A study of end-to-end Longest Path Problem with two missing vertices

National Dong Hwa University CSIE Department

Malaine Hydara, Seedy Gannes, Nasser Traore, W.E. Kevin Yanogo

June, 2017

Preliminary

Graphs

A graph is an abstract notation, used to model the idea of relations between objects. Graphs are commonly represented by a 2-tuple $G = (V, E)$, where V is a set of vertices (also called nodes) and E is a set of 2-element subsets of V called edges (or links), we note $E \subseteq \{\{u, v\} : u, v \in V\}$. [1]

Also, you can perceive E as a binary relation on V. In this case for non-directed graphs, the relation will be symmetric.

If each edge has a direction associated with it, the graph is called directed graph or Digraph. [2] In this case you cannot represent edges as 2-element sets, but rather as 2-tuples. Or represent E as a relation as mentioned above.

Graphs are usually depicted as points or circles connected by lines, and in a graph G if two vertices a and b are endpoints of an edge, we say that they are adjacent and write $a \sim b$. [3]

If vertex a is one of edge e's endpoints, a is incident to e and we write $a \in e$. For any vertex $v \in V$, we define the *degree* of v, denoted as $d(v)$, as the number of edges incident to v [4].

Graph theory is the area that aims to study the properties and applications of graphs. [6]

A lattice graph, also known as a mesh graph or grid graph, is a graph whose vertices are assigned 2D integer coordinates, and there are edges only between vertices with euclidian distance of 1 [7].

For a vertex v of a graph, let v_x and v_y. denote the x and y coordinates of its corresponding point.

Paths

A path graph P_n is a graph that can be drawn so that all vertices have degree $d(v) = 2$, excepted for the two endpoints with $d(v) = 1$.

A path is a sequence of vertices $v_1, v_2, ..., v_k$ such that v_{i-1} and v_i are adjacent for all i. A closed path (also called cycle) is a path where $v_1 = v_k$ [10].

A path is called simple if it does not have any repeated vertices. Therefore, a path $P = (v_1, v_2, ..., v_k)$ in G is a simple path if $v_i \neq v_j \ \forall \ i \neq j$ and $1 \leq i$, $j \leq k$. The length of a path may be measured by its number of edges [12].

If k is the largest possible in G, then P is a longest path. If $k = mn$, then the longest path P is also called Hamiltonian path.

Figure 1 gives an example for a Hamiltonian path on a rectangular grid graph.

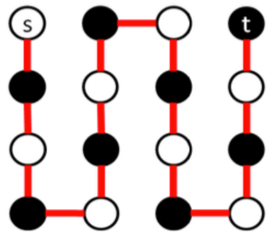

Figure 1: A Hamiltonian path for the rectangular grid graph $R(4, 4)$

Grid graphs

A two-dimensional grid graph, also known as a rectangular grid graph, is an $m \times n$ lattice graph $G_(m, n)$ that is the graph Cartesian product $P_m \times P_n$ of path graphs on m and n vertices [8].

In this article, our main focus will be the study of rectangular grid graphs with two holes. Note that a hole in a graph is a missing vertex [9].

A mesh is a rectangular grid graph $R(m, n)$ which is a finite solid grid graph where $V(R) = \{v = (v_x, v_y) | 1 \leq v_x \leq m, 1 \leq v_y \leq n\}$.

We color the vertices of the rectangular gird graph as black and white. A vertex v is colored white if $v_x + v_y$ is even. Otherwise, it is colored black.

We assume the vertex in the upper left corner has the coordinate $(1, 1)$. The number of vertices of $R(m, n)$ is mn. If mn is odd, then $R(m, n)$ odd-sized, else $R(m, n)$ is even-sized.

$R(m, n)$ is called a 1-rectangle if either $m = 1$ or $n = 1$, and a 2-rectangle if either $n = 2$ and $m > 1$, or $m = 2$ and $n > 1$. Figure 2 gives an example of a rectangular grid graph.

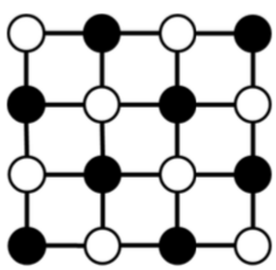

Figure 2: A rectangular grid graph $R(4, 4)$

Complexity

In graph theory, the Hamiltonian path problem and the Hamiltonian cycle problem are problems of determining whether a Hamiltonian path (path in a graph that visits each vertex exactly once) or a Hamiltonian cycle exists in a given graph [11].

Note that, the Hamiltonian path problem is known to be NP-complete on general graphs. In fact, a problem is said to be NP hard if it can be solved in Polynomial time using a Non-deterministic Turing machine.

NP is the set of all decision problems for which the "yes" answers can be verified in polynomial time ($O(n^k)$ where n is the problem size, and k is a constant), by a deterministic Turing machine.

P is the set of all decision problems which can be solved in polynomial time by a deterministic Turing machine. Since they can be solved in polynomial time, they can also be verified in polynomial time. Therefore P is a subset of NP.

A problem x that is in NP is also in NP-Complete if and only if every other problem in NP can be quickly transformed into x in polynomial time. Therefore, what makes NP-Complete so interesting is that if any one of the NP-Complete problems was to be solved quickly, then all NP problems can be solved quickly.

NP-Hard are problems that are at least as hard as the hardest problems in NP. Note that NP-Complete problems are also NP-hard. However not all NP-hard problems are NP (or even a decision problem), despite having NP as a prefix. That is the NP in NP-hard means non-deterministic polynomial time.

The longest path problem is the problem of finding a simple path of maximum length in a given graph. We can clearly see from its definition, that the longest path problem is a generalisation of the Hamiltonian path problem.

The end-to-end longest path problem is the problem of finding the longest path between a given pair of vertices [13].

Another well known problem in graph theory is the shortest path problem, but in this paper we only consider the end-to-end longest path problem.

The Hamiltonian path problem and the longest path problem are both classic problems in graph theory. They have been the subject of previous studies, and many algorithms have already been proposed in order to solve them.

In this paper, we will try to generalise the results of the previous studies, in order to solve the longest path problem in a rectangular grid graph with two missing vertices.

Previous Results

In this section, we will discuss some previously established results and algorithms on the Hamiltonian and the Longest path problems.

Let $(R(m,n),s,t)$ denote the rectangular grid graph $R(m,n)$ with two specified distinct vertices s and t. $L(R(m,n),s,t)$ designs the length of the longest path between s and t.

Longest path problem is NP-hard on every class of graphs on which the Hamiltonian path problem is NP-complete and remains NP-complete even when restricted to some small classes of graphs such as grid graphs.

Indeed, it has been proved that even if a graph has a Hamiltonian path, the problem of finding a path of length $n - n^{\epsilon}$ for any $\epsilon < 1$ is NP-hard, where n is the number of vertices of the graph. Moreover, there is no polynomial-time constant-factor approximation algorithm for the longest path problem unless $P = NP$.

Bulterman is the first one who proposed a linear time algorithm using the Dijkstra algorithm to find the longest path in a tree[14].

Uehara and Uno put forward a linear time and space algorithm by generalizing the algorithm in [14] to solve the longest path problem on weighted trees and block graphs.

An $O(n^2)$ time and space algorithm on cactus graphs which are tree-like graphs, where n is the number of vertices of the given graph, was also proposed by them in 2004 [15]. In 2007, they presented a linear time (respectively, $O(n^2)$) algorithms on block graphs and bipartite graphs (respectively, cactus graphs) [16].

The first step of their algorithm is to transform the block graphs into a weighted tree. And the second step is to do Breath-First Search starting from a cut vertex to find the farest leaf, and the last step is to do Breath-First Search starting from the leaf and find the path to the farest leaf to obtain the longest path.

An $O(n^3(m + nlogn))$ algorithm was also proposed for interval biconvex graphs [17].

Moreover, in 2007, Uehara and Valiente used the linear structure of bipartite permutation graphs to solve the longest path problem on bipartite permutation graphs in linear time and space [17]. And in 2008, Takahara solved this longest path problem on ptolemaic graphs, which takes $O(n^5)$ time and $O(n^2)$ space [18].

Recently, Ioannidou solved this problem on cocomparability graphs in polynomial time [19]. Next year, he presented an O(n4) algorithm on interval graphs [20].

In 2011, Mertzios proposed a polynomial time algorithm on circular-arc graphs [21].

In 2013, Dong [22] presented an algorithm for solving this problem on block graphs, cactus graphs, and probe block graphs.

In 1982, Itai et al. proved that Hamiltonian path problem on general grid graphs, with or without specified endpoints, is NP-complete [23]. But the problem on rectangular grid graphs is in P requiring only linear-time.

Twenty years later, Chen et al. improved their algorithm [23] and presented a parallel algorithm for Hamiltonian path problem in mesh architecture [24].

In recent years, Zhang and Liu proposed an approximation algorithm for the longest path problem in grid graphs and their algorithm runs in $O(n^2)$ time in 2011 [25].

In 2012, Keshavarz-Kohjerdi et al. presented a linear time algorithm for solving the longest path problem on rectangular grid graphs without holes [26].

Later, they improved it into a parallel algorithm in 2013 [27].

The table below shows some previous results for the longest path problem.

Graphs	Time Complexity
Block Graphs	$O(n)$ [Uehara 2004]
Bipartite Permutation Graphs	$O(n)$ [Uehara 2005]
Cactus Graphs	$O(n^2)$ [Uehara 2004]
Cocomparability Graphs	Polynomial Time [Ioannidou 2010]
Grid Graph	$O(n^2)$ [Zhang 2011]
Interval Biconvex Graphs	$O(n^3(m + nlogn))$ [Uehara 2005]
Interval Graphs	$O(n^4)$ [Ioannidou 2011]
Ptolemaic Graphs	$O(n^5)$ [Takahara 2008]
Rectangular Grid Graph	$O(mn)$ [Keshavarz-Kohjerdi 2013]
Trees	$O(n)$ [Bulterman 2002]

A cycle $C = (v_1, v_2, ..., v_c)$ in G is a cycle if $v_i \neq v_j$ $\forall$ $i \neq j$, such that $1 \leq i$, $j \leq c-1$ and $v_1 = v_c$. If $c = mn+1$, then the cycle is called Hamiltonian cycle.

By the following lemma, we can find a Hamiltonian cycle with the three sides of R being full of boundary edges as shown in Figure 3.

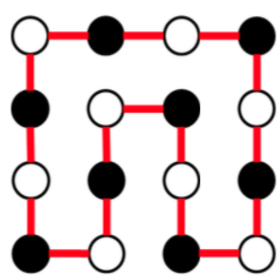

Figure 3: A Hamiltonian cycle for the rectangular grid graph $R(4, 4)$

Lemma 1 *[24] R(m,n) has a Hamiltonian cycle if and only if it is even-size and* $m, n > 1$.

Let $(R(m,n),s,t)$ denote the rectangular grid graph $R(m,n)$ with two given distinct vertices s and t. Without loss of generality, we assume $s_x \leq t_x$. And, $(R(m,n),s,t)$ is called Hamiltonian if there exists a Hamiltonian path between s and t in $R(m,n)$.

We use $P(R(m,n),s,t)$ to denote the problem of finding a longest path between two given distinct vertices s and t in a rectangular grid graph $R(m,n)$. Let $L(R(m,n),s,t)$ denote the length of longest path between s and t and $U(R(m,n),s,t)$ denote the upper bound of the length of the longest path between s and t.

Note that two different vertices v and $v\prime$ in $R(m,n)$ are called color-compatible if either both v and $v\prime$ are white and $R(m,n)$ is odd-sized, or v and $v\prime$ have different colors and $R(m,n)$ is even-sized.

The number of black and white vertices in an even-sized rectangular grid graph are the same. Therefore, any Hamiltonian path in the graph must have different colors for its two end-vertices.

On the other hand, an odd-sized rectangular grid graph has one more white vertex than black vertex. That is, the colors of the two end-vertices in any Hamiltonian path of an odd-sized rectangular grid graph must be white.

In addition, Itai et al. [23] showed that if one of the following conditions hold, then $(R(m,n),s,t)$ is not Hamiltonian:

(F1) $R(m,n)$ is a 1-rectangle and either s or t is not a corner vertex.
(F2) $R(m,n)$ is a 2-rectangle and (s,t) is a non-boundary edge, (i.e: (s,t) is an edge and it is not on the outer face).
(F3) $R(m,n)$ is isomorphic to a 3-rectangle grid graph $R\prime(m,n)$ such that s and t is mapped to $s\prime$ and $t\prime$ and all of the following three conditions hold:
1. m is even,
2. $s\prime$ is black, $t\prime$ is white,
3. $s\prime_y = 2$ and $s\prime_x < t\prime_x$ or $s\prime_y \neq 2$ and $s\prime_x < t\prime_x - 1$

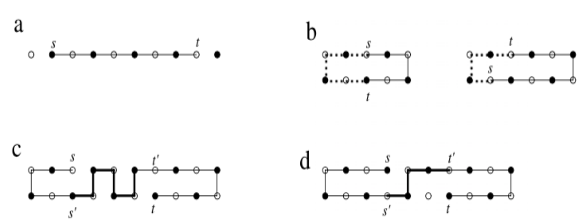

Figure 4: *(a) Longest path between s and t in a 1-rectangle, (b) Longest path between s and t in a 2-rectangle, ((c), (d)) Paths with length* $2m$ *and* $2m-1$ *for a 2-rectangle, respectively*

And also, $(R(m,n),s,t)$ is Hamiltonian if and only if s and t are color-compatible in $R(m,n)$, and s and t do not satisfy any of conditions (F1), (F2) and (F3). According to [26], $(R(m,n),s,t)$ must satisfies one of the following conditions:

(C0) s and t are color-compatible and none of (F1)-(F3) holds.

(C1) Neither (F1) nor (F2) holds and either:

- 1. $R(m,n)$ is even-sized and s and t are same-colored or
- 2. $R(m,n)$ is odd-sized and s and t are different-colored.

(C2)

- 1. $R(m,n)$ is odd-sized and s and t are black-colored and neither (F1) nor (F2*) holds, or
- 2. s and t are color-compatible and (F3) holds, where (F2*) is defined as follows:

(F2*) $R(m,n)$ is a 2-rectangle and $s_x = t_x$ or ($s_x = t_x - 1$ and $s_y \neq t_y$).

They also proved the following upper bounds for the length of longest paths:

$$U(R(m,n),s,t) = \begin{cases} t_x - s_x + 1 & \text{if } (F1)\ , \\ max(t_x + s_x, 2m - t_x - s_x + 2) & \text{if } (F2^*), \\ mn, & \text{if } (C0), \\ mn - 1, & \text{if } (C1), \\ mn - 2, & \text{if } (C2). \end{cases} \tag{1}$$

Theorem 1 *In a rectangular grid graph $R(m,n)$, a longest path between any two vertices s and t can be found in linear time and its length $L(R(m,n),s,t)$ is equal to $U(R(m,n),s,t)$.*

In [27], the authors presented a sequential longest path algorithm. For completeness, we showed it in Algorithm 1.

Algorithm 1: The sequential longest path algorithm

Input : A rectangular grid graph $R(m, n)$ with two disjoint vertices s and t

Output: A Longest path of G

1 **Step 1:** By a peeling operation, partitions $R(m, n)$ into five disjoints rectangular grid subgraphs $R1$ to $R5$, such that s, $t \in R5$

2 **Step 2:** Find a longest path between s and t in $R5$

3 **Step 3:** Construct Hamiltonian cycles in rectangular subgraphs $R1$ to $R4$

4 **Step 4:** Construct a longest path between s and t in R by combining the Hamiltonian cycles of $R1$ to $R4$ and the longest path of $R5$

For the case of the longest path problem in a rectangular grid graph with a hole, significant results have already been found, and the following relation gives the upper bound.

Let $M = (R(m, n), s, t, h)$. The Upper bound for the longest path length in M, denoted by $U(M)$, is defined by:

$$U(M) = \begin{cases} mn - 2 & \text{even-sized, } s, t \text{ diff. colors,} \\ mn - 3 & \text{even-sized, } s, t, h \text{ the same color,} \\ mn - 1 & \text{even-sized, } s, t \text{ the color, h not} \\ mn - 1, & \text{odd-sized, } s, t \text{ diff. colors, h white} \\ mn - 3, & \text{odd-sized, } s, t \text{ diff. colors, h black} \\ mn - 2, & \text{odd-sized, } s, t \text{ white} \\ mn - 2, & \text{odd-sized, } s, t \text{ black, h white} \\ mn - 4, & \text{odd-sized, } s, t, h \text{ black} \end{cases} \tag{2}$$

In this article, we will concentrate on the study of the longest path problem in a rectangular grid graph with two holes $(R(m, n), s, t, h_1, h_2)$.

The algorithms stated above can not directly solve this problem, therefore, our efforts will be directed on the elaboration of a reliable and fast algorithm that will solve this problem for any rectangular $(R(m, n), s, t, h_1, h_2)$ graph.

Figure 5 shows a rectangular grid graph with two holes.

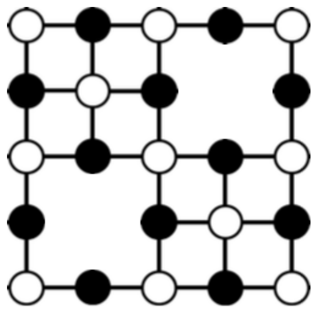

Figure 5: A rectangular grid graph with two holes

Statement of the Problem

Given a rectangular grid graph with two holes $(R(m,n), s, t, h_1, h_2)$, our problem will be to find the longest path between s and t.

In order to solve this problem, our method will be to first, use the divide-and-conquer approach to reduce the size of the problem.

Therefore, we will start by the peeling operation, in order to remove the extra paddings and reduce the size of the problem.

The figure below describes how the peeling operation is performed:

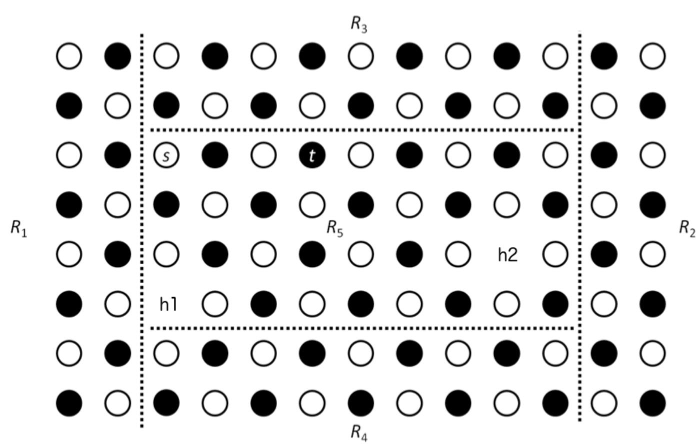

Figure 6: A peeling operation on $R(13,8)$ with two holes

Our approach will be to first peel 4 even-sized graphs on the sides, where Hamiltonian cycles can be created easily, and also reduce the size of R_5 as much as possible without compromising the length of the longest path.

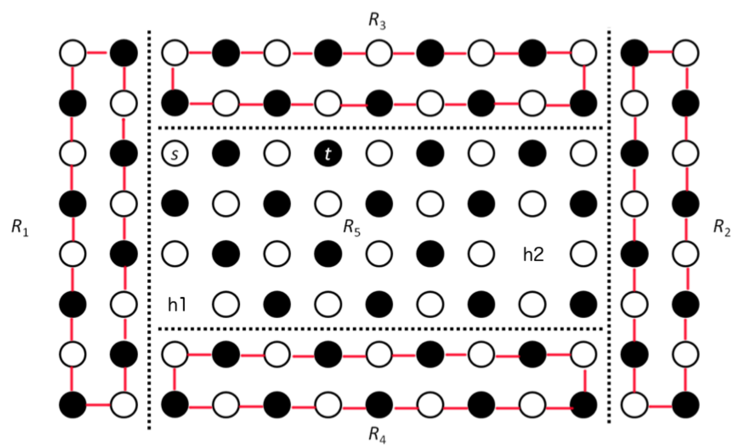

Figure 7: Hamiltonian cycles in R_1 - R_4

The second step after a peeling operation, is that we are going to construct a longest path $L(R(m,n), s, t, h_1, h_2)$ in R_5.

Recall that s, t, h_1 and h_2 must all remain in R_5 after the peeling operation.

In order to find the longest path, we will use converging to first get rid of a hole, and then we will solve for $L(R(m,n), s, t, h)$.

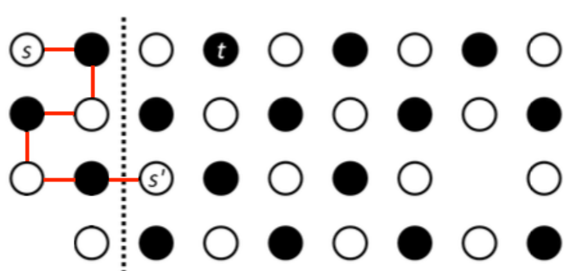

Figure 8: Convergence Process in R_5

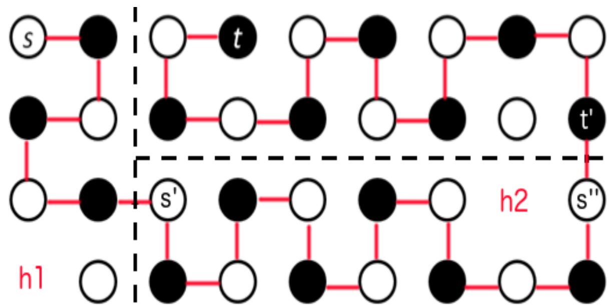

Figure 9: Convergence Process in R_5

In fact, the task of finding $L(R(m,n),s,t,h_1,h_2)$ in R_5 by converging, will lead us to one of the following cases:

1. There is only s or t

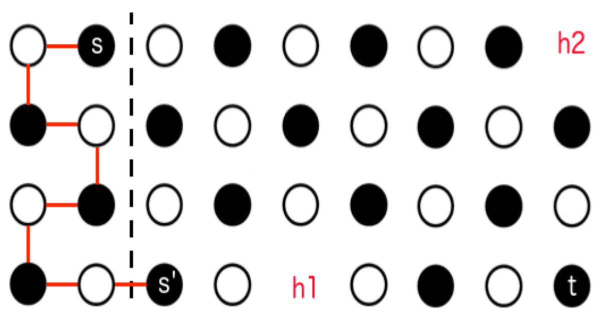

Figure 10: $R_5(9,4)$.

In this case, keep converging s or t until you get rid of a hole.
Then compute $L(R(m,n),s,t,h)$.

2. There is only one hole.

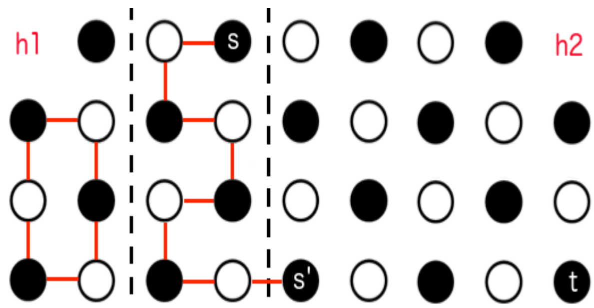

Figure 11: $R_5(9,4)$.

In this case, create a cycle to eliminate the hole. Note that we will loose at least one vertex. Then, compute $L(R(m,n),s,t,h)$.

3. There is only one hole with either s or t.

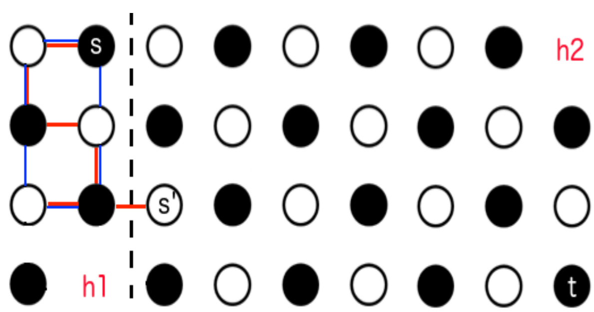

Figure 12: $R_5(9,4)$.

In this case, converge either s or t, and then create a cycle (it will have at least 3 missing vertices). Afterwards, compute $L(R(m,n),s,t,h)$ between the converged endpoint and the original one.

4. There is two holes.

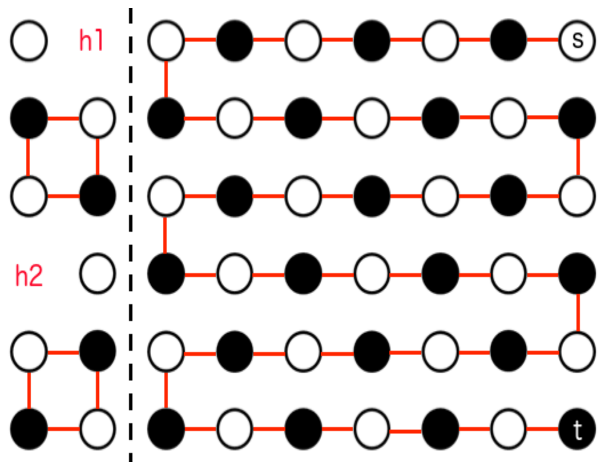

Figure 13: $R_5(9, 4)$.

In this case, there will be at least two cycles with at least three missing vertices. Then, find $L(R(m, n), s, t, h)$ between s and t that is, in this case, the Hamiltonian path between s and t.

For a given graph $R(m, n)$, if $n = 1$, $L(R(m, n), s, t, h_1, h_2)$ doesn't exist if $s_x < h_1x < t_x$ or $s_x < h_2x < t_x$ (ie: h_1 or h_2 are between s and t). Otherwise, the longest path exists and its length cannot exceed $|t_x - s_x| + 1$.

In a $(R(m, 2), s, t, h_1, h_2)$ graph, if h_1 and h_2 are under the same column and split s ant t in two parts (ie $h_1y = h_2y$ and $h_1x \neq h_2x$), then the longest path doesn't exist. Also, if $|h1_y - h2_y| = 1$ and $h1_x \neq h2_x$ (ie: they're at distance 1 and on different columns), while h_1 and h_2 are between s and t, then $L(R(m, n), s, t, h_1, h_2)$ doesn't exist.

Otherwise the longest path exists, and $U(R(m, n), s, t, h_1, h_2)$ can be found using the following relation:

In a rectangular grid graph with two holes $(R(m, n), s, t, h_1, h_2)$, the upper bound of the longest path between two vertices s and t can be defined as:

$$U(M) = \begin{cases} mn-4 & \text{if even-sized, } s,\ t \text{ diff colors, } h_1,\ h_2 \text{ same colors,} \\ mn-2 & \text{if even-sized, } s,\ t \text{ diff colors, } h_1,\ h_2 \text{ diff colors,} \\ mn-3 & \text{if even-sized, } s,\ t \text{ same colors, } h_1,\ h_2 \text{ same colors but diff than } (s,t), \\ mn-4 & \text{if even-sized, } s,\ t \text{ same colors, } h_1,\ h_2 \text{ same colors and same as } (s,t), \\ mn-3 & \text{if even-sized, } s,\ t \text{ same colors, } h_1,\ h_2 \text{ diff colors,} \\ mn-3 & \text{if odd-sized, } s,\ t \text{ diff colors, } \forall\ h_1,\ h_2, \\ mn-4 & \text{if odd-sized, } s,\ t \text{ same colors, } \forall\ h_1,\ h_2 \end{cases} \tag{3}$$

Our Algorithm

In this section, we will develop an algorithm to solve the longest path problem for any $(R(m,n), s, t, h_1, h_2)$, and we will use the upper bound found in the previous section to evaluate the correctness of our solution.

Algorithm 1: The longest path in a rectangular grid graph $(R(m,n), s, t, h_1, h_2)$
Data: A rectangular grid graph $(R(m,n), s, t, h_1, h_2)$
Result: The longest path $L(R(m,n), s, t, h_1, h_2)$

1. Use the peeling operation described in the last section, to partition $R(m,n)$ into 5 subgraphs R_1, R_2, R_3, R_4 and R_5.
2. While($R_5(m_5, n_5) \geq 4$)
   - Converge
3. Find Hamiltonian cycles in R_1 R_2 R_3 and R_4.
4. Find the longest path $L(R(m,n), s, t, h)$ in R_5 based on the smaller cases studied previously.
5. Connect the longest path in R_5 with the Hamiltonian cycles.

Algorithm 2: The Converging process in R_5

- Case 1: Keep converging s or t until you get rid of a hole.Then compute $L(R(m,n),s,t,h)$.

- Case 2: Create a cycle to eliminate the hole. Note that we will loose at least one vertex. Then, compute $L(R(m,n),s,t,h)$.

- Case 3: Converge either s or t, and then create a cycle (it will have at least 3 missing vertices). Afterwards, compute $L(R(m,n),s,t,h)$ between the converged endpoint and the original one.

- Case 4: In this case, there will be at least two cycles with at least three missing vertices. Then, find $L(R(m,n),s,t,h)$ between s and t that is, in this case, the Hamiltonian path between s and t.

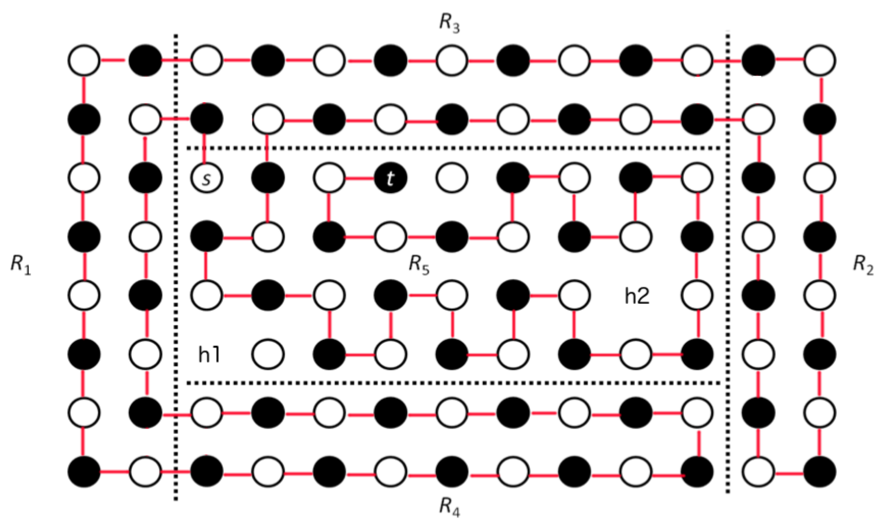

Figure 14: *Longest path $L(R(m,n),s,t,h_1,h_2)$ in R(13,8)*

Conclusion

In this paper, we proposed an algorithm to solve the longest path problem on rectangular grid graphs with two holes.

Future studies, may try to carry-on with our results, and extend them for more than two missing vertices. However for that case, we can already see that the upper bounds will probably get worst, meaning that the length of the longest path will decrease.

Also, more holes might lead to more cases to handle, and the task of solving the longest path problem may get more difficult.

References

[1] $proofwiki.org/wiki/Definition : Graph(GraphTheory)$.

[2] Foundations of Algorithms 5th Edition, R.E. Neapolitan, Section 3.2 (p101)..

[3] Stanford University, CME 305: Discrete Mathematics and Algorithms, Lecture 2.

[4] Stanford University, CME 305: Discrete Mathematics and Algorithms, Lecture 2.

[5] Cormen, Thomas H. Leiserson, Charles E. Rivest, Ronald L. Stein, Clifford (2001), "B.4 Graphs", Introduction to Algorithms (2 ed.), MIT Press and McGraw-Hill, pp. 1080-1084..

[6] Graph Theory With Applications, J.A. Bondy and U.S.R. Murty, Department of Combinatorics and Optimisation, University of Waterloo, Ontario, Canada.

[7] Robertson, N. Seymour, P. Thomas, R. (November 1994). "Quickly Excluding a Planar Graph". Journal of Combinatorial Theory, Series B. 62 (2): 323?348.

[8] Brouwer, A. E. Cohen, A. M. and Neumaier, A. Distance-Regular Graphs. New York: Springer-Verlag, 1989.

[9] Steve Borgatti, MB 022: Introduction to Organisational Behaviour, 2001.

[10] McGill University, MATH 340: Discrete Structures II, Professor O.Kharlampovich.

[11] Michael R. Garey and David S. Johnson (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, ISBN 0-7167-1045-5 A1.3: GT37-39, pp. 199-200.

[12] $wikipedia.org/wiki/Longest - path - problem$.

[13] Intelligent Systems and Applications: Proceedings of the International Computer Symposium (ICS) (p59).

[14] R. Bulterman, F. van der Sommen, G. Zwaan, T. Verhoeff, A. van Gasteren, and W. Feijen, On Computing a Longest Path in a Tree, Inf. Process. Lett, Vol. 81, No. 2, pp. 93-96, 2002..

[15] R. Uehara, and Y. Uno, Efficient Algorithms for the Longest Path Problem, ISAAC, pp. 871-883, 2004.

[16] R. Uehara, and Y. Uno, On Computing Longest Paths in Small Graph Classes, Int. J. Found. Comput. Sci., Vol. 18, No. 5, pp. 911-930, 2007.

[17] R. Uehara, and G. Valiente, Linear Structure of Bipartite Permutation Graphs and the Longest Path Problem, Inf. Process. Lett., Vol. 103, No. 2, pp. 71-77, 2007..

[18] Y. Takahara, S. Teramoto, and R. Uehara, Longest Path Problems on Ptolemaic Graphs, IEICE Transactions on Information and Systems, Vol. 91-D, No. 2, pp. 170?177, 2008.

[19] K. Ioannidou, and S.D. Nikolopoulos, The Longest Path Problem is Polynomial on Cocomparability Graphs, WG, pp. 27-38, 2010.

[20] K. Ioannidou, S.D. Nikolopoulos, and G.B. Mertzios, The Longest Path Problem Has a Polynomial Solution on Interval Graphs, Algorithmica, Vol. 61, No. 2, pp. 320-341, 2011.

[21] G.B. Mertzios, and I. Bezakova, Computing and Counting Longest Paths on Circular-Arc Graphs in Polynomial Time, Electronic Notes in Discrete Mathe- matics, Vol. 37, pp. 219-224, 2011.

[22] Ang-Lin Dong, A Study of the Longest Path Problem on Some Tree-like Graphs, Master thesis, National Dong Hwa University, 2013.

[23] A. Itai, C. Papadimitriou and J. Szwarcfiter, Hamiltonian Paths in Grid Graphs, SIAM J. Comput., 11(4):676-686, 1982.

[24] S. D. Chen, H. Shen and R. Topor, An Efficient Algorithm for Constructing Hamiltonian Paths in Meshes, J. Parallel Computing, 28(9):1293-1305, 2002.

[25] W. Q. Zhang and Y. J. Liu, Approximating the Longest Paths in Grid Graphs, Theoretical Computer Science, 412(39): 5340-5350, 2011.

[26] F. Keshavarz-Kohjerdi, A. Bagheri and A. Asgharian-Sardroud, A Linear-time Algorithm for the Longest Path Problem in Rectangular Grid Graphs, Discrete Applied Math., 160(3): 210-217, 2012.

[27] F. Keshavarz-Kohjerdi and A. Bagheri, An Efficient Parallel Algorithm for the Longest Path Problem in Meshes, The Journal of Supercomputing, Volume 65, Issue 2, pp 723-741, 2013.