

Um Algoritmo Paralelo para o Problema de Caminhos Hamiltonianos em Grafos Kneser

Andréia C. S. Gusmão, Leticia R. Bueno, Rodrigo A. Hausen

CMCC, Universidade Federal do ABC (UFABC)

Santo André – SP – Brasil

andrea.gusmao@ufabc.edu.br, leticia.bueno@ufabc.edu.br, hausen@compscinet.org

RESUMO

O grafo Kneser $K(n, k)$ tem como vértices os subconjuntos de cardinalidade k do conjunto $\{1, 2, \dots, n\}$. Dois vértices são adjacentes se os subconjuntos correspondentes são disjuntos. O grafo Kneser bipartido $B(n, k)$ tem como vértices os subconjuntos de cardinalidade k e os subconjuntos de cardinalidade $(n - k)$ de $\{1, 2, \dots, n\}$. Dois vértices são adjacentes se um é subconjunto do outro. Foi provado que um caminho hamiltoniano particular em um grafo reduzido de $K(2k + 1, k)$ leva a um caminho hamiltoniano em $K(2k + 1, k)$ e a um ciclo hamiltoniano em $B(2k + 1, k)$. Propomos um algoritmo paralelo para a busca destes caminhos hamiltonianos particulares nos grafos reduzidos de $K(2k + 1, k)$.

PALAVRAS-CHAVE. caminhos hamiltonianos, algoritmos paralelos, grafos Kneser.

Área Principal: TAG - Teoria e Algoritmos em Grafos

ABSTRACT

The Kneser graph $K(n, k)$ has as vertices the k -subsets of the set $\{1, 2, \dots, n\}$. Two vertices are adjacent if the corresponding subsets are disjoint. The bipartite Kneser graph $B(n, k)$ has as vertices the k -subsets and the $(n - k)$ -subsets of $\{1, 2, \dots, n\}$. Two vertices are adjacent if one is a subset of the other. It was proved that a particular hamiltonian path in a reduced graph of $K(2k + 1, k)$ gives a hamiltonian path in $K(2k + 1, k)$ and a hamiltonian cycle in $B(2k + 1, k)$. We propose a parallel algorithm to search for that particular hamiltonian path in the reduced graphs of $K(2k + 1, k)$.

KEYWORDS. hamiltonian paths, parallel algorithms, Kneser graphs.

Main area: TAG - Graph Theory and Algorithms in Graphs

1. Introdução

O Problema do Ciclo Hamiltoniano consiste em determinar se um grafo G tem um ciclo que passa por todos os vértices do grafo exatamente uma vez. O Problema do Caminho Hamiltoniano consiste em determinar se um grafo G tem um caminho que passa por todos os vértices do grafo exatamente uma vez. É conhecido que o problema de decisão relacionado a ambos os problemas é NP-Completo [Karp(1972), Garey e Johnson(1979)].

Sejam n, k inteiros tais que $1 \leq k \leq n$. O grafo Kneser, denotado por $K(n, k)$, tem como vértices os subconjuntos de cardinalidade k tomados de $\{1, 2, \dots, n\}$. Dois vértices são adjacentes se os subconjuntos correspondentes são disjuntos. Para $n = 2k + 1$, o grafo Kneser $K(2k + 1, k)$ é também chamado o grafo ímpar, denotado por O_k , para $k \geq 1$ (Figura 1(a)).

O grafo Kneser bipartido, denotado por $B(n, k)$, tem $\binom{n}{k} \cup \binom{n}{n-k}$ como seu conjunto de vértices, e dois vértices são adjacentes se um é subconjunto do outro. Suas arestas

representam a inclusão entre dois subconjuntos, isto é, $A \subsetneq B$ onde A é um k -subconjunto e B é um $(n - k)$ -subconjunto. Para $n = 2k + 1$, o grafo Kneser bipartido é chamado o *grafo dos níveis intermediários*, denotado por B_k (Figura 1(b) e Figura 1(c)).

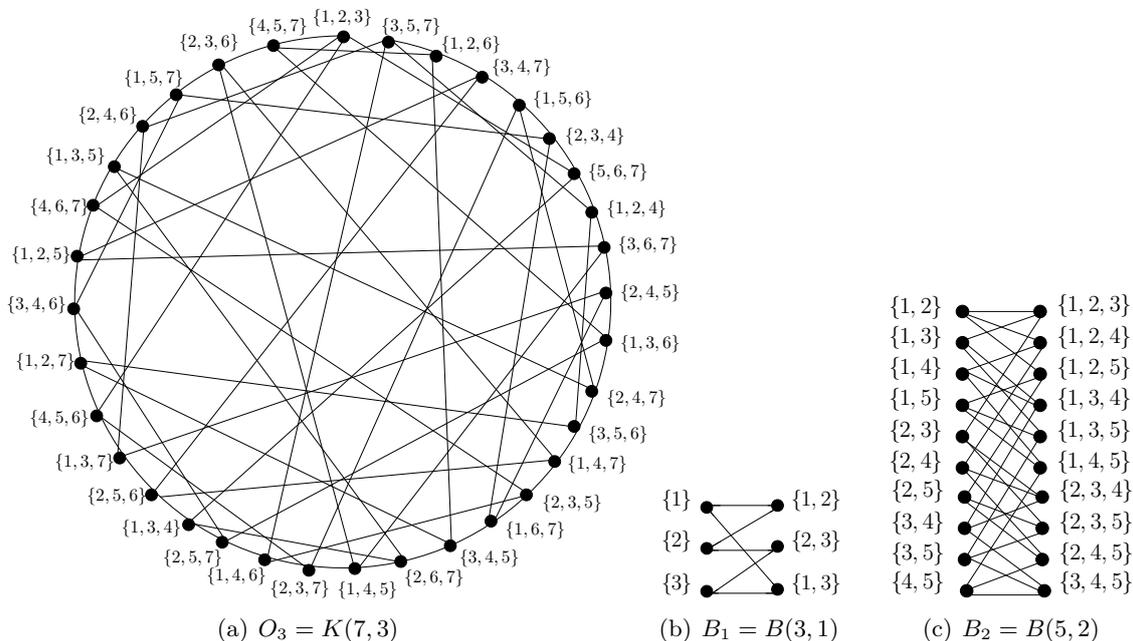


Figura 1: O grafo ímpar O_3 e os grafos Kneser bipartidos B_1 e B_2 .

Determinar ciclos e caminhos hamiltonianos nos grafos Kneser e Kneser bipartidos é um problema muito estudado. No entanto, o grafo $K(n, k)$ tem $\binom{n}{k}$ vértices e $\binom{n}{k} \binom{n-k}{k} / 2$ arestas, enquanto o grafo Kneser bipartido $B(n, k)$ tem $2 \binom{n}{k}$ vértices e $\binom{n}{k} \binom{n-k}{k}$ arestas (Tabela 1). Portanto, uma computação exata de ciclos e caminhos hamiltonianos nestes grafos não é viável para grandes valores de k .

| k | n | $ V(O_k) = \binom{2k+1}{k}$ | $ V(B_k) = 2 \binom{2k+1}{k}$ | $ V(\widetilde{O}_k) = \binom{2k+1}{k} / n$ |
|-----|-----|------------------------------|--------------------------------|--|
| 1 | 3 | 3 | 6 | 1 |
| 2 | 5 | 10 | 20 | 2 |
| 3 | 7 | 35 | 70 | 5 |
| 4 | 9 | 126 | 252 | 14 |
| 5 | 11 | 462 | 924 | 42 |
| 6 | 13 | 1.716 | 3.432 | 132 |
| 7 | 15 | 6.435 | 12.870 | 439 |
| 8 | 17 | 24.310 | 48.620 | 1.430 |
| 9 | 19 | 92.378 | 184.756 | 4.862 |
| 10 | 21 | 352.716 | 705.432 | 16.796 |
| 11 | 23 | 1.352.078 | 2.704.156 | 58.786 |
| 12 | 25 | 5.200.300 | 10.400.600 | 208.012 |
| 13 | 27 | 20.058.300 | 40.116.600 | 742.900 |
| 14 | 29 | 77.558.760 | 155.117.520 | 2.674.440 |
| 15 | 31 | 300.540.195 | 601.080.390 | 9.694.845 |
| 16 | 33 | 1.166.803.110 | 2.333.606.220 | 35.357.670 |
| 17 | 35 | 4.537.567.650 | 9.075.135.300 | 129.644.790 |
| 18 | 37 | 17.672.631.900 | 35.345.263.800 | 477.638.700 |

Tabela 1: Número de vértices de O_k , B_k e $\widetilde{O}_k = \widetilde{B}_k$, para $k \leq 18$.

[Chen(2003)] mostrou que o grafo $K(n, k)$ é hamiltoniano para $n \geq 2.62k + 1$ e

[Shields e Savage(2004)] mostraram que os grafos $K(n, k)$ e $B(n, k)$ são hamiltonianos para $n \leq 27$, com exceção do grafo O_2 que é o grafo de Petersen – grafo reconhecidamente não hamiltoniano. Os grafos Kneser e Kneser bipartidos são conexos somente para $n > 2k$ e os grafos mais esparsos de ambas as classes são os grafos O_k e B_k , sendo assim os grafos mais estudados em relação ao problema de ciclos e caminhos hamiltonianos. Para O_k e B_k , até agora foram determinados ciclos hamiltonianos em O_k , para $k \leq 13$ [Shields e Savage(2004)], e em B_k , para $k \leq 17$ [Shields e Savage(1999), Shields et al.(2009)]. Um caminho hamiltoniano foi determinado para O_k , para $k \leq 17$ [Bueno et al.(2009)]. A Tabela 2 mostra os resultados conhecidos para os grafos Kneser e Kneser bipartidos.

Um grafo reduzido sobre o grafo B_k , que denotamos por \widetilde{B}_k , foi proposto por [Dejter(1985), Reid(1990)]. Posteriormente, [Bueno et al.(2009)] mostraram um grafo reduzido sobre o grafo O_k , denotado por \widetilde{O}_k , e provaram que $\widetilde{B}_k = \widetilde{O}_k$. Em [Bueno et al.(2009), Shields e Savage(1999), Shields et al.(2009)], os autores provaram que um caminho hamiltoniano particular – o chamaremos de um *caminho viável* – em \widetilde{O}_k fornece um ciclo e um caminho hamiltonianos em B_k e O_k , respectivamente. Usando heurísticas, [Shields e Savage(1999), Shields et al.(2009)] determinaram caminhos viáveis em \widetilde{O}_k para $1 \leq k \leq 17$. Observe que o número de vértices de \widetilde{O}_k é menor que o número de vértices de O_k em um fator de n (Tabela 1).

Neste trabalho, propomos um algoritmo paralelo para a busca de caminhos viáveis no grafo reduzido \widetilde{O}_k . Mostramos que as coberturas de vértices por ciclos disjuntos em O_k fornecidas em [Johnson e Kierstead(2004)] são também coberturas de vértices por ciclos disjuntos em \widetilde{O}_k . Utilizamos um algoritmo paralelo para concatenar os ciclos da cobertura em um único caminho. Finalmente, utilizamos o algoritmo de [Shields e Savage(1999)] para adicionar ao caminho os vértices eventualmente perdidos durante a concatenação realizada pelo algoritmo paralelo.

Na Seção 2., discutimos algumas definições usadas em nosso algoritmo, que é apresentado na Seção 3. Resultados computacionais são discutidos na Seção 4. Finalmente, algumas conclusões e trabalhos futuros são apresentados na Seção 5.

2. Preliminares

Nesta seção, apresentamos algumas definições que serão utilizadas no algoritmo proposto.

2.1. O Grafo Reduzido Sobre B_k e O_k

Utilizamos a operação de redução para os grafos B_k e O_k de [Dejter(1985), Reid(1990), Bueno et al.(2009)]. Seja \mathbb{Z}_n o conjunto $\{1, \dots, n\}$ com aritmética módulo n . Consideramos cada vértice de $K(n, k)$ e $B(n, k)$ como um subconjunto de \mathbb{Z}_n . Definimos dois k -subconjuntos particulares de \mathbb{Z}_n que são $r_1 = \{1, \dots, k\}$ e $r_2 = \{2, 4, 6, \dots, n-1\}$.

Dado um conjunto $v = \{v_1, v_2, \dots, v_t\} \subseteq \mathbb{Z}_n$ e $\delta \in \mathbb{Z}_n$, denote $\bar{v} = \mathbb{Z}_n \setminus v$ o *complemento* de v em relação a \mathbb{Z}_n e denote $v + \delta = \{v_1 + \delta, v_2 + \delta, \dots, v_t + \delta\}$. Dizemos que $u, v \subseteq \mathbb{Z}_n$ satisfazem $u \sim v$ se (i) $u = v + \delta$ ou (ii) $\bar{u} = v + \delta$ para algum $\delta \in \mathbb{Z}_n$. Observe que \sim é uma relação de equivalência. Nos referimos à classe de equivalência de v definida por \sim como $\sigma(v)$.

Dado um grafo G , definimos o *grafo quociente* \widetilde{G} como o grafo obtido de G por identificar os vértices que são equivalentes de acordo com \sim . Desta forma, os vértices de \widetilde{G} são as classes de equivalência $\sigma(v)$ para $v \in V(G)$ e se $\{u, v\} \in E(G)$, então $\sigma(u)\sigma(v) \in E(\widetilde{G})$ (Figura 2). O grafo quociente \widetilde{G} é chamado o *grafo reduzido*. Foi provado que:

Tabela 2: Ciclos e/ou caminhos hamiltonianos nos grafos Kneser e Kneser bipartidos

| Grafos Kneser | | | | | | | | |
|---------------|-------------|----------|----------|----------|----------|----------|----------|----------|
| k | n | | | | | | | |
| | $2k + 1$ | $2k + 2$ | $2k + 3$ | $2k + 4$ | $2k + 5$ | $2k + 6$ | $2k + 7$ | $2k + 8$ |
| 1 | 3^h | 4^h | 5^h | 6^h | 7^h | 8^h | 9^h | 10^h |
| 2 | 5^P | 6^h | 7^h | 8^h | 9^h | 10^h | 11^h | 12^h |
| 3 | $7^{a,h,l}$ | 8^h | 9^h | 10^h | 11^h | 12^h | 13^h | 14^h |
| 4 | $9^{a,l}$ | 10^S | 11^S | 12^S | 13 | 14 | 15 | 16 |
| 5 | $11^{l,S}$ | 12^S | 13^S | 14^S | 15^S | 16 | 17 | 18 |
| 6 | $13^{l,S}$ | 14^S | 15^S | 16^S | 17^S | 18^S | 19 | 20 |
| 7 | $15^{M,S}$ | 16^S | 17^S | 18^S | 19^S | 20^S | 21^S | 22 |
| 8 | 17^S | 18^S | 19^S | 20^S | 21^S | 22^S | 23^S | 24^S |
| 9 | 19^S | 20^S | 21^S | 22^S | 23^S | 24^S | 25^S | 26^S |
| 10 | 21^S | 22^S | 23^S | 24^S | 25^S | 26^S | 27^S | 28 |
| 11 | 23^S | 24^S | 25^S | 26^S | 27^S | 28 | 29 | 30 |
| 12 | 25^S | 26^S | 27^S | 28 | 29 | 30 | 31 | 32^S |
| 13 | 27^S | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
| 14 | 29^b | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| 15 | 31^b | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| 16 | 33^b | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 17 | 35^b | 36 | 37 | 38 | 39 | 40 | 41 | 42 |
| 18 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |

| Grafos Kneser Bipartidos | | | | | | | | |
|--------------------------|--------------|------------|----------|----------|----------|----------|----------|----------|
| k | n | | | | | | | |
| | $2k + 1$ | $2k + 2$ | $2k + 3$ | $2k + 4$ | $2k + 5$ | $2k + 6$ | $2k + 7$ | $2k + 8$ |
| 1 | $3^{D,R}$ | $4^{I,D}$ | 5^I | 6^I | 7^I | 8^I | 9^I | 10^I |
| 2 | $5^{D,R}$ | 6^D | 7^V | 8^I | 9^I | 10^I | 11^I | 12^I |
| 3 | $7^{D,R}$ | 8^D | 9^D | 10 | 11^U | 12^U | 13 | 14^U |
| 4 | $9^{D,R}$ | $10^{D,S}$ | 11^S | 12^S | 13 | 14 | 15 | 16 |
| 5 | $11^{D,R,S}$ | $12^{D,S}$ | 13^S | 14^S | 15^S | 16 | 17 | 18 |
| 6 | $13^{D,R}$ | 14^D | 15^S | 16^S | 17^S | 18 | 19 | 20 |
| 7 | $15^{D,R}$ | 16^D | 17^S | 18^S | 19^S | 20^S | 21 | 22 |
| 8 | $17^{D,R,V}$ | 18^S | 19^S | 20^S | 21^S | 22 | 23 | 24 |
| 9 | $19^{D,R,V}$ | 20^S | 21^S | 22^S | 23^S | 24^S | 25 | 26 |
| 10 | $21^{R,V}$ | 22^S | 23^S | 24^S | 25^S | 26^S | 27 | 28 |
| 11 | $23^{R,V}$ | 24^S | 25^S | 26^S | 27^S | 28 | 29 | 30 |
| 12 | 25^V | 26^S | 27^S | 28 | 29 | 30 | 31 | 32^S |
| 13 | 27^V | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
| 14 | 29^V | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
| 15 | 31^V | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| 16 | 33^V | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 17 | 35^V | 36 | 37 | 38 | 39 | 40 | 41 | 42 |
| 18 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |

- \blacksquare : [Chen(2003)]
- b : caminho ham. [Bueno et al.(2009)]
- a : [Balaban(1972)]
- h : [Heinrich e Wallis(1978)]
- l : [Meredith e Lloyd(1972)]
- S : [Shields e Savage(2004)]
- V : [Shields e Savage(1999), Shields et al.(2009)]
- D : [Dejter et al.(1988)]
- I : [Simpson(1991)]
- P : grafo de Petersen
- M : [Mather(1976)]
- R : [Reid(1990)]
- U : [Hurlbert(1994)]

Teorema 1 ([Shields e Savage(1999)], [Bueno et al.(2009)]). *Se existe um caminho hamiltoniano em \widetilde{O}_k que inicia em $\sigma(r_1)$ e termina em $\sigma(r_2)$ – o qual chamamos de caminho viável – então existe um ciclo hamiltoniano em B_k e um caminho hamiltoniano em O_k .*

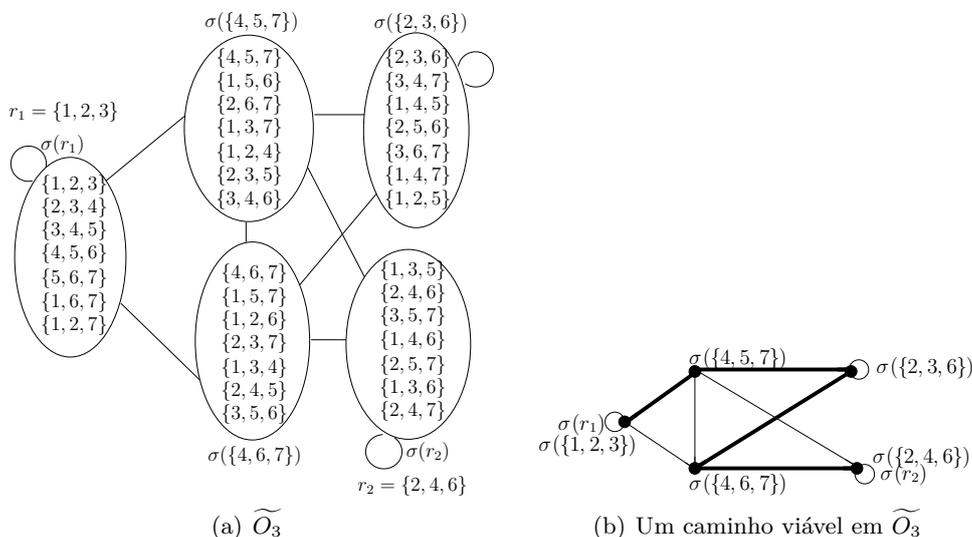


Figura 2: O grafo reduzido \widetilde{O}_3 e um caminho viável em \widetilde{O}_3 .

2.2. Representação por Strings Binárias

Existe uma correspondência biunívoca entre os k -subconjuntos de \mathbb{Z}_n e o conjunto de strings binárias de comprimento n com exatamente k 1's e $(n - k)$ 0's, o que simplifica a representação computacional e a manipulação dos grafos $K(n, k)$ e $B(n, k)$. A correspondência $b_n b_{n-1} \dots b_1 \leftrightarrow \{i | b_i = 1\}$ é uma bijeção das strings de n bits para os subconjuntos de \mathbb{Z}_n .

Considere os vértices de uma classe de equivalência representados por strings binárias tal que os vértices estejam ordenados lexicograficamente. O menor vértice da classe em termos lexicográficos é chamado o *representante da classe*. Usamos ao longo do texto o representante da classe para nos referir a um vértice do grafo reduzido (Figura 3).

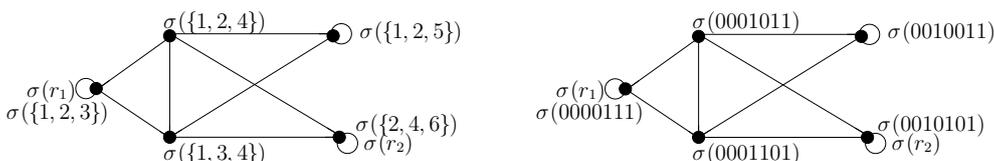


Figura 3: O grafo \widetilde{O}_3 com os vértices identificados pelo representante da classe.

Definimos o *complemento de um bit binário* x por: (i) $\bar{x} = 1$ se $x = 0$, e; (ii) $\bar{x} = 0$ se $x = 1$. Dada uma *string* binária de n bits $X = x_1 x_2 \dots x_n$, o *shift* de X é definido por $sh(X) = x_2 x_3 \dots x_n x_1$. Por exemplo, se $X = 00011$ então $sh(X) = 00110$, $sh^2(X) = 01100$, $sh^3(X) = 11000$ e assim por diante; trivialmente $sh^n(X) = X$.

2.3. Emparelhamentos Modulares

O conceito de *emparelhamento modular* foi introduzido em [Duffus et al.(1994)] para os grafos B_k . Seja A um k -subconjunto de \mathbb{Z}_n e $n = 2k + 1$. Então A é adjacente no emparelhamento m_i para $i = 1, 2, \dots, k + 1$, ao conjunto $m_i(A) = A \cup \{\bar{a}_j\}$, onde

$$j \equiv i + \sum_{a \in A} a \pmod{k + 1}$$

e \bar{a}_j é o j -ésimo maior elemento em \bar{A} .

[Duffus et al.(1994)] mostraram que os emparelhamentos modulares são emparelhamentos perfeitos que formam uma 1-fatorização de B_k . [Johnson e Kierstead(2004)] mostraram que os emparelhamentos modulares levam a uma 2-fatorização de O_k com $\frac{(k+1)}{2}$ 2-fatores para k ímpar. Para k par, os emparelhamentos modulares resultam em $\frac{k}{2}$ 2-fatores e um emparelhamento perfeito em O_k .

Um emparelhamento modular m_i em B_k é facilmente projetado no grafo O_k substituindo-se os conjuntos de tamanho $(k+1)$ pelo seu complemento. Assim, dado um emparelhamento modular m_i em B_k , denotamos por $\Pi(m_i)$, a projeção de m_i em O_k onde, para todo vértice A em $V(B_k)$, $\Pi(A) = A$ se A é um k -subconjunto e $\Pi(A) = \bar{A}$ se A é um $(k+1)$ -subconjunto.

Um 2-fator em um grafo G é uma cobertura de vértices por ciclos disjuntos e é, por consequência, um subgrafo gerador de G . Queremos fornecer ao algoritmo paralelo uma cobertura de vértices por ciclos disjuntos. Para isso, mostramos que as arestas de \widetilde{O}_k podem ser determinadas por $\Pi(m_i)$, para $i = 1, 2, \dots, k+1$. É suficiente mostrar que cada aresta de \widetilde{O}_k representa arestas de O_k de um mesmo $\Pi(m_i)$.

Lema 1. Cada aresta de \widetilde{O}_k representa n arestas de O_k do mesmo emparelhamento modular m_i , para $i = 1, 2, \dots, k+1$.

Demonstração. [Bueno et al.(2009)] demonstraram que uma aresta $\{\sigma(u), \sigma(v)\} \in E(\widetilde{O}_k)$ implica em um emparelhamento perfeito em O_k , formado pelos vértices de $\sigma(u)$ e os vértices de $\sigma(v)$ e pelas arestas $\{sh^t(u), sh^t(v)\}$, para $1 \leq t \leq n$. Sem perda de generalidade, considere uma aresta $\{A, \Pi(m_i(A))\} \in E(O_k)$. [Duffus et al.(1994)] mostraram que $m_i(sh^t(A)) = sh^t(m_i(A))$. Segue que todas as arestas do emparelhamento perfeito $\{sh^t(A), sh^t(\Pi(m_i(A)))\}$ estão em $\Pi(m_i)$. \square

3. O Algoritmo

Até o momento, os algoritmos implementados para procurar ciclos e caminhos hamiltonianos em grafos Kneser e Kneser bipartidos [Shields e Savage(1999), Shields e Savage(2004), Shields et al.(2009)] não utilizaram uma abordagem paralela. Fornecemos os ciclos de uma cobertura de vértices por ciclos disjuntos [Johnson e Kierstead(2004)] para um algoritmo que, paralelamente, os concatena em busca de um caminho viável. Pode ser, entretanto, que nem todos os vértices do grafo possam ser colocados no caminho retornado pelo algoritmo paralelo. Então utilizamos o algoritmo de [Shields e Savage(1999)] – que denotamos por SS99 – para adicionar adequadamente estes vértices ao caminho.

3.1. Representação Computacional da Solução e do Grafo

A solução do caminho viável é representada através de uma sequência de vértices em um vetor, onde cada posição do vetor equivale à posição do vértice no caminho. Para a representação do grafo \widetilde{O}_k , utilizamos um vetor para os representantes de classe em string binária – representados computacionalmente por um vetor de bits –, através do qual determinamos as adjacências entre os vértices em tempo quadrático em n (Algoritmo 1), o que dispensa a necessidade de armazenar as arestas em memória.

Considere $n = 2k+1$ e que cada vértice tem n bits dos quais k bits são 1 e os demais $(k+1)$ bits são 0. Para determinar se dois vértices $\sigma(u), \sigma(v) \in V(\widetilde{O}_k)$ são adjacentes, a cada iteração do Algoritmo 1, setamos 1 onde há um 0 em u (que tem k 1's), obtendo um vértice u' com $(k+1)$ 1's. Geramos o complemento de u' , obtendo uma adjacência de u em

Algoritmo 1: Algoritmo para determinar se dois vértices são adjacentes

Entrada: Vértices u e v

```

1 início
2   para  $i$  de 0 até  $(n - 1)$  faça
3      $aux \leftarrow u$ 
4     se  $aux[i] = 0$  então
5        $aux[i] \leftarrow 1$ 
6        $aux \leftarrow \text{complemento}(aux)$ 
7        $aux \leftarrow \text{representante\_classe}(aux)$ 
8       se  $aux = v$  então
9         retorna true
10      fim se
11    fim se
12  fim para
13  retorna false
14 fim

```

O_k . Determinamos o representante de classe de \bar{u}' em \widetilde{O}_k ordenando \bar{u}' lexicograficamente e, finalmente, comparamos se o representante de classe encontrado é o vértice v .

Obter o complemento de uma string binária (linha 6) tem tempo linear em n . Determinar o representante de classe de um vértice (linha 7) requer encontrar a maior sequência de zeros consecutivos e movê-los para a esquerda da string e tem tempo $O(n)$. Portanto, o Algoritmo 1 executa em tempo $O(n^2)$.

3.2. Concatenação dos Caminhos

No algoritmo paralelo (Algoritmo 2), cada processo recebe dois ou mais caminhos e retorna um caminho e uma lista de vértices que não puderam ser concatenados – aos quais chamamos *prejuízos*. Assim, embora uma cobertura de vértices por ciclos disjuntos em \widetilde{O}_k seja um conjunto de ciclos, estes são visualizados pelo algoritmo apenas como caminhos, desconsiderando-se a aresta entre o primeiro e último vértices.

O vértice $\sigma(r_2)$ deve ser o último vértice do caminho viável e, portanto, é removido do caminho em que se encontra e é armazenado na lista de prejuízos, para ser adicionado posteriormente. Além disso, se $\sigma(r_1)$ e $\sigma(r_2)$ estiverem no mesmo ciclo da cobertura de vértices por ciclos disjuntos, o ciclo é dividido em dois caminhos de forma que $\sigma(r_1)$ seja o primeiro vértice de um caminho e $\sigma(r_2)$ seja o primeiro vértice do outro caminho.

Seja np o número de processos $p = 0, \dots, (np-1)$ utilizados na execução do algoritmo paralelo. Seja nc o número de caminhos de uma cobertura em \widetilde{O}_k . Na distribuição dos caminhos entre os processos, cada processo recebe $\lfloor \frac{nc}{np} \rfloor$ caminhos e os demais caminhos são distribuídos um para cada dos primeiros processos.

Para a concatenação, cada processo escolhe dois caminhos P_1 e P_2 e retorna um único caminho P . Os dois caminhos P_1 e P_2 são excluídos do processo e o caminho P é adicionado. Essa operação é repetida até que cada processo tenha somente um caminho. O objetivo é minimizar o número de prejuízos, que denotamos por ρ . Sem perda de generalidade, assumamos que $|P_1| \leq |P_2|$.

O algoritmo usa uma variável α para controlar o número de tentativas de concatenação. Assim, no princípio, $\alpha = 0$, sendo incrementada a cada tentativa de concatenação, até que a concatenação seja bem-sucedida ou $\alpha = |P_1|$.

Denotamos por $W' \circ W''$ o caminho obtido por percorrer os vértices de um caminho W' seguidos pelos vértices de um caminho W'' . Dado um caminho W' , denotamos por $\overleftarrow{W'}$ o caminho atravessado do último para o primeiro vértice. Considere $P_1 = (v_1, v_2, \dots, v_q)$ e $P_2 = (w_1, w_2, \dots, w_m)$. Primeiramente, o algoritmo verifica se P_1 e P_2 podem ser concatenados nos extremos:

- se $\{v_1, w_1\} \in E(\widetilde{O}_k)$, então $P = \overleftarrow{P_1} \circ P_2$ (Figura 4(a));
- se $\{v_1, w_m\} \in E(\widetilde{O}_k)$, então $P = P_2 \circ P_1$ (Figura 4(b));
- se $\{w_1, v_q\} \in E(\widetilde{O}_k)$, então $P = P_1 \circ P_2$ (Figura 4(c));
- se $\{w_m, v_q\} \in E(\widetilde{O}_k)$, então $P = P_1 \circ \overleftarrow{P_2}$.

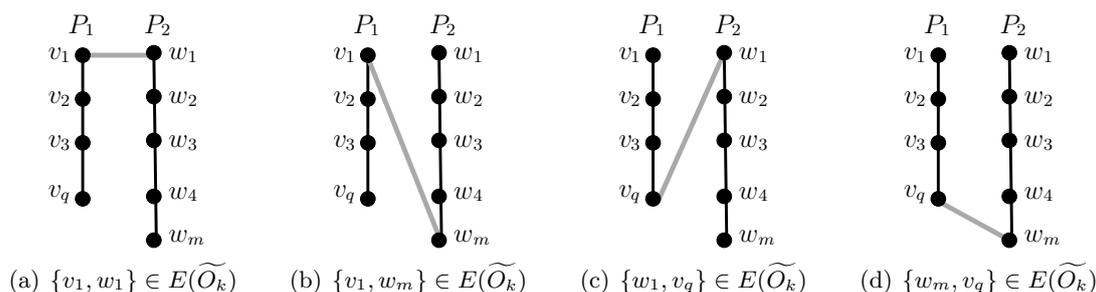


Figura 4: Possíveis concatenações de P_1 e P_2 com $\rho = \alpha = 0$.

Se não é possível concatenar P_1 e P_2 , o algoritmo verifica se P_1 pode ser colocado entre dois vértices w_i e w_j de P_2 , para $1 \leq i < j \leq m$, tal que $\{w_i, w_j\} \in E(\widetilde{O}_k)$, $\{v_1, w_i\} \in E(\widetilde{O}_k)$ e $\{v_q, w_j\} \in E(\widetilde{O}_k)$. Nesse caso, $P = (w_1, \dots, w_i) \circ P_1 \circ (w_j, \dots, w_m)$. Analogamente, o algoritmo verifica se P_2 pode ser colocado entre dois vértices de P_1 . Se não é possível, então, enquanto $\alpha < |P_1|$, o algoritmo verifica:

- se $\{v_1, w_{1+\alpha}\} \in E(\widetilde{O}_k)$, então $P = \overleftarrow{P_1} \circ (w_{1+\alpha}, \dots, w_m)$;
- se $\{v_1, w_{m-\alpha}\} \in E(\widetilde{O}_k)$, então $P = \overleftarrow{P_1} \circ (w_1, \dots, w_{m-\alpha})$;
- se $\{w_1, v_{q-\alpha}\} \in E(\widetilde{O}_k)$, então $P = (v_1, \dots, v_{q-\alpha}) \circ P_2$;
- se $\{w_1, v_{1+\alpha}\} \in E(\widetilde{O}_k)$, então $P = (v_{1+\alpha}, \dots, v_q) \circ P_2$.

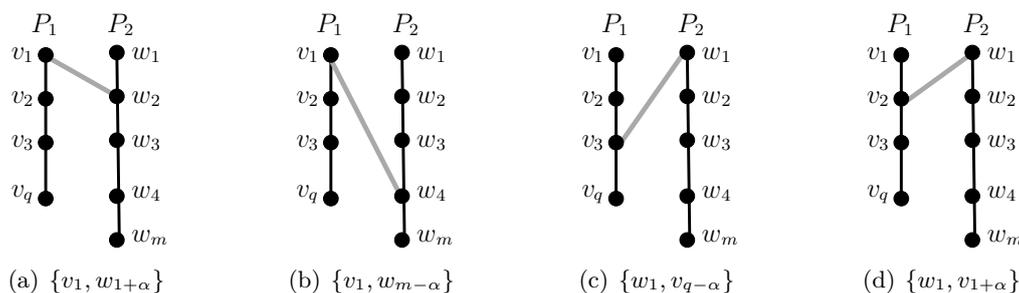


Figura 5: Concatenações possíveis de P_1 e P_2 com $\alpha = 1$.

A Figura 5 ilustra essas possibilidades para $\alpha = 1$. Como o vértice $\sigma(r_1)$ é o primeiro vértice do caminho viável, este não deve se tornar prejuízo. Assim, se $\sigma(r_1)$ está em P_1 ou P_2 , para qualquer valor de α , somente as concatenações que permitem $\sigma(r_1)$ na primeira posição de P são verificadas.

No melhor caso, o algoritmo concatena dois caminhos com $\rho = 0$. No pior caso, concatena com $\rho = (|P_1| - 1)$ ou não consegue concatenar o que faz $\rho = |P_1|$. Nesse caso, o processo retorna $P_2 = P$ ou $P_1 = P$ se $\sigma(r_1) \in P_1$.

Note ainda que os prejuízos podem ser caminhos. Por isso, quando um processo termina de concatenar todos os caminhos recebidos no início do algoritmo, em uma última tentativa procura minimizar os prejuízos concatenando os caminhos da lista de prejuízos ao caminho P .

Quando todos os processos terminam, uma barreira os sincroniza e, então os primeiros $\lfloor \frac{np}{2} \rfloor$ processos recebem o caminho retornado pelos últimos $\lfloor \frac{np}{2} \rfloor$ processos para concatenar com seu caminho obtido P . A cada iteração, o número de processos é reduzido pela metade, até que exista apenas um processo com um único caminho.

Algoritmo 2: Algoritmo paralelo para encontrar um caminho viável em \widetilde{O}_k

Entrada: Número de processos np e m caminhos da cobertura de vértices por ciclos disjuntos em \widetilde{O}_k

```

1 início
2   se processo = 0 então
3     distribui  $m$  caminhos para os  $np$  processos
4   fim se
5   se  $m > 1$  então
6     se processo  $\neq 0$  então
7       recebe os caminhos enviados pelo processo 0
8     fim se
9     enquanto houver mais de 1 caminho faça
10       $P \leftarrow \text{concatena}(P_1, P_2)$ 
11      exclui  $P_1$  e  $P_2$  do processo
12      insere  $P$  ao final dos caminhos do processo
13    fim enquanto
14    Barreira para sincronizar  $np$  processos
15    enquanto ( $\text{processo} < np$ ) e ( $np > 1$ ) faça
16       $q \leftarrow (np - 1) - \text{processo}$ 
17      se processo  $\geq \frac{np}{2}$  então
18        adiciona_prejuízos( $P$ )
19        envia  $P$  para o processo  $q$ 
20      senão
21         $W \leftarrow$  recebe  $P$  do processo  $q$ 
22         $P \leftarrow \text{concatena}(P, W)$ 
23        adiciona_prejuízos( $P$ )
24      fim se
25       $np \leftarrow np/2$ 
26    fim enquanto
27  fim se
28 fim

```

3.3. Adicionando os Prejuízos ao Caminho

Pela construção do Algoritmo 2, o primeiro vértice de P é $\sigma(r_1)$, e $\sigma(r_2)$ está na lista de prejuízos. Se $|P| = |V(\widetilde{O}_k)| - 1$ e o último vértice de P é adjacente a $\sigma(r_2)$, o vértice $\sigma(r_2)$ é adicionado ao final de P , que se torna um caminho viável de \widetilde{O}_k . Caso contrário, utilizamos o algoritmo SS99 para adicionar os prejuízos em P . A partir de um caminho P , o algoritmo

SS99 trabalha adicionando vértices ao final de P . Após a execução do algoritmo SS99, se $|P| = |V(\widetilde{O}_k)|$, então P é um caminho viável.

4. Resultados Computacionais

O algoritmo paralelo foi desenvolvido em C/MPI e executado em um supercomputador Altix 4700 com processador Itanium Intel 1.4GHz e 257GBytes de memória RAM. A Tabela 3 mostra um comparativo de tempo de execução entre o algoritmo proposto (GBH) e o algoritmo SS99 implementado. O tempo do algoritmo GBH consiste no tempo de execução do algoritmo paralelo mais o tempo de execução de SS99 para concatenar os prejuízos. A Tabela 4 mostra os resultados parciais obtidos até agora pelo algoritmo GBH para $15 \leq k \leq 18$.

Tabela 3: Tempo de execução dos algoritmos GBH e SS99 para \widetilde{O}_k . *: ainda em execução

| Tempo de Execução | | |
|-------------------|--------|--------|
| k | GBH | SS99 |
| 1 | 0s | 0s |
| 2 | 0s | 0s |
| 3 | 0s | 0s |
| 4 | 0s | 0s |
| 5 | 0s | 0s |
| 6 | 0s | 0s |
| 7 | 0s | 0s |
| 8 | 0s | 0s |
| 9 | 1s | 1s |
| 10 | 7s | 10s |
| 11 | 83s | 75s |
| 12 | 30min | 28min |
| 13 | 8h | 6, 30h |
| 14 | 3, 69d | 3, 68d |

Tabela 4: Caminhos parciais obtidos pelo algoritmo GBH para $15 \leq k \leq 18$

| k | $ V(\widetilde{O}_k) $ | $ P $ | $\% V(\widetilde{O}_k) $ em P | Tempo |
|-----|------------------------|-------------|---------------------------------|---------|
| 15 | 9.694.845 | 9.333.137 | 96% | 20d 12h |
| 16 | 35.357.670 | 28.662.709 | 81% | 20d 12h |
| 17 | 129.644.790 | 89.676.529 | 69% | 19d |
| 18 | 477.638.700 | 267.750.220 | 56% | 3d |

5. Conclusões e Trabalhos Futuros

Neste trabalho, propomos um algoritmo paralelo para procurar caminhos viáveis no grafo reduzido \widetilde{O}_k utilizando coberturas de vértices por ciclos disjuntos. Até agora determinamos caminhos viáveis para $k \leq 14$ e resultados parciais para $15 \leq k \leq 18$.

Um algoritmo para a geração de coberturas de vértices por ciclos disjuntos é executado apenas uma vez para cada cobertura, que é armazenada em arquivo e fornecida como entrada ao algoritmo paralelo. Assim, a geração de coberturas não onera o tempo de execução do algoritmo paralelo. Para o grafo \widetilde{O}_{16} , que tem mais de 35 milhões de vértices,

o algoritmo levou cerca de 28 minutos para retornar uma cobertura. Sobre a escolha de uma cobertura, dentre os $(k + 1)$ emparelhamentos modulares, constatamos que, em geral, as coberturas que favorecem o desempenho do algoritmo são aquelas determinadas pelo emparelhamento modular m_j onde $j \approx (k + 1)/4$. A razão parece ser o menor número de ciclos nessas coberturas, diminuindo assim o número necessário de concatenações de caminhos no algoritmo paralelo.

Na implementação do algoritmo SS99, para verificar se um vértice já está em um caminho, necessitamos encontrar o vértice na representação computacional do grafo. A princípio, a busca utilizada era sequencial. Utilizamos então o algoritmo de [Sawada(2003)] para gerar os vértices de \widetilde{O}_k e observamos que converter as strings binárias em números decimais fornece uma sequência em ordem crescente. Dessa maneira, podemos utilizar uma busca binária, reduzindo, assim, a complexidade de cada busca no pior caso de $O(m)$ para $O(\log m)$ onde $m = |V(\widetilde{O}_k)|$. Essa modificação resultou em um tempo de execução do algoritmo GBH de menos de 4 dias para determinar um caminho viável em \widetilde{O}_{14} , quando anteriormente o tempo necessário requereu 18 dias.

Como trabalho futuro, pretendemos otimizar o tempo de execução do algoritmo, modificando algumas estruturas de dados e operações relacionadas. Por exemplo, no algoritmo paralelo, pretendemos experimentar algumas escolhas aleatórias na concatenação dos caminhos que, no momento, é feito de forma exaustiva.

[Bueno et al.(2011)] provaram que $\widetilde{K}(n, k) = \widetilde{B}(n, k)$ e que um caminho viável no grafo reduzido de $K(2k + 2, k)$ leva a um ciclo hamiltoniano em $K(2k + 2, k)$ e $B(2k + 2, k)$, para k ímpar. Em um trabalho em andamento, estendemos esse resultado para um caminho viável em $\widetilde{K}(n, k)$ que leva a: (i) um caminho hamiltoniano em $K(2k + 3, k)$, para $k \equiv 1$ ou $2 \pmod{3}$ e (ii) um ciclo hamiltoniano em $K(2k + 4, k)$, para $k \equiv 1$ ou $3 \pmod{4}$. Além disso, determinamos uma cobertura de vértices por ciclos disjuntos também nos grafos reduzidos de $K(2k + 2, k)$, $K(2k + 3, k)$ e $K(2k + 4, k)$. Pretendemos modificar o algoritmo paralelo para ser aplicado a esses grafos. Até o momento, é conhecido que existem ciclos hamiltonianos para esses grafos somente até $k \leq 12$, $k \leq 12$ e $k \leq 11$, respectivamente.

Referências

- Balaban, A. T.** (1972). Chemical graphs, part XIII; combinatorial patterns. *Revue Roumaine de Mathématiques Pures et Appliquées*, 17:3–16.
- Bueno, L. R., Faria, L., Figueiredo, C. M. H., e Fonseca, G. D.** (2009). Hamiltonian paths in odd graphs. *Applicable Analysis and Discrete Mathematics*, 3(2):386–394.
- Bueno, L. R., Figueiredo, C. M. H., Faria, L., Mendonça, C. F. X., e Hausen, R. A.** (2011). Hamiltonian cycles in Kneser graphs for $n = 2k + 2$. *Electronic Notes in Discrete Mathematics*, 37:291–296. LAGOS'11 - VI Latin-American Algorithms, Graphs and Optimization Symposium.
- Chen, Y.-C.** (2003). Triangle-free hamiltonian Kneser graphs. *Journal of Combinatorial Theory Series B*, 89(1):1–16.
- Dejter, I. J.** (1985). Hamilton cycles and quotients of bipartite graphs. In *Graph theory with applications to algorithms and computer science (Kalamazoo, Mich., 1984)*, pages 189–199, Wiley, New York.
- Dejter, I. J., Cordova, J., e Quintana, J. A.** (1988). Two hamilton cycles in bipartite reflective Kneser graphs. *Discrete Mathematics*, 72(1):63–70.

- Duffus, D. A., Kierstead, H. A., e Snevily, H. S.** (1994). An explicit 1-factorization in the middle of the boolean lattice. *Journal of Combinatorial Theory, Series A*, 65:334–342.
- Garey, M. R. e Johnson, D. S.** (1979). *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York.
- Heinrich, K. e Wallis, W. D.** (1978). Hamiltonian cycles in certain graphs. *Journal of the Australian Mathematical Society Series A*, 26:89–98.
- Hurlbert, G.** (1994). The antipodal layers problem. *Discrete Mathematics*, 128:237–245.
- Johnson, J. R. e Kierstead, H. A.** (2004). Explicit 2-factorisations of the odd graph. *Order*, 21:19–27.
- Karp, R. M.** (1972). Reducibility among combinatorial problems. In Miller, R. e Thatcher, J., editors, *Complexity of Computer Computations*, pages 85–103, New York. Plenum Press.
- Mather, M.** (1976). The Rugby footballers of Croam. *Journal of Combinatorial Theory Series B*, 20(1):62–63.
- Meredith, G. H. J. e Lloyd, E. K.** (1972). The hamiltonian graphs O_4 to O_7 . In *Combinatorics (Proc. Conf. Combinatorial Math., Math. Inst., Oxford, 1972)*, pages 229–236. Inst. Math. Appl., Southend.
- Reid, M.** (1990). E-mail para J Gallian. Correspondência descrevendo trabalho de David Moews e Mike Reid no problema dos grafos intermediários.
- Sawada, J.** (2003). A fast algorithm to generate necklaces with fixed content. *Theoretical Computer Science*, 301(1–3):477–489.
- Shields, I. e Savage, C. D.** (1999). A Hamilton path heuristic with applications to the middle two levels problem. In *Proceedings of the Thirtieth Southeastern International Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, FL, 1999)*, volume 140, pages 161–178.
- Shields, I. e Savage, C. D.** (2004). A note on Hamilton cycles in Kneser graphs. *Bulletin of the Institute for Combinatorics and Its Applications*, 40:13–22.
- Shields, I., Shields, B. J., e Savage, C. D.** (2009). An update on the middle levels problem. *Discrete Mathematics*, 309(17):5271–5277.
- Simpson, J. E.** (1991). Hamiltonian bipartite graphs. In *Proceedings of the Twenty-second Southeastern Conference on Combinatorics, Graph Theory, and Computing (Baton Rouge, LA, 1991)*, volume 85, pages 97–110.