Algoritmos de Aproximação para o Problema de Extensão de Quadrados Latinos Diagonais

Vítor G. Chagas[1], **Flávio K. Miyazawa**[1]

[1] Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Cidade Universitária Zeferino Vaz - Barão Geraldo, Campinas, SP

{vitor.chagas,fkm}@ic.unicamp.br

Abstract. *In this work, we study the problem of extending latin squares restricted to diagonal latin squares. For this problem, we present three approximation algorithms adapted from the literature: a greedy algorithm, a matching-based algorithm, and a local search.*

Resumo. *Neste trabalho, estudamos o problema de extensão de quadrados latinos restrito a quadrados latinos diagonais. Para tal problema, apresentamos três algoritmos de aproximação adaptados da literatura: um algoritmo guloso, um algoritmo baseado em emparelhamento, e uma busca local.*

1. Introdução

Um *quadrado latino* é um grid quadrado composto por n^2 células preenchidas com n símbolos de forma que nenhum símbolo ocorra mais de uma vez em alguma linha ou coluna do grid. Por conta de sua relação com a teoria de quasigrupos e geometrias finitas, quadrados latinos tem sido estudados extensivamente dos pontos de vista algébrico e combinatório [Keedwell e Dénes 2015]. Também há aplicações práticas que utilizam o conceito de quadrados latinos, como no design de redes óticas [Kumar et al. 1999].

Neste trabalho, investigamos uma classe particular de quadrados latinos chamada de *quadrados latinos diagonais*, que possuem a restrição adicional de que nenhum símbolo ocorre mais de uma vez nas duas diagonais do grid. Alguns estudos focam em propriedades relacionadas à existência e construção de tais quadrados latinos [Hilton 1973, Hilton 1974], porém, para o melhor de nosso conhecimento, não há trabalhos dedicados à aproximabilidade de se estender quadrados latinos diagonais.

Por conta disso, estudamos algoritmos de aproximação para o problema. Apresentamos três algoritmos de aproximação adaptados da literatura: um algoritmo guloso, um algoritmo baseado em emparelhamento, e uma busca local.

1.1. Trabalhos Relacionados

A complexidade computacional de se estender quadrados latinos foi investigada por [Colbourn 1984, Hajirasouliha et al. 2007]. Algoritmos de aproximação para tal problema foram propostos por [Kumar et al. 1999, Gomes et al. 2004, Hajirasouliha et al. 2007]. Um trabalho de natureza similar foi conduzido por [Haraguchi e Ono 2015] para alguns *puzzles* relacionados com a completude de quadrados latinos.

2. Preliminares

Seja n um número natural, e seja $[n] = \{1, \dots, n\}$. Um grid quadrado A de n^2 células é representado como um matriz $n \times n$. Para $i, j \in [n]$, denotamos a célula na linha i e coluna j por (i, j), e o valor da célula (i, j) em A é dado por $A(i, j)$.

Um *quadrado latino* (LS – *latin square*) de ordem n é um par (L, S) onde L é uma matriz $n \times n$, $S \subset \mathbb{Z}$ é um conjunto de símbolos de cardinalidade n, e cada símbolo de S ocorre exatamente uma vez em cada linha e cada coluna de L. Um *quadrado latino parcial* (PLS – *partial latin square*) (L, S) é um LS exceto que algumas células de L podem estar vazias. Quando não ambíguo ($0 \notin S$), assumimos que $L(i, j) = 0$ quando a célula (i, j) está vazia. Quando $S = [n]$, omitimos S.

Considere um PLS L. Denotamos por $|L|$ o número de células não-vazias em L. L é *compatível* com outro PLS A se para todo $(i, j) \in [n]^2$ vale que $L(i, j) = 0$ ou $A(i, j) = 0$, e $L + A$ é um PLS. Quando L e A são compatíveis, denotamos $L + A$ como $L \oplus A$. Um PLS L' é dito ser uma *extensão* de L se para toda célula não-vazia (i, j) de L, vale que $L'(i, j) = L(i, j)$, isto é, L' pode ser obtido preenchendo algumas células vazias de L. Equivalentemente, existe um PLS A tal que $L' = L \oplus A$. Se L pode ser estendido para um LS, dizemos que L é *completável*. Dizemos que uma célula vazia (i, j) *admite* o símbolo k se L continua sendo um PLS após atribuir $L(i, j) = k$. Se uma célula vazia (i, j) não admite nenhum símbolo, dizemos que (i, j) está *bloqueada*, e caso todas as células vazias de L estejam bloqueadas, dizemos que L está *bloqueado*. O problema de estender um PLS é definido a seguir.

Problema 1 (*Partial Latin Square Extension* – `PLSE`)**.** Dado um PLS L, encontre uma extensão $L \oplus A$ tal que $|A|$ seja o maior possível.

Dada uma matriz A de tamanho $n \times n$, chamamos de *diagonal principal* de A as células $\{(i, i) : i \in [n]\}$, e de *diagonal secundária* as células $\{(i, n + 1 - i) : i \in [n]\}$. Um *quadrado latino diagonal parcial* (PDLS – *partial diagonal latin square*) é um PLS com a restrição adicional de que cada símbolo ocorre no máximo uma vez nas diagonais principal e secundária, e um PDLS com nenhuma célula vazia é chamado de *quadrado latino diagonal* (DLS – *diagonal latin square*). Definimos PDLSs compatíveis e extensões de um PDLS de forma análoga à definida para PLSs. O problema de estender um PDLS também é similarmente definido a seguir.

Problema 2 (*Partial Diagonal Latin Square Extension* – `PDLSE`)**.** Dado um PDLS L, encontre uma extensão $L \oplus A$ tal que $|A|$ seja o maior possível.

3. Complexidade de `PDLSE`

Um problema intrinsecamente relacionado com o `PLSE` é o problema de partição em triângulos de grafos tripartidos, o qual denotaremos por `3EDM`. Nesse problema, é dado um grafo tripartido G, e deseja-se encontrar o maior número de triângulos (cliques de tamanho 3) aresta-disjuntos. [Holyer 1981] provou que é NP-completo decidir se as arestas de um grafo podem ser particionadas em subgrafos isomorfos ao grafo completo K_n para $n \geq 3$. [Colbourn 1984] mostrou que a prova de Holyer continua válida para $n = 3$ quando o grafo é tripartido, provando então que a versão de decisão de `3EDM` é NP-completo. Ademais, através da redução desse problema para a versão de decisão de `PLSE` (decidir se um PLS é completável), Colbourn provou que o último também é

NP-completo. [Hajirasouliha et al. 2007] estenderam ainda mais a prova de Holyer para provar que `3EDM` é APX-difícil. Além disso, eles apresentaram reduções polinomiais entre `3EDM` e `PLSE` que preservam valor, demonstrando assim que os problemas são equivalentes. Com isso, os autores concluíram que `PLSE` também é APX-difícil.

É possível utilizar uma abordagem similar a de [Hajirasouliha et al. 2007] para mostrar que `PDLSE` também é APX-difícil. Para tal, podemos definir uma versão mais generalizada do problema `3EDM`, em que algumas arestas podem admitir cores, e uma solução só é viável se não houver mais de um triângulo com uma cor em comum. Tais cores são definidas de forma a fazer uma correspondência com células que dividem uma mesma diagonal, e assim dado uma instância dessa versão mais generalizada do problema `3EDM`, é possível construir um PDLS de forma que o número máximo de células que podem ser preenchidas é igual ao número máximo de triângulos que podem ser escolhidos, o que implica na APX-dificuldade de `PDLSE`.

Teorema 1. *`PDLSE` é APX-difícil.*

4. Algoritmos de Aproximação para PDLSE

Nesta seção, apresentamos alguns algoritmos de aproximação originalmente propostos para o `PLSE` e os estendemos para o `PDLSE`. Consideramos três tipos de algoritmos: um algoritmo guloso, um algoritmo baseado em emparelhamento, e uma busca local.

4.1. Algoritmo Guloso

Um algoritmo guloso para o `PLSE` consiste em iterativamente atribuir a uma célula (i, j) um símbolo k que é válido para o PLS, até que a solução se torne bloqueada. Em outras palavras, o algoritmo guloso consiste na obtenção de uma solução maximal. [Kumar et al. 1999] mostraram que tal algoritmo é uma $1/3$-aproximação para o `PLSE`.

Teorema 2 ([Kumar et al. 1999])**.** *Para qualquer instância do `PLSE`, uma solução bloqueada é uma solução $1/3$-aproximada.*

Esse resultado advém da análise do quanto a escolha de uma atribuição pode impactar negativamente uma solução ótima. Para o caso do `PLSE`, a atribuição do símbolo k em uma célula (i, j) pode impedir a atribuição de até 3 células em uma solução ótima, que são a própria célula (i, j), mais alguma célula na linha i e outra na coluna j que tenham sido preenchidas com o símbolo k. Para o `PDLSE`, esse número aumenta para 5, já que também pode haver uma célula na mesma diagonal principal e diagonal secundária com o valor k em uma solução ótima, e isso resulta em uma $1/5$-aproximação.

Teorema 3. *Para qualquer instância do `PDLSE`, uma solução bloqueada é uma solução $1/5$-aproximada.*

No entanto, note que a única possibilidade em que uma atribuição pode impactar negativamente em 5 células de uma solução ótima é na célula central de um PLS de ordem ímpar. Como se trata de apenas uma célula, podemos tentar atribuir todos os possíveis valores para essa célula, ou mantê-la vazia, e para cada possibilidade aplicar o algoritmo guloso. Com isso, garantimos um fator de aproximação de $1/4$.

Teorema 4. *Existe uma $1/4$-aproximação para `PDLSE`.*

4.2. Algoritmo Baseado em Emparelhamento

[Kumar et al. 1999] também apresentaram um algoritmo para o PLSE baseado em emparelhamento, o qual chamaremos de MATCHING. O algoritmo consiste em iterativamente encontrar uma atribuição máxima para cada símbolo. Para encontrar uma atribuição de um símbolo k, construímos um grafo bipartido $G = (R \cup C, E)$ com $R = \{r_1, \ldots, r_n\}$, $C = \{c_1, \ldots, c_n\}$ e $\{r_i, c_j\} \in E$ se e somente se a célula (i, j) admite k. Note que um emparelhamento em G representa um conjunto de atribuições válidas para k no PLS. Assim, encontramos um emparelhamento máximo M^k e atribuímos k nas células que correspondem a M^k. [Kumar et al. 1999] mostraram que MATCHING é uma $1/2$-aproximação.

Para o problema PDLSE, um emparelhamento em G como descrito anteriormente não necessariamente resulta em uma atribuição válida para o PDLS, pois as restrições impostas nas diagonais podem não ser respeitadas. No entanto, é possível generalizar o problema de emparelhamento adicionando cores às arestas de forma que tais restrições sejam respeitadas, e ainda encontrar tal emparelhamento em tempo polinomial. Assim, restringindo a busca para emparelhamentos máximos que respeitam tais cores no algoritmo MATCHING, o fator de aproximação se estende para PDLSE.

Teorema 5. *Existe uma $1/2$-aproximação para PDLSE.*

4.3. Busca Local

Seja $k \geq 3$ um inteiro. No problema *k-set packing* (k−SP), é dado um conjunto de elementos $\mathcal{U}$ e uma família $\mathcal{F}$ de subconjuntos de $\mathcal{U}$, onde cada conjunto $F \in \mathcal{F}$ possui tamanho no máximo k. Uma subfamília $\mathcal{F}' \subseteq \mathcal{F}$ é dita ser um *packing* se os conjuntos de $\mathcal{F}'$ são disjuntos entre si. O objetivo do problema é encontrar um *packing* de $\mathcal{F}$ de cardinalidade máxima.

O problema k−SP é NP-difícil [Garey e Johnson 1979] e os melhores algoritmos de aproximação para o problema são baseados em busca local. Os melhores resultados atualmente são os de [Cygan 2013] e [Fürer e Yu 2014] que propõem algoritmos com fator de aproximação $3/(k + 1) - \varepsilon$ para qualquer $\varepsilon > 0$.

O problema PLSE pode ser reduzido para 3−SP, o que implica em uma $(3/4 - \varepsilon)$-aproximação para PLSE. Ademais, é possível adaptar tal redução para que PDLSE também seja redutível para 3−SP, fornecendo assim o mesmo fator de aproximação.

Teorema 6. *Para qualquer $\varepsilon > 0$, existe uma $(3/4 - \varepsilon)$-aproximação para PDLSE.*

5. Conclusão

Apresentamos algoritmos de aproximação para o problema de extensão de quadrados latinos diagonais – PDLSE. Dentre os algoritmos apresentados, o baseado em busca local fornece o melhor fator de aproximação, $3/4 - \varepsilon$ para qualquer $\varepsilon > 0$. Tal fator também é o melhor conhecido atualmente para o problema de extensão de quadrados latinos – PLSE. Ambos os problemas também são APX-difíceis. Tais resultados podem sugerir que PDLSE, embora seja um caso particular de PLSE, ainda seja tão difícil quanto PLSE em termos de aproximação. Além disso, não se sabe se tal fator é o melhor possível para PLSE. No melhor de nosso conhecimento, o melhor fator de inaproximabilidade para PLSE é de $94/95$ [Chlebík e Chlebíková 2006].

Agradecimentos. Agradecemos aos revisores anônimos pelos comentários e sugestões. O presente trabalho foi realizado com apoio do CNPq (Proc. 163645/2021-3, 313146/2022-5) e da FAPESP (Proc. 2015/11937-9, 2016/01860-1).

Referências

Chlebík, M. e Chlebíková, J. (2006). Complexity of approximating bounded variants of optimization problems. *Theoretical Computer Science*, 354(3):320–338.

Colbourn, C. J. (1984). The complexity of completing partial latin squares. *Discrete Applied Mathematics*, 8(1):25–30.

Cygan, M. (2013). Improved approximation for 3-dimensional matching via bounded pathwidth local search. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 509–518, Los Alamitos, CA, USA. IEEE Computer Society.

Fürer, M. e Yu, H. (2014). Approximating the k-set packing problem by local improvements. In Fouilhoux, P., Gouveia, L. E. N., Mahjoub, A. R., e Paschos, V. T., editors, *Combinatorial Optimization*, pages 408–420, Cham. Springer International Publishing.

Garey, M. R. e Johnson, D. S. (1979). *Computers and intractability*, volume 174. freeman San Francisco.

Gomes, C. P., Regis, R. G., e Shmoys, D. B. (2004). An improved approximation algorithm for the partial latin square extension problem. *Operations Research Letters*, 32(5):479–484.

Hajirasouliha, I., Jowhari, H., Kumar, R., e Sundaram, R. (2007). On completing latin squares. In Thomas, W. e Weil, P., editors, *STACS 2007*, pages 524–535, Berlin, Heidelberg. Springer Berlin Heidelberg.

Haraguchi, K. e Ono, H. (2015). How simple algorithms can solve latin square completion-type puzzles approximately. *Journal of Information Processing*, 23(3):276–283.

Hilton, A. J. W. (1973). On double diagonal and cross latin squares. *Journal of the London Mathematical Society*, s2-6(4):679–689.

Hilton, A. J. W. (1974). Some simple constructions for double diagonal latin squares. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 215–229.

Holyer, I. (1981). The NP-completeness of some edge-partition problems. *SIAM Journal on Computing*, 10(4):713–717.

Keedwell, A. D. e Dénes, J. (2015). *Latin squares and their applications*. Elsevier.

Kumar, S. R., Russell, A., e Sundaram, R. (1999). Approximating latin square extensions. *Algorithmica*, 24(2):128–138.