Discrete Applied Mathematics 160 (2012) 210–217

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

A linear-time algorithm for the longest path problem in rectangular grid graphs

Fatemeh Keshavarz-Kohjerdi [a,*], Alireza Bagheri [b], Asghar Asgharian-Sardroud [b]

[a] *Department of Computer Engineering, Islamic Azad University, North Tehran Branch, Tehran, Iran*

[b] *Department of Computer Engineering & IT, Amirkabir University of Technology, Tehran, Iran*

ARTICLE INFO

Article history:
Received 13 December 2010
Received in revised form 11 August 2011
Accepted 14 August 2011
Available online 29 September 2011

Keywords:
Hamiltonian path
Hamiltonian cycle
Grid graph
Longest path
Rectangular grid graph

ABSTRACT

The longest path problem is a well-known NP-hard problem and so far it has been solved polynomially only for a few classes of graphs. In this paper, we give a linear-time algorithm for finding a longest path between any two given vertices in a rectangular grid graph.

1. Introduction

The longest path problem, i.e. the problem of finding a simple path with the maximum number of vertices, is one of the most important problems in graph theory. The well-known NP-complete Hamiltonian path problem [4,8], i.e. deciding whether there is a simple path that visits each vertex of the graph exactly once, is a special case of the longest path problem. Only a few polynomial-time algorithms are known for the longest path problem for special classes of graphs.

Trees are the first class of graphs that a polynomial-time algorithm for the longest path problem has been found for (i.e. finding the diameter of an unweighted tree). Originally, this algorithm was proposed by Dijkstra around 1960 but Bulterman et al. [2] provided a proof for it, and later it was improved by Uehara and Uno [16] for the case of weighted trees. They also solved the problem for block graphs in linear time and for cacti in quadratic time. Furthermore, they introduced interval biconvex graphs as a subclass of interval graphs and solved the problem for them in $O(n^3(m + n\log n))$ time, where n denotes the number of vertices and m denotes the number of edges of the given graph. Recently, Ioannidou et al. [13] showed that the problem is polynomial for general interval graphs. Their algorithm is based on dynamic programming and runs in $O(n^4)$ time. More recently, Mertzios and Corneil [15] solved the problem in polynomial time for the larger class of graphs i.e. cocomparability graphs. Also, there is an $O(n^3)$-time algorithm for the problem for complete m-partite digraphs proposed by Gutin [9].

In the area of approximation algorithms it has been shown that the problem is not in APX, i.e. there is no polynomial-time approximation algorithm with constant factor for the problem unless P $=$ NP [9]. Also, it has been shown that finding a path of length $n - n^\epsilon$ is not possible in polynomial time unless P $=$ NP [11]. To our knowledge, the best-known approximation

* Corresponding author. Fax: +98 02166495521.
E-mail addresses: fatemeh.keshavarz.2003@gmail.com (F. Keshavarz-Kohjerdi), ar_bagheri@aut.ac.ir (A. Bagheri), asgharian@aut.ac.ir (A. Asgharian-Sardroud).

0166-218X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2011.08.010

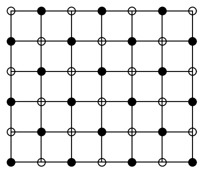

Fig. 1. The rectangular grid graph $R(7, 6)$.

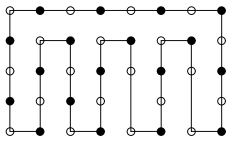

Fig. 2. A Hamiltonian cycle for the rectangular grid graph $R(8, 5)$.

algorithm for the problem has the ratio of $O(n(\log\log n/\log n)^2)$ [1]. For more related results on approximation algorithms on general graphs see [5–7,17].

Rectangular grid graphs first appeared in [14], in which Luccio and Mugnia tried to solve the Hamiltonian path problem. The rectangular grid graph $R(n, m)$ is the subgraph of G^∞ (the infinite grid graph) induced by $V(m, n) = \{\upsilon | 1 \leq \upsilon_x \leq m, 1 \leq \upsilon_y \leq n\}$, where υ_x and υ_y are respectively x and y coordinates of υ (see Fig. 1). A solid grid graph is a grid graph without holes. The Hamiltonian path problem has been studied for grid graphs in [10], where the authors gave necessary and sufficient conditions for the existence of a Hamiltonian path in rectangular grid graphs and proved that the problem for general grid graphs is NP-complete. Also, Chen et al. [3] presented a parallel algorithm for finding a Hamiltonian path between two given vertices of a rectangular grid graphs in constant time when there is a processor for each vertex. There is a polynomial-time algorithm for finding the Hamiltonian cycle in solid grid graphs [12]. In this paper, we consider the longest path between two given vertices in rectangular grid graphs.

The paper is organized as follows. In Section 2, preliminaries and background are presented. Some upper bounds on the lengths of longest paths are given in Section 3. In Section 4, the algorithm for finding the longest path problem is introduced and the conclusion is given in Section 5.

2. Preliminaries and background

In this section, we present definitions and some previously established results on the Hamiltonian and the longest path problems in grid graphs which have been presented in [3,10].

The *two-dimensional integer grid* G^∞ is an infinite graph with vertex set of all the points of the Euclidean plane with integer coordinates. In this graph, there is an edge between any two vertices at unit distance. For a vertex υ of this graph, let υ_x and υ_y denote the x and y coordinates of its corresponding point (sometimes we use (υ_x, υ_y) instead of υ). We color the vertices of the two-dimensional integer grid as black and white. A vertex υ is colored *white* if $\upsilon_x + \upsilon_y$ is even, and it is colored *black* otherwise. A *grid graph* G_g is a finite vertex-induced subgraph of the two-dimensional integer grid. In a grid graph G_g, each vertex has degree at most four. Clearly, there is no edge between any two vertices of the same color. Therefore, G_g is a bipartite graph. Note that any cycle or path in a bipartite graph alternates between black and white vertices. This property helps us to bound the length of maximum paths or cycles in G_g. A *rectangular grid graph* $R(m, n)$ (or R for short) is a grid graph whose vertex set is $V(R) = \{\upsilon | 1 \leq \upsilon_x \leq m, 1 \leq \upsilon_y \leq n\}$. In the figures we assume that $(1, 1)$ are the coordinates of the vertex in the upper left corner. The size of $R(m, n)$ is defined to be mn. $R(m, n)$ is called *odd-sized* if mn is odd, and it is called *even-sized* otherwise. In this paper, without loss of generality we assume that $m \geq n$. $R(m, n)$ is called an *n-rectangle*.

The following lemma states a result concerning the Hamiltonicity of even-sized rectangular graphs.

Lemma 2.1 (*[3]*). *$R(m, n)$ has a Hamiltonian cycle if and only if it is even-sized and $m, n > 1$.*

Fig. 2 shows a Hamiltonian cycle for an even-sized rectangular grid graph, found by Lemma 2.1. Each Hamiltonian cycle found by this lemma contains all the boundary edges on the three sides of the rectangular graph. This shows that for an even-sized rectangular graph R, we can always find a Hamiltonian cycle such that it contains all the boundary edges except exactly one side of R which contains an even number of vertices.

Two different vertices υ and υ' in $R(m, n)$ are called *color-compatible* if either both υ and υ' are white and $R(m, n)$ is odd-sized, or υ and υ' have different colors and $R(m, n)$ is even-sized. Let $(R(m, n), s, t)$ denote the rectangular grid graph $R(m, n)$ with two specified distinct vertices s and t. Without loss of generality, we assume that $s_x \leq t_x$. $(R(m, n), s, t)$ is called *Hamiltonian* if there exists a Hamiltonian path between s and t in $R(m, n)$. An even-sized rectangular grid graph contains the same number of black and white vertices. Hence, the two end vertices of any Hamiltonian path in the graph must have different colors. Similarly, in an odd-sized rectangular grid graph the number of white vertices is greater by 1 than the number of black vertices. Therefore, the two end vertices of any Hamiltonian path in such a graph must be white. Hence,

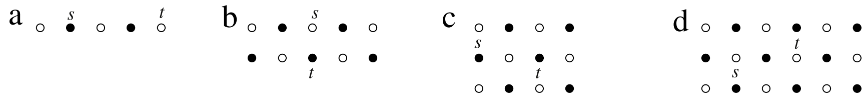

Fig. 3. Rectangular grid graph in which there is no Hamiltonian path between s and t.

the color-compatibility of s and t is a necessary condition for $(R(m,n),s,t)$ to be Hamiltonian. Furthermore, Itai et al. [10] showed that if one of the following conditions holds, 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 (Fig. 3(a)).
(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 (Fig. 3(b)).
(F3) $R(m,n)$ is isomorphic to a 3-rectangle grid graph $R'(m,n)$ such that s and t is mapped to s' and t' and:
1. m is even,
2. s' is black, t' is white,
3. $s'_y = 2$ and $s'_x < t'_x$ (Fig. 3(c)) or $s'_y \neq 2$ and $s'_x < t'_x - 1$ (Fig. 3(d)).

They also presented a linear-time algorithm for finding the Hamiltonian path when the necessary conditions are satisfied. Later, Chen et al. improved this algorithm and adjusted it for parallel execution [3].

Theorem 2.1 (*[10]*). *Let $R(m,n)$ be a rectangular graph and s and t be two distinct vertices of it. $(R(m,n),s,t)$ is Hamiltonian if and only if s and t are color-compatible, and $R(m,n)$, s and t do not satisfy any of conditions* (F1)–(F3).

In the following we use $P(R(m,n),s,t)$ to indicate the problem of finding a longest path between vertices s and t in a rectangular grid graph $R(m,n)$ and $L(R(m,n),s,t)$ to show the lengths of longest paths between s and t. If S is a subgraph of $R(m,n)$ then by $R-S$ we refer to the graph obtained by removing all vertices of S from $R(m,n)$.

3. An upper bound on the length of longest paths

In this section, for $(R(m,n),s,t)$ we give an upper bound on the length of longest paths between s and t. By the length of a path we mean the number of vertices of the path. The following lemmas give the bound.

Lemma 3.1. *In an even-sized rectangular grid graph $R(m,n)$, the length of any path between two same-colored vertices cannot exceed $mn-1$.*

Proof. Since $R(m,n)$ is bipartite, colors of vertices of any path must alternate between black and white. Therefore, any path between two same-colored vertices must have odd length and because mn is even, its length cannot exceed $mn-1$. □

Lemma 3.2. *In an odd-sized rectangular grid graph $R(m,n)$, the length of any path between two different-colored vertices cannot exceed $mn-1$ and the length of any path between two black-colored vertices cannot exceed $mn-2$.*

Proof. Like in the proof of Lemma 3.1, colors of vertices of any path in $R(m,n)$ must alternate between black and white. Any path between two different-colored vertices must have even length while mn is odd. Therefore, its length cannot exceed $mn-1$. Moreover, if two end vertices of a path are black-colored, then its length is odd and has one more black vertex than white vertex. But $R(m,n)$ is odd-sized and has one more white vertex than black vertex. Hence, the length of the path cannot exceed $mn-2$. □

Note that these lemmas do not put any bound on the length of paths between color-compatible vertices.

Lemma 3.3. *Let $R(m,n)$ be a rectangular graph and s and t be two vertices of it. If $n=1$, then the length of any path between s and t cannot exceed $|t_x - s_x| + 1$. If $n=2$ and $s_x = t_x$ or ($s_x = t_x - 1$ and $s_y \neq t_y$), then the length of any path between s and t cannot exceed $\max(t_x + s_x, 2m - t_x - s_x + 2)$. If $n=3$ and $R(m,n)$, s and t satisfy* (F3)*, then the length of any path between s and t cannot exceed $mn-2$.*

Proof. For $n=1$, there is only one single path between s and t that has the specified length. If $n=2$ and $s_x = t_x$ or ($s_x = t_x - 1$ and $s_y \neq t_y$), then removing s and t clearly disconnects the graph into two components and a simple path between s and t can only go through one of these components. Therefore, its length cannot exceed the size of the largest component, i.e. $\max(t_x + s_x, 2m - t_x - s_x + 2)$. In the case where $n=3$, when $R(m,n)$, s and t satisfy (F3), mn must be even and the colors of s and t must be different. Therefore, any path between s and t must have even length and because of Theorem 2.1 this length cannot exceed $mn-2$. □

On the basis of Theorem 2.1 and Lemmas 3.1–3.3, we can classify instances of the longest path problem $P(R(m,n),s,t)$ by means of the following conditions:

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

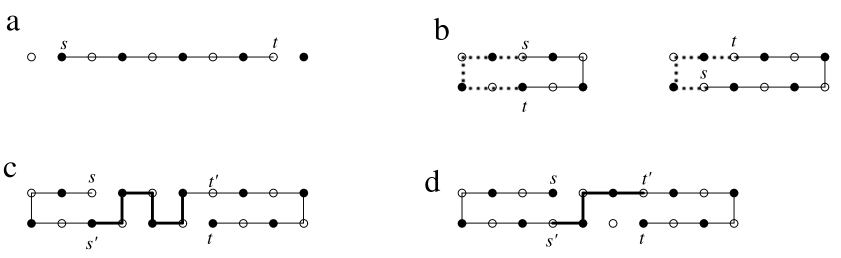

Fig. 4. (a) Longest path between s and t in a 1-rectangle. (b) Longest path between s and t in a 2-rectangle. ((c), (d)) Paths with length $2m$ and $2m-1$ for a 2-rectangle, respectively.

(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.

Here (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 $s_y \neq t_y$).

It is easy to show that any $(R(m, n), s, t)$ must satisfy one of conditions (C0), (C1), (C2), (F1) and (F2*). We define $U(R(m, n), s, t)$ to be the upper bound on the length of a longest path between s and t in $R(m, n)$. According to the given lemmas, if $(R(m, n), s, t)$ satisfies (C0), (C1) or (C2), then $U(R(m, n), s, t)$ is mn, $mn - 1$ or $mn - 2$, respectively. Otherwise $U(R(m, n), s, t)$ can be computed using Lemma 3.3. So, we have

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

4. The longest path algorithm

In this section, we present an algorithm for finding a longest path between two vertices of a rectangular grid graph. First, we solve the problem for 1-rectangles and 2-rectangles.

Lemma 4.1. *Let $P(R(m, n), s, t)$ be a longest path problem with $n = 1$ or $n = 2$. Then $L(R(m, n), s, t) = U(R(m, n), s, t)$.*

Proof. For a 1-rectangle obviously the lemma holds for the single possible path between s and t (see Fig. 4(a)). For a 2-rectangle, if removing s and t splits the graph into two components, then the path going through all vertices of the larger component has length equal to $U(R(m, n), s, t)$ (see Fig. 4(b)). Otherwise, let s' be the vertex adjacent to s and t' be the vertex adjacent to t such that $s'_y \neq s_y$ and $t'_y \neq t_y$. Then we make a path from s to s' and a path from t to t' as shown in Fig. 4(c), (d), and connect s' to t' by a path such that at most one vertex remains out of the path as depicted in this figure. □

From now on, we assume that $m \geq n > 2$, so one of conditions (C0), (C1) and (C2) should hold. Following the technique used in [10] we develop an algorithm for finding longest paths. The algorithm recursively divides the given rectangular graph. This recursive procedure ends when the graph becomes small enough for solving the problem by case analysis.

A *separation* of a rectangular graph R is a partition of R into two vertex disjoint rectangular grid graphs R_1 and R_2, i.e. $V(R) = V(R_1) \cup V(R_2)$, and $V(R_1) \cap V(R_2) = \emptyset$ [10]. A rectangular grid subgraph S of R strips a longest path problem $P(R, s, t)$ and is called a *strip* if:

1. S is even-sized and is not a 1-rectangle.
2. S and $R - S$ is a separation of R.
3. $s, t \in R - S$.
4. $U(R, s, t) = U(R - S, s, t) + |S|$, i.e. the stripping does not reduce the upper bound, where $|S|$ denotes the number of vertices of S.

See Fig. 5 as an example of stripping. Note that any even-sized subgraph of R contains the same number of black and white vertices and, on the basis of the definition of U when $m, n \geq 3$ and conditions 1–3 for stripping hold, condition 4 may fail only if (C0) holds and $R - S$ is a k-rectangle ($k \leq 3$) or one of conditions (C1) and (C2) holds and $R - S$ is a k-rectangle ($k \leq 2$). The following lemma shows that we can divide the problem by stripping. In this lemma, two non-incident edges e_1 and e_2 are parallel if each end vertex of e_1 is adjacent to some end vertex of e_2.

Lemma 4.2. *Let $P(R(m, n), s, t)$ be a longest path problem, and S be a strip of it and $m, n > 2$. If there is a path of length $U(R - S, s, t)$ between s and t in $R - S$, then there is a path of length $U(R, s, t)$ between s and t in R.*

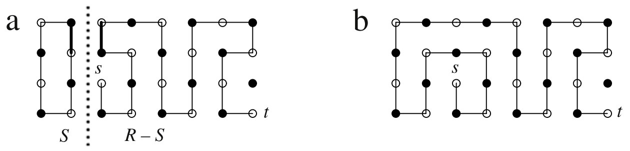

Fig. 5. A strip of $P(R(8,4),s,t)$.

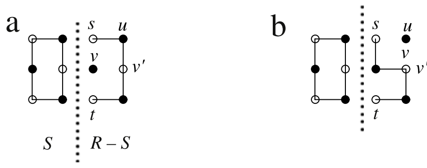

Fig. 6. A strip of $(R(4,3),s,t)$ when R is a 3-rectangle and s,t are the corners of $R-S$.

Proof. Since S is even-sized, by Lemma 2.1 we can find a Hamiltonian cycle in S containing all the boundary edges of S facing $R-S$, (i.e. we can find a Hamiltonian cycle of S such that it contains all the edges of S that are parallel to some edge of $R-S$). Furthermore, any path P of length $U(R-S,s,t)$ in $R-S$ contains all the vertices of $R-S$ except one or two vertices. Therefore, P should contain a boundary edge of $R-S$ that is adjacent to S, except when R is a 3-rectangle, in which case $R-S$ may have no boundary edge adjacent to S (see Fig. 6(a)). But in this case, s should be adjacent to S (otherwise three vertices remain outside of P). Let an edge (s,v) of R be adjacent to S. Then P must contain an edge (s,u) such that u is not adjacent to S and an edge (u,v') such that v' is adjacent to v as depicted in Fig. 6(a) (we may need to swap the roles of s and t to find such edges). Note that if (u,v') is not in P, then v' is not in P and we can increase the size of P by replacing (s,u) with edges (s,v), (v,v') and (v',u). Hence, replacing the edges (s,u) and (u,v') with edges (s,v) and (v,v') we can modify P such that it contains a boundary edge adjacent to S (see Fig. 6(b)). Using two parallel edges of P and the Hamiltonian cycle of S such as the two darkened edges of Fig. 5(a) we can combine them as illustrated in Fig. 5(b) and obtain a path of length $U(R,s,t)$ for R. □

Let v and v' be two distinct vertices in R. If $v_x \le 2$ and $v'_x \ge m-1$, then v and v' are called *antipodes*. We have the following lemma:

Lemma 4.3. *Let $P(R(m,n),s,t)$ be a longest path problem which cannot be stripped (by any strip S) and $m,n>2$. Then s and t are antipodes, or $(m,n)\in\{(5,4),(4,4)\}$ and (C0) holds, or $(m,n)=(4,4)$ and (C1) holds.*

Proof. Let S be a subgraph of R such that $V(S)=\{v\in V(R)|v_x\le 2\}$. Except for the exceptional cases (i.e. $(m,n)\in\{(5,4),(4,4)\}$ and (C0) holds, or $(m,n)=(4,4)$ and (C1) holds), we show that if $s\notin S$, then S is a strip for R. This contradicts the fact that R has no strip. Therefore, s and t must be antipodes in general. Itai et al. [10] have proved the lemma for the case where $(R(m,n),s,t)$ satisfies (C0). Note that if $s\notin S$, then S satisfies conditions 1–3 of stripping and when (C1) or (C2) holds it may not satisfy condition 4 only if $R-S$ is k-rectangle ($k\le 2$). When (C1) holds we have the following cases:

Case 1. $m>4$, $n>2$. Hence $R-S$ is neither a 1-rectangle nor a 2-rectangle; then S is a strip.
Case 2. $m=3$, $n=3$. Hence $R-S$ is a 1×3 rectangular grid graph and s,t are different-colored and there is a path of length 2 between s and t, so S is a strip (Fig. 7(a)).
Case 3. $m=4$, $n=3$. s and t must be same-colored because (C1) holds. It is easy to find a path of length 5 between s and t in $R-S$ and merge it with a Hamiltonian cycle of S. So S is a strip (Fig. 7(b), (c)).

And if (C2) holds then S is a strip because:

Case 1. $m>4$, $n>2$. $R-S$ is neither a 1-rectangle nor a 2-rectangle.
Case 2. $m=3$, $n=3$. This cannot occur because $R-S$ has just one black vertex while both s and t should be black-colored when (C2) holds.
Case 3. $m=4$, $n=4$. This cannot happen because when (C2) holds either R is odd-sized or (F3) holds, but R is even-sized and (F3) holds just for 3-rectangles.
Case 4. $m=4$, $n=3$. (F3) must hold. It is easy to find a path of length 4 between s and t in $R-S$, so S is a strip (Fig. 8). □

Let $P(R(m,n),s,t)$ be a longest path problem and (p,q) be an edge of R. Then we say that (p,q) *splits* $(R(m,n),s,t)$ if there exists a separation of R into R_p and R_q such that (Fig. 9):

1. $s,p\in R_p$ and $q,t\in R_q$.
2. $U(R(m,n),s,t)=U(R_p,s,p)+U(R_q,q,t)$.

A longest path problem $P(R(m,n),s,t)$ is called *prime* if it cannot be stripped or split.

Lemma 4.4. *Let $P(R(m,n),s,t)$ be a prime longest path problem; then $m,n\le 3$ or $(m,n)=(5,4)$ and (C0) holds.*

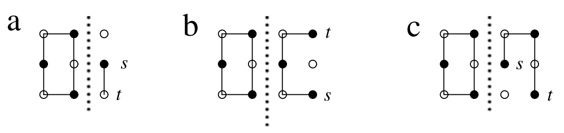

Fig. 7. (a) A strip of $(R(3,3), s, t)$. ((b), (c)) A strip on $(R(4,3), s, t)$, when $U(R, s, t) = mn - 1$.

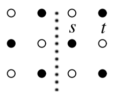

Fig. 8. A strip of $(R(4,3), s, t)$, when (F3) holds.

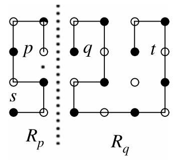

Fig. 9. A split of $R(6,4)$.

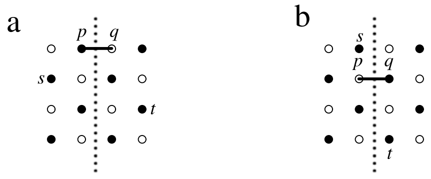

Fig. 10. A split on $R(4,4)$.

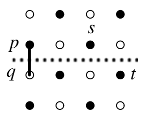

Fig. 11. A split of $(R(4,4), s, t)$, when s and t are not antipodes.

Proof. The lemma has been proved for the case where (C0) holds (see [10]). For the other cases (i.e. where (C1) or (C2) holds), first assume that s and t are antipodes. Let S be a subgraph of R such that $V(S) = \{v \in V(R) | v_x \leq 2\}$ and $s \in S$. We show that there exists a split edge (p, q) such that $R_p = S$:

Case 1. $m > 4, n \geq 4$. There is at least one vertex v such that $v_x = 2$, and v is colored differently from s and is not connected to s by a non-boundary edge of S. Let q be adjacent vertex of v in $R - S$. Clearly $q \neq t$ since $t_x \geq m - 1 > 3 = q_x$. Because p and s are color-compatible, $U(R_p, s, p) = 2 \times n$. $U(R_q, q, t) = U(R, s, t) - 2 \times n$, because q and s have the same colors, R_q is $(m-2) \times n$ and $n > 3$. So the edge (p, q) splits the problem.

Case 2. $m \geq 4, n = 3$. Always $((2,1),(3,1))$ or $((2,2),(3,2))$ or $((2,3),(3,3))$ splits (R, s, t).

Case 3. $m = 4, n = 4$. In this case, we assume s and t are same-colored; otherwise (C0) holds. So (C1) holds and one of $((2,1),(3,1))$ or $((2,4),(3,4))$ splits (R, s, t) (Fig. 10(a)), except when $s = (2,1)$ and $t = (3,4)$, or $s = (2,4)$ and $t = (3,1)$. In these cases, $((2,2),(3,2))$ splits (R, s, t) (Fig. 10(b)).

The only case where s and t are not antipodes (based on Lemma 4.3) should be isomorphic to the graph depicted in Fig. 11, which can be split by $((1,2),(1,3))$. □

Lemma 4.5 (*[10]*). *If* $(R(5,4), s, t)$ *satisfies* (C0), *then there is a Hamiltonian path between s and t in R.*

The following lemma completes our divide and conquer algorithm for the longest path problem in rectangular grid graphs.

Lemma 4.6. *In any prime longest path problem $P(R, s, t)$, the lengths of longest paths are equal to $U(R, s, t)$.*

Proof. Considering Lemmas 4.4 and 4.5, we just need to check the cases of $m, n \leq 3$. We showed that when $n = 1, 2$, the problem can be solved easily. The only remaining case is that where $m, n = 3$; the longest paths of all the possible problems in this case are depicted in Fig. 12 (the isomorphic cases are omitted). □

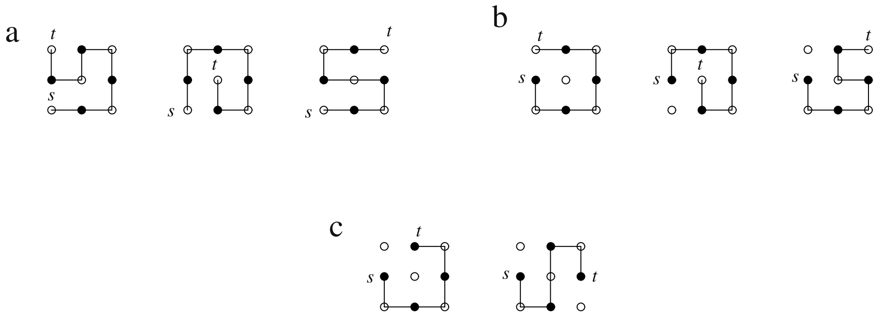

Fig. 12. For $n = m = 3$, (a) s and t are white, and then there is Hamiltonian path, (b) s and t have different colors, and then there is a path with $U(R, s, t) = mn - 1$ and (c) s and t are black, and then there is a path with $U(R, s, t) = mn - 2$.

Since all the proofs presented in this section were constructive, they give us an algorithm for finding a longest path in rectangular grid graphs. The pseudo-code of the algorithm is given in Algorithm 4.1. In this algorithm, *HamiltonianCycle* is the procedure that finds the Hamiltonian cycle of a strip based on Lemma 2.1, *MergeStrip* is a procedure that merges a path and a cycle based on Lemma 4.2 and *MergeSplit* is a procedure that connects two paths by simply adding an edge between their end vertices p and q. The algorithm first checks whether the input graph R is a 1-rectangle or 2-rectangle, then finds the longest path according to Lemma 4.1. Otherwise, if R can be stripped, then it strips R, by S, and recursively finds the longest path of $R - S$. Then the longest path of $R - S$ and the Hamiltonian cycle of S are merged into a single path. Otherwise, if R can be split, then it splits R into R_p and R_q, and recursively finds the longest paths of R_p and R_q. Then these two longest paths are merged into a single path. Otherwise, (R, s, t) is a prime problem, and the algorithm finds the longest path of R according to Lemma 4.6. Theorem 4.1 summarizes our results.

Theorem 4.1. *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 (i.e., $L(R, s, t)$) is equal to $U(R, s, t)$.*

Proof. When $n = 1$ or $n = 2$ the theorem is true due to Lemma 4.1. When $n > 2$ we can strip and split the problem without changing the upper bound of the length of its longest paths until it becomes prime. In this case, we can find a path of the length equal to the $U(R, s, t)$ according to Lemmas 4.5 and 4.6. The algorithm divides the problem in $O(1)$ time and solves the subproblems recursively and then merges the results in $O(1)$ time. Because at most $m + n$ strips and splits are possible on R, in time $O(m+n)$ the algorithm can find the structure of the longest path. But, if it needs to report the path (by reporting the sequence of its vertices), then it needs $O(mn)$ time. However, the time complexity of the algorithm is $O(mn)$, i.e. linear in the number of vertices of $R(m, n)$. □

Algorithm 4.1 The longest path algorithm

```
  procedure LongestPath($R(m, n), s, t$)
 1: if $R$ is 1-rectangle or 2-rectangle then
 2:    return the solution of $(R(m, n), s, t)$ based on Lemma 4.1
 3: else
 4:    if $R$ can be stripped then
 5:       let $S$ be a strip of $R$
 6:       $P \leftarrow$ LongestPath($R - S, s, t$)
 7:       $C \leftarrow$ HamiltonianCycle($S$)
 8:       return MergeStrip($P, C, s, t$)
 9:    else
10:       if $R$ can be split then
11:          let $R$ be split to $R_p$ and $R_q$
12:          $P_1 \leftarrow$ LongestPath($R_p, s, p$)
13:          $P_2 \leftarrow$ LongestPath($R_q, q, t$)
14:          return MergeSplit ($P_1, P_2, p, q$)
15:       else
16:          /* $(R(m, n), s, t)$ is a prime problem */
17:          return the solution of $(R(m, n), s, t)$ based on Lemma 4.6
18:       end if
19:    end if
20: end if
```

As a complete example, the construction of a longest path between s and t in $R(12, 5)$ is shown step by step in Fig. 13.

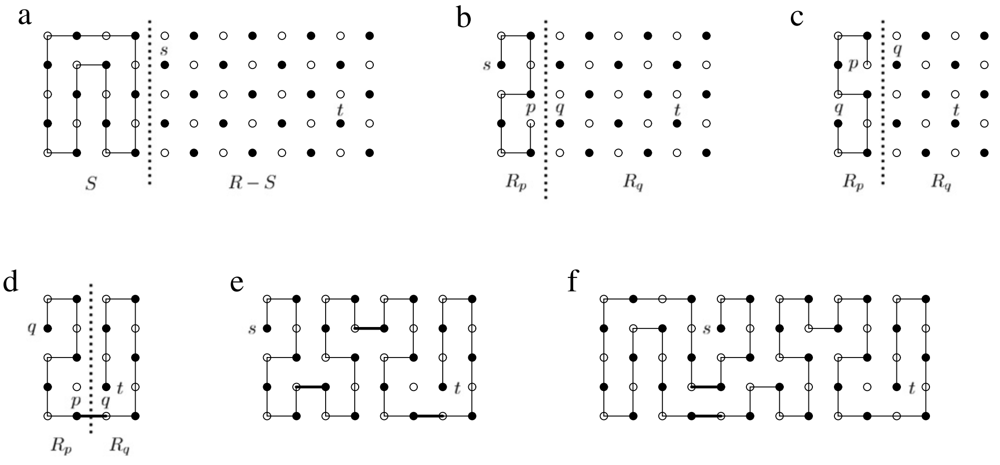

Fig. 13. Example of $R(12, 5)$ with $L = mn - 1$. (a) Stripping, by S. ((b)–(d)) Three consequent splits. (e) A longest path of $R - S$. (f) A solution of $P(R, s, t)$.

5. Conclusion and future work

We presented a linear-time algorithm for finding a longest path in a rectangular grid graph between any two given vertices. Since the longest path problem is NP-hard in general grid graphs [10], it remains open whether the problem is polynomially solvable in solid grid graphs.

Acknowledgments

The authors would like to thank the referees for a very careful reading and many helpful comments and suggestions, which have led to great improvement of the paper.

References

[1] A. Björklund, T. Husfeldt, Finding a path of superlogarithmic length, SIAM J. Comput. 32 (6) (2003) 1395–1402.
[2] R.W. Bulterman, F.W. van der Sommen, G. Zwaan, T. Verhoeff, A.J.M. van Gasteren, W.H.J. Feijen, On computing a longest path in a tree, Inform. Process. Lett. 81 (2) (2002) 93–96.
[3] S.D. Chen, H. Shen, R. Topor, An efficient algorithm for constructing Hamiltonian paths in meshes, J. Parallel Comput. 28 (9) (2002) 1293–1305.
[4] R. Diestel, Graph Theory, Springer, New York, 2000.
[5] T. Feder, R. Motwani, Finding large cycles in Hamiltonian graphs, in: Proc. 16th Annual ACM–SIAM Symp. on Discrete Algorithms, SODA, ACM, 2005, pp. 166–175.
[6] H.N. Gabow, Finding paths and cycles of superpolylogarithmic length, in: Proc. 36th Annual ACM Symp. on Theory of Computing, STOC, ACM, 2004, pp. 407–416.
[7] H.N. Gabow, S. Nie, Finding long paths, cycles and circuits, in: 19th annual International Symp. on Algorithms and Computation, ISAAC, in: LNCS, vol. 5369, 2008, pp. 752–763.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
[9] G. Gutin, Finding a longest path in a complete multipartite digraph, SIAM J. Discrete Math. 6 (2) (1993) 270–273.
[10] A. Itai, C. Papadimitriou, J. Szwarcfiter, Hamiltonian paths in grid graphs, SIAM J. Comput. 11 (4) (1982) 676–686.
[11] D. Karger, R. Montwani, G.D.S. Ramkumar, On approximating the longest path in a graph, Algorithmica 18 (1) (1997) 82–98.
[12] W. Lenhart, C. Umans, Hamiltonian cycles in solid grid graphs, in: Proc. 38th Annual Symposium on Foundations of Computer Science, FOCS'97, 1997, pp. 496–505.
[13] K. Ioannidou, G.B. Mertzios, S. Nikolopoulos, The longest path problem is polynomial on interval graphs, in: Proc. of 34th Int. Symp. on Mathematical Foundations of Computer Science, vol. 5734, Springer-Verlag, Novy Smokovec, High Tatras, Slovakia, 2009, pp. 403–414.
[14] F. Luccio, C. Mugnia, Hamiltonian paths on a rectangular chessboard, in: Proc. 16th Annual Allerton Conference, 1978, pp. 161–173.
[15] G.B. Mertzios, D.G. Corneil, A simple polynomial algorithm for the longest path problem on cocomparability graphs, in: Proc. of CoRR, 2010.
[16] R. Uehara, Y. Uno, On computing longest paths in small graph classes, Internat. J. Found. Comput. Sci. 18 (5) (2007) 911–930.
[17] Z. Zhang, H. Li, Algorithms for long paths in graphs, Theoret. Comput. Sci. 377 (1–3) (2007) 25–34.