Algorithms for Hamiltonian Paths in Kneser Graphs*

Andréia Cristina dos Santos Gusmão[1]
Orientadora: Letícia Rodrigues Bueno[1]
Co-orientador: Rodrigo de Alencar Hausen[1]

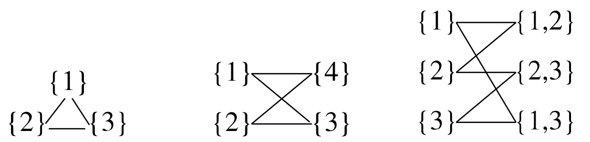
[1]Centro de Matemática, Computação e Cognição – CMCC
Universidade Federal do ABC (UFABC) – Santo André – SP – Brasil

deiasgusmao@gmail.com, leticia.bueno@ufabc.edu.br, hausen@compscinet.org

Abstract. *The Kneser graph $K(n,k)$ has as vertices the k-subsets of the set $\{1,\ldots,n\}$ and two vertices are adjacent if the k-subsets are disjoint. The bipartite Kneser graph $B(n,k)$ has as vertices the k and the $(n-k)$-subsets of $\{1,\ldots,n\}$ and 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 $B(2k+1,k)$ gives a hamiltonian path in $K(2k+1,k)$ and a hamiltonian cycle in $B(2k+1,k)$. We use a structural property in $K(2k+1,k)$ to devise a parallel algorithm and to improve a known algorithm, both to search for such a particular hamiltonian path in the reduced graph of $B(2k+1,k)$.*

1. Introduction

Let $\mathbb{Z}_n$ be the set $\{1,...,n\}$ and let $\binom{\mathbb{Z}_n}{k}$ be the family of all k-subsets of $\mathbb{Z}_n$. The *Kneser graph* $K(n,k)$ has $\binom{\mathbb{Z}_n}{k}$ as its vertex set, and two k-subsets are adjacent if they are disjoint (Figures 1(a) and 1(b)). The Kneser graph $K(2k+1,k)$ is called the *odd graph* and it is denoted by O_k (Figure 1(a)). Notice that O_2 is the well-known Petersen graph.



(a) $K(3,1) = O_1$ (b) $K(4,1)$ (c) $B(3,1) = B_1$

Figure 1. For $k=1$, the graphs: (a) O_k; (b) $K(2k+2,k)$, and; (c) B_k

The *bipartite Kneser graph* $B(n,k)$ has $\binom{\mathbb{Z}_n}{k} \cup \binom{\mathbb{Z}_n}{n-k}$ as its vertex set and an undirected edge $\{u,v\}$ represents that u either contains or is contained into v (Figure 1(c)). The graph $B(2k+1,k)$ is isomorphic to the subgraph of the $(2k+1)$-cube graph induced by the vertices having exactly k or $(k+1)$ ones. Hence, $B(2k+1,k)$ is also called the *middle levels graph*, and it is denoted by B_k.

[Lovász 1970] conjectured that every connected vertex-transitive graph has a hamiltonian path. For $n \geq 2k+1$, the graphs $K(n,k)$ and $B(n,k)$ form well-studied families of connected vertex-transitive graphs, both $\binom{n-k}{k}$-regular. However, a direct computation of hamiltonian paths or cycles in $K(n,k)$ and $B(n,k)$ is not feasible for large values of k, because $K(n,k)$ has $\binom{n}{k}$ vertices and $B(n,k)$ has $2\binom{n}{k}$ vertices. Apart from

*Research partially supported by Brazilian agencies CNPq and CAPES.

the Petersen graph, $K(n,k)$ and $B(n,k)$ are hamiltonian if $n \geq 2.62k+1$ [Chen 2003], or if $n \leq 27$ (see [Shields and Savage 2004] and references therein). [Johnson 2011] showed how to construct hamiltonian cycles in $K(2k+i,k)$ for $k \leq 2l+1$ using hamiltonian cycles from $K(2k+\frac{i}{2},k)$ for $k \leq l$. Since O_k is hamiltonian for $k \leq 13$, then $K(2k+2,k)$ is hamiltonian for $k \leq 27$ and, consequently, $K(2k+4,k)$ is hamiltonian for $k \leq 55$, and so on. Notice that O_k and B_k are the sparsest among all of the graphs $K(n,k)$ and $B(n,k)$, respectively. It was proved that B_k is hamiltonian for $k \leq 19$, and O_k has a hamiltonian path or a hamiltonian cycle for $k \leq 19$ [Bueno et al. 2009, Shields and Savage 1999, Shields and Savage 2004, Shields et al. 2009, Shimada and Amano 2011].

For a reduced graph of B_k and O_k, denoted by $R(B_k)$ and $R(O_k)$ respectively, it was proved that $R(B_k) = R(O_k)$ [Dejter 1985, Bueno et al. 2009]. Also, a particular hamiltonian path – we shall call it a *useful path* – in $R(B_k)$ provides a hamiltonian cycle and a hamiltonian path in B_k and O_k, respectively [Shields and Savage 1999, Bueno et al. 2009]. Later, it was proved that $R(K(n,k)) = R(B(n,k))$ and that a useful path in the reduced graph of $B(2k+2,k)$ implies a hamiltonian cycle in both graphs $B(2k+2,k)$ and $K(2k+2,k)$ [Bueno et al. 2011].

At the moment, all known useful paths have been determined by heuristic algorithms. First, [Shields and Savage 1999] proposed a heuristic based on Pósa's path reversal strategy [Pósa 1976] to search for useful paths in $R(B_k)$, and had its running time further improved in [Shields et al. 2009]. They determined useful paths for $k \leq 17$. Later, [Shimada and Amano 2011] partitioned the vertices of $R(B_k)$ such that the existence of a path in each set of vertices implies a useful path. They used the algorithm of [Shields and Savage 1999] to determine useful paths for $k = 18, 19$.

In our work, we have proved that a useful path in $R(K(2k+3,k))$ gives a hamiltonian path in $K(2k+3,k)$. By applying the algorithm in [Shields and Savage 1999] to the graph $R(K(2k+3,k))$, we have found two new results: a hamiltonian path in $K(29,13)$ and in $K(31,14)$. Also, we have used a property in O_k and B_k to develop a parallel algorithm [Gusmão et al. 2013a] and an improvement [Gusmão et al. 2013b] of the algorithm in [Shields and Savage 1999] to search for useful paths in $R(B_k)$. The second algorithm is faster than the algorithms in [Shields and Savage 1999, Shields et al. 2009].

The present text is meant to be a brief introduction to the basic ideas underlying the proofs and algorithms contained in the Master's thesis [Gusmão 2013] and in the papers [Gusmão et al. 2013a, Gusmão et al. 2013b]. Obviously, it does not delve too much into the details due to space constraints.

2. The Algorithms

Consider $\mathbb{Z}_n$ with arithmetical operations modulo n, and the vertices of $K(n = 2k+i, k)$ as k-subsets of $\mathbb{Z}_n$. Let $r_1 = \{1, 2, \ldots, k\}$ and $r_2 = \{2, 4, 6, \ldots, 2k = n - i\}$ be two k-subsets of $\mathbb{Z}_n$. Given a set $A \subseteq \mathbb{Z}_n$ and an integer $\delta \in \mathbb{Z}$, define $A + \delta$ as the set $\{a + \delta : a \in A\}$ and $\overline{A}$ as the set $\mathbb{Z}_n \setminus A$. Define the equivalence relation $\sim$ as follows: given $A, B \subset \mathbb{Z}_n$, $B \sim A$ if either: (i) $B = A + \delta$ or (ii) $\overline{B} = A + \delta$. Refer to the equivalence class of A under $\sim$ as $\sigma(A)$. The *reduced graph* $R(G)$ of a graph G has the equivalence classes $\sigma(v)$ as vertices, for each $v \in V(G)$, and if $\{u, v\} \in E(G)$ then $\{\sigma(u), \sigma(v)\} \in E(R(G))$.

We refer to a hamiltonian path starting with $\sigma(r_1)$ and ending with $\sigma(r_2)$ as a *useful* *path*. [Shields and Savage 1999] showed that the existence of a useful path in $R(B_k)$ implies that B_k is hamiltonian. A useful path in $R(K(n,k))$ which implies a hamiltonian cycle or path in the correspondent connected graphs $K(n,k)$ and/or $B(n,k)$ seems to be true whenever k and n are relatively prime. [Bueno et al. 2011, Bueno et al. 2009] proved this implication for $n = 2k+1$ and for $n = 2k+2$ if k is odd. We extended [Gusmão et al. 2013b] this result for $K(2k+3,k)$ and $k \equiv 1$ or $2 \pmod 3$.

A j-*factor* of a graph G is a j-regular spanning subgraph of G. For instance, an 1-*factor* is a perfect matching and a 2-factor is a covering of the vertices of the graph by disjoint cycles. A graph G is j-*factorable* if G is the union of disjoint j-factors. [Duffus et al. 1994] determined a 1-factorization of B_k hoping that the union of two suitable 1-factors would provide a hamiltonian cycle of B_k. Unfortunately, it turned out not to be the case for that 1-factorization. However, that 1-factorization were used to find a 2-factorization of O_k [Johnson and Kierstead 2004]. We showed that the 2-factorization in [Johnson and Kierstead 2004] gives a 2-factorization in $R(B_k) = R(O_k)$ as well (Lemma 2 in [Gusmão 2013]). This is the main property used by our algorithms.

2.1. The Parallel Algorithm

Searching for a useful path, the parallel algorithm tries to concatenate the cycles from a 2-factor in $R(B_k)$. At the end, if there are vertices that are not in the path, we use an implementation of the algorithm in [Shields and Savage 1999], henceforth denoted by SS99, to add the remaining vertices to the path.

The algorithm's idea is: repeatedly, each process receives two or more paths and return a single path and the vertices that could not be added – we call them *loss*. Since the vertex $\sigma(r_2)$ must be the last vertex of a useful path, it is a loss in order to be added only at the end. Naturally, the objective is minimizing the number of loss. However, the vertex $\sigma(r_1)$ is the first vertex of a useful path. Therefore, only concatenations with $\sigma(r_1)$ in the first position are verified. Notice that the loss can be paths, not only a single vertex.

At the end, the first vertex of P is $\sigma(r_1)$, and $\sigma(r_2)$ is a loss. If $|P| = |V(R(O_k))| - 1$ and the last vertex of P is adjacent to $\sigma(r_2)$, then P with $\sigma(r_2)$ is a useful path of $R(O_k)$. Otherwise, we use the algorithm SS99 to properly add the loss to P, since SS99 works adding vertices at the end of a path. The parallel algorithm in details is given in [Gusmão 2013, Gusmão et al. 2013a].

2.2. The Improved SS99 Algorithm

A backtracking search is guaranteed to find a hamiltonian path in $R(B_k)$ from a vertex $\sigma(r_1)$ to a vertex $\sigma(r_2)$ if such a path exists. However, it is an exhaustive search that runs in exponential time in the worst case and, therefore, impracticable except for relatively small graphs. [Pósa 1976] noticed that reaching a dead end in a backtrack search could be used as an opportunity to modify the current path by a *rotation*, and then possibly continuing. If $P = (v_1, v_2, \ldots, v_r)$ is a path in $R(B_k)$ and there is an edge $\{v_r, v_j\}$ for some $1 \leq j \leq r-1$, then a rotation at v_j is the path $P' = (v_1, v_2, \ldots, v_j, v_r, v_{r-1}, \ldots, v_{j+1})$ obtained by removing the edge $\{v_j, v_{j+1}\}$ and inserting the edge $\{v_j, v_r\}$. Pósa's path reversal strategy tries to extend a path P until no longer possible and avoiding the last vertex $\sigma(r_2)$ until the end. At this point, it selects a neighbor of the last vertex of P and performs a rotation.

The heuristic SS99 uses Pósa's strategy until all neighbors of the last vertex of P are already on the path. If the path is not a hamiltonian path, the algorithm performs a breadth-first search to search for a sequence of rotations resulting in a path that can be further extended. If no such sequence is found, it ends without finding a hamiltonian path.

Let m_l be a modular matching in B_k, for $l = 1, \ldots, k+1$, and let $\Pi(m_l)$ be the 2-factor in $R(B_k)$ constructed from m_l. We proposed a modification of Shields and Savage's algorithm which speeds up the construction of a useful path – this algorithm is denoted by MM. We choose one of the 2-factors in $R(B_k)$, say $\Pi(m_l)$ for $1 \leq l \leq k+1$ and, at the point the algorithm SS99 adds a vertex v in the path, we extend the cycle $C \in \Pi(m_l)$ containing v into a path P_v starting in v. Then, instead of adding only the vertex v, we add the whole path P_v to the useful path being constructed. If reversals are necessary, we proceed as in SS99. Determining the cycle which contains a vertex $v \in R(B_k)$ requires constant time, so the time complexity of the algorithm SS99 does not increase.

In general, the 2-factor $\Pi(m_l)$, for $l \approx \lceil \frac{k}{4} \rceil$, has the smallest number of cycles and, therefore, longer cycles (see Table 1 in [Gusmão et al. 2013b]). Since longer cycles collaborate to a faster running time of our algorithm, we have chosen the 2-factor $\Pi(m_l)$ of $R(B_k)$ with the smallest number of cycles as a starting point for the algorithm MM. For further details, we refer to [Gusmão et al. 2013b].

3. Experimental Results

The running times of the algorithms that search for useful paths in $R(B_k)$ are summarized in Table 1. Our implementation of SS99 is denoted by SS99*. The algorithms MM and SS99* were implemented in C++ and executed on a computer with a 3.20 GHz Intel(R) Core(TM) i5 processor, 4GBytes of RAM and 32-bit GNU/Linux operating system. [Shields et al. 2009] have improved the algorithm SS99, resulting in the algorithm denoted by SSS09 in this paper. The running times of the algorithms SS99 and SSS09 in Table 1 are presented in [Shields et al. 2009] and were obtained on a 2.4 GHz Intel Pentium 4 system with 512 MB of RAM. The parallel algorithm – denoted by Par – was implemented in C/MPI and executed on an Altix supercomputer 4700 Itanium Intel 1.4GHz processor and 257GBytes of RAM.

The parallel algorithm tries to concatenate the cycles by comparing almost all possibilities, that affected its running time. However, notice that, for $k = 17$, the time of execution of the algorithm MM is about 41 times faster than the algorithm SSS09, which cannot be attributed only to the difference in the hardware. In SS99*, our implementation of SS99, we made some code optimizations which made it slightly faster, so the difference between the times of SS99 and SS99* cannot be attributed only to the difference of system configuration. However, SS99* is still slower than SSS09. On the other hand, even though both MM and SS99* have been executed on the same computer, MM is noticeably faster. For $k = 14, 15$, the algorithm MM is more than 50 times faster than SS99* and, for $k = 16$, 118 times faster.

Since modular matchings are defined only for the graphs B_k and O_k, we have modified SS99* to search for useful paths in $R(K(2k+3, k))$. [Shields and Savage 2004] showed that $K(2k+3, k)$ is hamiltonian for $k \leq 12$. Our results [Gusmão et al. 2013b] show that $K(2k+3, k)$ has a hamiltonian path for $k = 13, 14$ as well, since we have found a useful path for $k \leq 14$ and $k \equiv 1$ or $2 \pmod 3$ (Table 3 in [Gusmão et al. 2013b]).

Table 1. Running time to find a useful path in $R(B_k)$

			Time (s)				
k	n	$\|V(R(B_k))\|$	SS99	SSS09	SS99*	MM	Par
8	17	1,430	0	0	0	0	0
9	19	4,862	0	0	0	0	1
10	21	16,796	1	0	0	0	7
11	23	58,786	5	2	2	1	83
12	25	208,012	105	10	18	2	1,800
13	27	742,900	1,732	99	180 (3m)	9	8h
14	29	2,674,440	24,138 (6,7h)	799 (13,31m)	2,460 (41m)	44	3,69d
15	31	9,694,845	307,976	9,446	36,450	692	–
			(3,5d)	(2,62h)	(10,12h)	(11,53m)	–
16	33	35,357,670	–	106,118	463,260	3,910	–
			–	(1,2d)	(5,04d)	(65,16m)	–
17	35	129,644,790	–	1,765,497	–	42,204	–
				(20,4d)		(11,72h)	–

4. Conclusion and Future Work

In the research leading to our Master's thesis research, we proved that a useful path in $R(K(2k+3,k))$ implies a hamiltonian path in $K(2k+3,k)$ for $k \equiv 1$ or $2 \pmod 3$ [Gusmão et al. 2013b]. By applying the algorithm in [Shields and Savage 1999] to the graph $R(K(2k+3,k))$, we found two new results: a hamiltonian path in $K(29,13)$ and in $K(31,14)$.

We also devised a parallel algorithm [Gusmão 2013, Gusmão et al. 2013a] to search for useful paths in the reduced graph $R(B_k)$. The algorithm tries to concatenate the cycles from a 2-factor in $R(B_k)$ in order to obtain a useful path. Using this algorithm, we have determined useful paths for $k \leq 14$.

For $n = 2k+1$, we improved the algorithm in [Shields and Savage 1999] by using the 2-factors in $R(B_k)$ [Gusmão et al. 2013b], which makes it faster than the algorithms in [Shields and Savage 1999, Shields et al. 2009].

Using the idea of our two proposed algorithms, it is feasible to devise an even faster algorithm for B_k. [Shields et al. 2009] have noticed that a large portion of the time of the algorithm in [Shields and Savage 1999] is used to perform rotation operations instead of finding promising sequences of rotations. They made some changes that significantly improved the running time of their previous algorithm. Since our algorithm is a modification of the algorithm in [Shields and Savage 1999], the modifications in [Shields et al. 2009] should improve the running time of our algorithm as well. Also, [Shimada and Amano 2011] proposed a strategy to partition the vertices of $R(B_k)$ into three sets, such that if there exists a particular path in each set, then a useful path in $R(B_k)$ can be constructed. This allows a parallel search to be performed on the graph. The authors adapted the algorithm in [Shields and Savage 1999] to search for that particular path in each set. Since the cycles in the modular matchings in $R(B_k)$ contain vertices of more than a set, our algorithm MM cannot run on each set separately. However, we are working in a new parallel algorithm, in which the cycles of a 2-factor in $R(B_k)$ are partitioned into several sets, and also on adapting our algorithm MM to run on each set separately.

References

Bueno, L., Figueiredo, C., Faria, L., Mendonça, C., and Hausen, R. (2011). Hamiltonian cycles in Kneser graphs for $n = 2k+2$. *Electron. Notes Discrete Math.*, 37(0):291–296. LAGOS'11 – VI Latin-American Algorithms, Graphs and Optimization Symposium.

Bueno, L. R., Faria, L., Figueiredo, C. M. H., and Fonseca, G. D. (2009). Hamiltonian paths in odd graphs. *Appl. Anal. Discrete Math.*, 3(2):386–394.

Chen, Y.-C. (2003). Triangle-free hamiltonian Kneser graphs. *J. Combin. Theory Ser. B*, 89(1):1–16.

Dejter, I. J. (1985). Graph theory with applications to algorithms and computer science. chapter Hamilton cycles and quotients of bipartite graphs, pages 189–199. John Wiley & Sons, Inc., New York, NY, USA.

Duffus, D. A., Kierstead, H. A., and Snevily, H. S. (1994). An explicit 1-factorization in the middle of the boolean lattice. *J. Combin. Theory Ser. A*, 65:334–342.

Gusmão, A. C. S. (2013). Um algoritmo paralelo para ciclos hamiltonianos em grafos Kneser. MsC thesis, Universidade Federal do ABC.

Gusmão, A. C. S., Bueno, L. R., and Hausen, R. A. (2013a). Um algoritmo paralelo para o problema de caminhos hamiltonianos em grafos Kneser. In *XLV Simpósio Brasileiro de Pesquisa Operacional (SBPO 2013), Natal-RN*, pages 3102–3113.

Gusmão, A. C. S., Bueno, L. R., Hausen, R. A., Figueiredo, C. M. H., and Faria, L. (2013b). A note on the middle levels problem. *Discrete Appl. Math.* Submitted in dez/2013. Preprint available in http://professor.ufabc.edu.br/~leticia.bueno/paper.zip.

Johnson, J. R. (2011). An inductive construction for hamilton cycles in Kneser graphs. *Electron. J. Combin.*, 18(1).

Johnson, J. R. and Kierstead, H. A. (2004). Explicit 2-factorisations of the odd graph. *Order*, 21:19–27.

Lovász, L. (1970). Problem 11. In *Combinatorial Structures and their Applications*. Gordon and Breach.

Pósa, L. (1976). Hamiltonian circuits in random graphs. *Discrete Math.*, 14(4):359 – 364.

Shields, I. and Savage, C. (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. and Savage, C. D. (2004). A note on Hamilton cycles in Kneser graphs. *Bull. Inst. Combin. Appl.*, 40:13–22.

Shields, I., Shields, B. J., and Savage, C. D. (2009). An update on the middle levels problem. *Discrete Math.*, 309(17):5271–5277.

Shimada, M. and Amano, K. (2011). A note on the middle levels conjecture. *CoRR*, abs/0912.4564.