Arredondamento de PL para o Problema de Localização de Instalações Balanceado*

Lehilton Lelis Chaves Pedrosa

[1] Instituto de Computação
Universidade de Campinas (Unicamp)
Campinas, SP

lehilton@ic.unicamp.br

Resumo. *No problema de localização de instalações, recebemos um espaço métrico contendo um conjunto de clientes e um conjunto de localizações, cada uma associada a um custo de abertura de instalação. O problema é escolher um subconjunto de localizações para abrir instalações de forma a minimizar o custo de abertura mais o custo de conectar cada cliente a alguma instalação aberta. Na versão balanceada do problema, cada cliente pode ser vermelho ou azul e só consideramos soluções em que o conjunto de clientes associado a cada instalação tenha tantos clientes vermelhos quantos azuis. Neste resumo, discutimos estratégias para construir algoritmos de aproximação para o problema utilizando arredondamento de PL.*

1. Introdução

Em problemas de particionamento, o objetivo é particionar um conjunto de clientes em um espaço métrico de forma a minimizar uma determinada função de custo. Versões clássicas dos problemas ignoram informações de origem dos clientes, o que em certas aplicações pode levar a decisões que amplificam vieses ou injustiças entre grupos (discrepâncias no número de representantes, etc.) [Chierichetti et al. 2017]. Essa questão levou a uma linha de trabalhos recentes, com objetivo de projetar algoritmos *justos*. Uma tal noção foi estudada por [Chierichetti et al. 2017], que definiram problemas de particionamento balanceados, em que o conjunto de entrada é dividido em dois grupos. Em uma solução, a fração do grupo minoritário em cada parte deve ser maior que um certo valor dado na entrada. Outras noções relacionadas incluem versões com várias cores com demandas por cores ou versões com *outliers* [Chakrabarty et al. 2020].

Chierichetti et al. consideraram apenas problemas em que o tamanho da partição é restrita a um dado número k (problemas dos k-centros ou das k-medianas), reduzindo as versões balanceadas para as versões clássicas. Em outros casos, como no problema de localização de instalações (*Facility Location Problem*, FLP), o número de partes é irrestrito, mas há um custo associado a cada parte instalada [Byrka and Aardal 2010]. Neste resumo, estudamos uma versão justa do FLP e consideramos o caso perfeitamente balanceado, em que cada grupo tem representação igual em cada parte. Primeiro, reduzimos o problema para a versão clássica ao custo de uma unidade no fator de aproximação. Depois, ilustramos como obter um fator melhor explorando arredondamento de PL.

*Apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo número 312186/2020-7.

2. Uma redução simples

Uma instância do FLP balanceado é composta por um conjunto de instalações F e um conjunto de clientes $D = R \cup B$ com $|R| = |B|$. Dizemos que um cliente em R é vermelho e um cliente em B é azul. O custo de conectar um cliente i a uma instalação k é $d(i, k)$ e o custo de abrir uma instalação k é $f(k)$. Uma solução é composta por um subconjunto de instalações abertas H bem como uma atribuição $\alpha : D \to H$ tal que para toda instalação $k \in H$, vale $|\alpha^{-1}(k) \cap R| = |\alpha^{-1}(k) \cap B|$ e o objetivo é encontrar uma solução que minimiza o custo total dado por $\sum_{i \in D} d(i, \alpha(i)) + \sum_{k \in H} f(k)$.

Chierichetti et al. definiram uma decomposição em *fairlets*. Para dois inteiros coprimos r, b com $r \leq b$, um r/b-fairlet é um conjunto com até $r+b$ clientes cuja fração de clientes vermelhos e de clientes azuis é cada uma pelo menos r/b. Os autores mostraram que uma solução de um problema r/b-balanceado pode ser decomposta em uma coleção de r/b-fairlets, de forma que resolver o problema balanceado se reduz a encontrar uma decomposição de custo mínimo. No caso de grupos perfeitamente balanceados, $r = b = 1$ e uma decomposição em fairlets é simplesmente um emparelhamento perfeito. Ajustando essa estratégia para o FLP balanceado, obtemos a seguinte redução.

Teorema 1. *Se FLP admite β-aproximação, FLP balanceado admite $(\beta{+}1)$-aproximação.*

3. Um algoritmo de arredondamento

Um limitante inferior para o valor ótimo do problema é dado pela relaxação do problema como o programa linear (PL) abaixo. A variável y_k indica que uma instalação k foi aberta e a variável x_{ijk} indica que i é pareada com j e que ambos clientes são atribuídos à instalação k.

$$
\begin{array}{lrl}
\min & \displaystyle\sum_{i \in R, j \in B, k \in F} (d(i,k) + d(j,k))\, x_{ijk} & + \displaystyle\sum_{k \in F} f(k)\, y_k \\
\text{sujeito a} & \sum_{j \in B, k \in F} x_{ijk} = 1 & \forall\, i \in R, \\
& \sum_{i \in R, k \in F} x_{ijk} = 1 & \forall\, j \in B, \\
& \sum_{j \in B} x_{ijk} \leq y_k & \forall\, i \in R, k \in F, \\
& \sum_{i \in R} x_{ijk} \leq y_k & \forall\, j \in B, k \in F, \\
& x_{ijk}, y_k \geq 0 & \forall\, i \in R, j \in B, k \in F.
\end{array}
$$

Dada uma solução do PL, para cada $i \in R$, definimos $C_i^* = \sum_{j \in B, k \in F} d(i, k)\, x_{ijk}$ e, para cada $j \in B$, definimos C_j^* analogamente. Também, definimos $C^* = \sum_{i \in D} C_i^*$ e $F^* = \sum_{k \in F} f(k) y_k$. Com isso, podemos reescrever a função-objetivo como $C^* + F^*$. O *suporte* de um cliente $i \in D$, denotado por $S(i)$, é o conjunto de instalações k tais que $x_{ijk} > 0$. Uma instalação de $S(i)$ pode estar longe de i, então iremos utilizar uma estratégia de filtragem [Lin and Vitter 1992] para um fator de escala $\gamma > 1$. Mais precisamente, para cada cliente i, consideramos a ordenação de $S(i)$ em ordem não-decrescente de distância até i e definimos $N(i)$ como o conjunto dos elementos no menor prefixo dessa ordenação tal que $\sum_{k \in N(i)} y_k \geq 1/\gamma$.

Iremos supor, sem perda de generalidade, que a solução devolvida para o PL é *completa*, isso é, para cada variável x_{ijk}, ou $x_{ijk} = 0$ ou $x_{ijk} = y_k$ e, além disso, que para todo cliente i, $\sum_{k\in N(i)} y_k = 1/\gamma$. Caso contrário, poderíamos considerar uma instância equivalente do problema e modificar a solução devolvida de forma a torná-la completa [Byrka and Aardal 2010]. Para cada cliente $i \in R$, definimos as seguintes variáveis:

$$C_i^{\text{avg}} = \textstyle\sum_{j\in B, k\in N(i)} x_{ijk}\, d(i,k) \quad \text{e} \quad C_i^{\max} = \max_{k\in N(i)} d(i,k).$$

Definimos C_j^{avg} e $C_j^{\max}$ para cada cliente $j \in B$ analogamente. Observe que, como $N(i)$ contém as instalações mais próximas do suporte cujas variáveis somam $1/\gamma$, temos que $C_i^{\text{avg}} \leq C_i^*$ e $C_i^{\max} \leq \frac{\gamma}{\gamma-1} C_i^*$.

3.1. Algoritmo

Enquanto a redução da Seção 2 obtém um emparelhamento perfeito antes de executar um algoritmo para FLP como caixa-preta, ao utilizar um algoritmo de arredondamento de PL, podemos encontrar um emparelhamento e limitar seu custo diretamente pela função-objetivo. Fazemos isso com o seguinte lema, que é a ideia central para o novo algoritmo. Para cada par $i \in R$ e $j \in B$, defina

$$z_{ij} = \sum_{k\in F} x_{ijk}.$$

Lema 1. *Dada uma solução do PL, podemos encontrar um emparelhamento perfeito M entre R e B tal que cada par (i, j) aparece em M com probabilidade z_{ij}.*

Uma vez obtido um emparelhamento M, o problema se reduz a selecionar um subconjunto de instalações, já que cada par de clientes pode ser conectado à instalação mais próxima de um deles. Para todos os pares (i, j), vamos designar uniformemente um dos dois como *ativo*. Assim, ou todos os clientes vermelhos estarão ativos com probabilidade $1/2$, ou todos os clientes azuis estarão ativos. Em seguida, suponha que escolhemos como ativos os vértices vermelhos, já que o outro caso é simétrico.

Para selecionar instalações, utilizaremos um algoritmo de *clustering*. Cada parte corresponderá a um conjunto $N(i)$ em torno de um cliente $i \in R$, chamado de centro do *cluster*. Inicialize $F' \leftarrow F$. Enquanto houver $i \in R$ tal que $N(i) \subseteq F'$, escolha o cliente i que minimiza o valor de $C_i^{\text{avg}} + C_i^{\max}$, crie um cluster $N(i)$ e remova $N(i)$ de F'. Para cada cluster em torno de i, abrimos uma instalação $\ell \in N(i)$ com probabilidade $\gamma\, y_\ell$. Além disso, para cada instalação $k \in F'$, que não está em nenhum cluster, abrimos k independentemente com probabilidade $\gamma\, y_k$.

Finalmente, para cada par de clientes (i, j), conectamos ambos à instalação aberta k cuja soma $d(i, k) + d(j, k)$ é a menor possível.

3.2. Análise

Vamos separar a análise entre custo de abertura de instalação e custo de conexão. Para o custo de abertura, observamos que cada instalação $k \in F$ é aberta com probabilidade exatamente $\gamma\, y_k$, o que implica diretamente no seguinte limitante.

Lema 2. *O custo esperado de abertura de instalações é de no máximo γF.*

Para o custo de conexão, analisamos o custo esperado de conexão para cada par de M separadamente. Consideramos a seguinte estratégia (possivelmente sub-ótima) para conectar pares de clientes a instalações.

Primeiro, verificamos se existe alguma instalação aberta no suporte de i e, se existir, conectamos ambos clientes a ela. A probabilidade desse evento ocorrer e o custo esperado condicionado são dados pelo Lema 3, que é similar a lemas conhecidos para o FLP tradicional [Byrka and Aardal 2010]. Nesse caso, o custo de conexão do par de clientes pode ser limitado supondo que eles foram atribuídos à instalação aberta em $S(i)$.

Lema 3. *$Pr(H \cap S(i) \neq \emptyset) \geq 1 - e^{-\gamma}$ e $E\left[\min_{k \in S(i)} d(i,k) \middle| H \cap S(i) \neq \emptyset\right] \leq C_i^*$.*

Lema 4. *Se $H \cap S(i) \neq \emptyset$, o custo esperado de i, j é no máximo $2C_i^* + d(i,j)$.*

Se não existir instalação aberta em $S(i)$, então tampouco há instalação aberta em $N(i)$. Assim, existe instalação $k \in N(i') \cap N(i)$ em algum cluster cujo centro é i'. Nesse caso, conectamos o par de clientes à instalação aberta em $N(i')$.

Lema 5. *Se $H \cap S(i) = \emptyset$, o custo esperado de i, j é no máximo $2(2C_i^{\max} + C_i^{\text{avg}}) + d(i,j)$.*

Combinando os lemas anteriores, obtemos o resultado principal.

Teorema 2. *O algoritmo é uma $2{,}34$-aproximação para o FLP balanceado.*

4. Direções futuras

Para facilitar a apresentação, o algoritmo de arredondamento realiza apenas uma atribuição de instalações simplificada, mas estratégias de arredondamento mais sofisticadas [Li 2013, Benedito and Pedrosa 2019] podem levar a fatores de aproximação melhores. Mesmo esse algoritmo simplificado tem fator de aproximação menor do que o melhor possível com o Teorema 1, que é $2{,}488$ (utilizando o algoritmo de [Li 2013] para o FLP). Em trabalhos futuros, queremos estudar como estender esse algoritmo para versões do problema não perfeitamente balanceadas ou com mais de duas cores.

A ideia central do algoritmo é utilizar o Lema 1 para extrair um emparelhamento das variáveis do PL. Isso sugere que podemos encontrar um limitante inferior mais forte se codificarmos emparelhamentos diretamente na formulação, contanto que possamos encontrar uma solução do PL em tempo polinomial. Em um possível algoritmo, podemos realizar clustering em torno dos pares, ao invés de clientes individualmente.

Referências

Benedito, M. P. and Pedrosa, L. L. (2019). Approximation algorithms for median hub location problems. *Journal of Combinatorial Optimization*, pages 1–27.

Byrka, J. and Aardal, K. (2010). An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM J. Comput.*, 39:2212–2231.

Chakrabarty, D., Goyal, P., and Krishnaswamy, R. (2020). The non-uniform k-center problem. *ACM Transactions on Algorithms*, 16(4):1–19.

Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. (2017). Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Li, S. (2013). A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222(0):45–58.

Lin, J. and Vitter, J. S. (1992). ϵ-approximations with minimum packing constraint violation. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 771–782.