} \hat{p}_{ib} & & \\
& \text{sujeito a} \quad \displaystyle\sum_{i \in S_b} x_{ib} \leq 1 & \forall\, b \in B & (7) \\
& v_{ib} x_{ib} - \hat{p}_{ib} \geq 0 & \forall\, b \in B, \forall\, i \in S_b & (8) \\
& \hat{p}_{ib} \geq p_i - R_i(1 - x_{ib}) & \forall\, b \in B, \forall\, i \in S_b & (9) \\
& \hat{p}_{ib} \leq p_i & \forall\, b \in B, \forall\, i \in S_b & (10) \\
& y_{ib} \geq \dfrac{v_{ib} - p_i + \epsilon}{v_{ib} + \epsilon} & \forall\, b \in B, \forall\, i \in S_b & (11) \\
& p_i \geq \displaystyle\sum_{j \in S_b} \hat{p}_{jb} - R_i(1 - y_{ib}) & \forall\, b \in B, \forall\, i \in S_b & (12) \\
& x_{ib},\, y_{ib} \in \{0, 1\} & \forall\, b \in B, \forall\, i \in S_b & (13) \\
& \hat{p}_{ib} \geq 0 & \forall\, b \in B, \forall\, i \in S_b & (14) \\
& p_i \geq 0 & \forall\, i \in I & (15)
\end{array}
$$

As restrições (7) garantem que cada comprador receba no máximo um único item. Quando $x_{ib} = 0$, as restrições (8) e (14) garantem que $\hat{p}_{ib} = 0$. Já quando $x_{ib} = 1$, as restrições (9) e (10) garantem que $\hat{p}_{ib} = p_i$. As restrições (11) garantem que $y_{ib} = 1$ se o item i é viável para b. As restrições (12) garantem que cada comprador receba seu item viável de valor mínimo, caso exista algum, pois quando $y_{ib} = 1$, i é viável para b e o preço pago por b pelo item que irá receber deve ser menor ou igual p_i.

3. Resultados Computacionais

Para avaliação e comparação iniciais das formulações, foram realizados testes de desempenho utilizando uma máquina Intel®Xeon®CPU E5-2630 v4 2.20GHz com 32GB de memória RAM, sistema operacional Ubuntu 20.04.4 LTS. As formulações foram implementadas utilizando a linguagem C++ e a API do software Gurobi.

Para obtenção de instâncias, utilizou-se o gerador *Unit-Demand Market Models* [Fernandes et al. 2016]. Todas as instâncias testadas podem ser consideradas pequenas, pois vão de 50 compradores e 50 itens, até 150 compradores e 150 itens. As únicas instâncias densas são as do modelo *rc*, pois nela os compradores oferecem valorações não nulas para todos os itens. Além disso, foi utilizado um tempo limite de execução de 3600s.

Por meio da Tabela 1, podemos comparar o desempenho das formulações. A formulação (PLI), já reduzida em número de variáveis considerando o Teorema 1, obteve um desempenho superior ou equivalente para a maioria das instâncias, mesmo gerando um modelo maior. Curiosamente, o modelo foi sempre solucionado no nó raiz através do uso heurísticas primais e adição de cortes, não chegando a iniciar a fase de *branching*. Já a formulação (PLIM) obteve bons resultados apenas em instâncias pequenas, não sendo capaz de resolver três instâncias e obtendo um tempo de processamento maior, de forma geral. Em nossos experimentos observamos que tal formulação apresenta dificuldades em provar a otimalidade de uma solução encontrada por uma heurística primal.

Tabela 1. Tempo de execução ou GAP para cada uma das formulações.

			(PLI)		(PLIM)	
Instância	$\|I\|$	$\|B\|$	Gap	Tempo	Gap	Tempo
c-n=50-c=14-p=7-o=8-l=1-h=100-d=0.25	50	50	0%	**1.1s**	0%	149s
c-n=100-c=14-p=7-o=8-l=1-h=100-d=0.25	100	100	0%	**26s**	0.02%	3600s
c-n=150-c=14-p=7-o=8-l=1-h=100-d=0.25	150	150	0%	**625s**	0.06%	3600s
n-n=50-h=3-d=8-m=10	50	50	0%	0.7s	0%	**0.5s**
n-n=100-h=3-d=8-m=10	100	100	0%	**2.5s**	0%	15s
n-n=150-h=3-d=8-m=10	150	150	0%	**3.6s**	0%	18s
p-n=50-e=800-q=200-d=0.25	50	50	0%	**35s**	0%	1028s
p-n=100-e=800-q=200-d=0.25	100	100	0%	**12s**	0.04%	3600s
p-n=150-e=800-q=200-d=0.25	150	150	0%	**11s**	0%	1090s
rc-n=25	25	25	0%	22s	0%	**0.48s**
rc-n=50	50	50	0%	**248s**	0%	262s
rc-n=100	100	100	0%	1952s	0%	**394s**

4. Conclusões

Esta pesquisa foi motivada pelo aspecto prático do Problema da Compra Mínima e por sua investigação ser escassa na literatura até então. Foram propostas duas formulações para resolver o problema: uma baseada em uma formulação para o Problema da Compra Máxima (denotada por PLI) e outra baseada em uma formulação da Precificação Livre de Inveja (denotada por PLIM).

Por meio de testes computacionais foi possível demonstrar o desempenho superior de formulação (PLI) em instâncias pequenas. Tal formulação só pode ser usada na prática graças a um teorema que prova que basta considerar um subconjunto de preços para cada item. De fato, podemos considerar, para cada item, apenas preços que sejam iguais às valorações dadas àquele item. Tal teorema pode ser útil futuramente para o projeto de outros algoritmos para o Problema da Compra Mínima.

Os trabalhos futuros envolvem o estudo de meios para tornar as formulações mais enxutas, com menos variáveis e restrições, de forma que consigam processar instâncias mais densas, e também de outras formas de *branching* que possam ser mais eficientes que a utilizada por padrão pelo Gurobi. Ademais, pretende-se também utilizar geração de cortes para acelerar o processo de prova de otimalidade.

Referências

Aggarwal, G., Feder, T., Motwani, R., and Zhu, A. (2004). Algorithms for multi-product pricing. In *International Colloquium on Automata, Languages, and Programming*, pages 72–83. Springer.

Briest, P. and Krysta, P. (2007). Buying cheap is expensive: Hardness of non-parametric multi-product pricing. In *SODA*, volume 7, pages 716–725. Citeseer.

Fernandes, C. G., Ferreira, C. E., Franco, A. J., and Schouery, R. C. (2016). The envy-free pricing problem, unit-demand markets and connections with the network pricing problem. *Discrete Optimization*, 22:141–161.

Heilporn, G., Labbé, M., Marcotte, P., and Savard, G. (2010). A polyhedral study of the network pricing problem with connected toll arcs. *Networks: An International Journal*, 55(3):234–246.

Myklebust, T. G., Sharpe, M., and Tunçel, L. (2016). Efficient heuristic algorithms for maximum utility product pricing problems. *Computers & Operations Research*, 69:25–39.

Rusmevichientong, P., Van Roy, B., and Glynn, P. W. (2006). A nonparametric approach to multiproduct pricing. *Operations Research*, 54(1):82–98.

Shioda, R., Tunçel, L., and Myklebust, T. G. (2011). Maximum utility product pricing models and algorithms based on reservation price. *Computational Optimization and Applications*, 48(2):157–198.