Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

An approximation algorithm for the longest cycle problem in solid grid graphs

Asghar Asgharian Sardroud, Alireza Bagheri *

Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 7 October 2014
Received in revised form 5 August 2015
Accepted 5 October 2015
Available online xxxx

Keywords:
Longest cycle
Hamiltonian cycle
Approximation algorithm
Solid grid graph

ABSTRACT

Although, the Hamiltonicity of solid grid graphs are polynomial-time decidable, the complexity of the longest cycle problem in these graphs is still open. In this paper, by presenting a linear-time constant-factor approximation algorithm, we show that the longest cycle problem in solid grid graphs is in APX. More precisely, our algorithm finds a cycle of length at least $\frac{2n}{3}+1$ in 2-connected n-node solid grid graphs.

1. Introduction

The longest cycle and path problems are well-known NP-hard problems in graph theory. There are various results which show that these problems are hard to approximate in general graphs. For example, assuming that P $\neq$ NP, it has been shown that there is no polynomial-time constant-factor approximation algorithm for the longest path problem and also it is not possible to find a path of length $n-n^{\epsilon}$ in polynomial-time in Hamiltonian graphs [14]. The Color coding technique introduced by Alon et al. [1] is one of the first approximation algorithms for these problems which can find paths and cycles of length $\log n$. Later, Björklund et al. introduced another technique with better approximation ratio, i.e. $O(n \log\log n/\log^2 n)$, for finding long paths [3,9]. To our knowledge, the result of Gabow [8], which can find a cycle or path of length $\exp(\Omega(\sqrt{\log l/\log\log l}))$ in graphs with the longest cycle of length l, is the best polynomial-time approximation algorithm for finding the longest cycles. The results also show that these problems are hard to approximate even in bounded-degree and Hamiltonian graphs [6,7]. These problems are even harder to approximate in the case of directed graphs as showed in [4]. For more related results on approximation algorithms on general graphs, see [2,10,17].

There are few classes of graphs in which the longest path or the longest cycle problems are polynomial [5,11,12,15,16,18]. In the case of grid graphs, Itai et al. [13] showed that the Hamiltonian path and cycle problems are NP-complete. Grid graphs are vertex-induced subgraphs of the infinite integer grid G^{∞}. Later, Umans et al. showed that the Hamiltonicity of solid grid graphs, i.e. the grid graphs in which each internal face has length four, is decidable in polynomial time [19]. However, to our knowledge, there is no result on finding or approximating the longest cycle in this class of graphs, but there is only a $\frac{5}{6}$-approximation algorithm for finding the longest paths in grid graphs that have square-free cycle covers [20]. In this paper, we introduce a linear-time constant-factor approximation algorithm for the longest cycle problem in solid grid graphs. Our algorithm first finds a vertex-disjoint cycle set containing at least $\frac{2n}{3}+1$ of the vertices of a given 2-connected, n-vertex solid grid graph and then merge them into a single cycle.

* Corresponding author.
E-mail addresses: asgharian@aut.ac.ir (A. Asgharian Sardroud), ar_bagheri@aut.ac.ir (A. Bagheri).

http://dx.doi.org/10.1016/j.dam.2015.10.022

We organized the paper as follows. In Section 2, we present the terminology and some preliminary concepts. Our algorithm for finding the cycle set of the desired length is given in Section 3, and in Section 4 we show that these cycles can be merged into a single cycle of the same size. Finally, in Section 5 we conclude the paper.

2. Preliminaries

In this section, we present the definitions which is used during the paper and the necessary concepts about solid grid graphs. Grid graphs are vertex-induced subgraphs of the infinite integer grid whose vertices are the integer coordinated points of the plane and there is an edge between any two vertices of unit distance. Let G be a solid grid graph, i.e. a grid graph that has no inner face of length more than four. We consider solid grid graphs as plane graphs, considering their natural embedding on the integer grid. The vertices of G adjacent to the outer face are called *boundary vertices*, and the set of boundary vertices of G form its *boundary*. The boundary of connected plane graph G should be a closed walk, i.e. a cycle in which vertices and edges may be repeated, which is called *boundary walk* (considering that a single vertex is a walk of length zero). We use $|W|$ to refer to the number of (not necessarily distinct) edges of a closed walk W. Each *cut vertex* of G, i.e. a vertex of G that its removal makes G disconnected, is a repeated vertex in its boundary walk and vice versa, therefore, G is 2-connected, if and only if its boundary is a cycle. If G is not connected, its boundary should be a set of closed walks, i.e. the set of boundary walks of its connected components. Let cycle C be the boundary of G. We say a vertex of boundary cycle C is *convex vertex*, *flat vertex* or *concave vertex* respectively if its degree in G is two, three or four. The embedding of any cycle of the plane graph G is a simple rectilinear polygon, as a result, in each cycle of G the number of convex vertices should be four more than the number of concave vertices. Also, note that, because solid grid graphs are vertex-induced, their boundary cycles cannot contain two consecutive concave vertices. We define two edges of G to be *parallel edges* if they are not incident to a common vertex, but both of them are adjacent to the same inner face. When G' is a subgraph of G, we use the notation $G \setminus G'$ to denote the graph obtained from G after removing all the vertices of G' and their incident edges. It is easy to show that $G \setminus G'$ is also a solid grid graph, when G' is the boundary of G, or it is a maximal 2-connected subgraph of G.

3. Finding the cycle set

Let G be a 2-connected, n-node solid grid graph and C be its boundary cycle. Given such a graph G, we present an algorithm that finds a set of vertex-disjoint cycles S in G containing at least $\frac{2n}{3} + 1$ of the vertices of G. In the next section, by merging these cycles, we construct a cycle of the desired length.

Let S be initially empty. We add C to S, and since the $G \setminus C$ may be not 2-connected, we repeat the process recursively on its disjoint 2-connected subgraphs. Let $\{G'_1, \dots, G'_m\}$ be a maximal set of disjoint maximal 2-connected subgraphs of $G \setminus C$. The pseudocode of the procedure for constructing cycle set S is given in Algorithm 3.1.

Algorithm 3.1 The algorithm of finding the cycle set S

procedure FindCycleSet(G)
1: **if** $|G| < 4$
2:     **return** $\varnothing$
3: **let** C be the boundary cycle of G
4: **let** $\{G'_1, \dots, G'_m\}$ be the set of subgraphs of G as defined above
5: $S \leftarrow \{C\}$
6: **foreach** G'_i, $1 \le i \le m$ **do**
7:     **if not** G'_i is a connected component of $G \setminus C$ having $|G'_i| = 4$
8:         $S \leftarrow S \cup$ FindCycleSet(G'_i)
9: **return** S

The line 7 of the algorithm, excludes some length four cycles from S, because these cycles may be unmergable in the next step of our algorithm. Also, a maximal set $\{G'_1, \dots, G'_m\}$ of disjoint maximal 2-connected subgraphs of $G \setminus C$, used in the algorithm, can be constructed as follows. Because $G \setminus C$ is a solid grid graph (it is planar and each of its inner faces has length four), we can easily find its connected components and cut vertices only by checking its boundary vertices. Clearly, any 2-connected component of $G \setminus C$ is a maximal 2-connected subgraph without any overlap with other such subgraphs of $G \setminus C$. Moreover, the isolated vertices, the vertices of degree one and the cut vertices of degree two in $G \setminus C$ cannot be in any 2-connected subgraph of $G \setminus C$, so we can discard them easily. The reminder of $G \setminus C$ consists of some maximal 2-connected subgraphs of $G \setminus C$ that may overlap with each other only at the cut vertices of G. Therefore, we can construct a maximal set of disjoint maximal 2-connected subgraphs from the reminder of $G \setminus C$, only by checking the cut vertices of $G \setminus C$.

The following lemma shows that the sum of the lengths of the cycles in S is at least $\frac{2n}{3} + 1$.

Lemma 3.1. *The sum of the length of the cycles in S is at least $\frac{2n}{3} + 1$.*

Please cite this article in press as: A. Asgharian Sardroud, A. Bagheri, An approximation algorithm for the longest cycle problem in solid grid graphs, Discrete Applied Mathematics (2015), http://dx.doi.org/10.1016/j.dam.2015.10.022