Theoretical Computer Science 521 (2014) 1–7

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Recognizing some complementary products

Márcia R. Cappelle [a], Lucia Penso [b], Dieter Rautenbach [b,*]

[a] Instituto de Informática, Universidade Federal de Goiás, Goiânia, Brazil
[b] Institut für Optimierung und Operations Research, Universität Ulm, Ulm, Germany

ARTICLE INFO

Article history:
Received 6 December 2012
Accepted 12 November 2013
Communicated by P.G. Spirakis

Keywords:
Combinatorial problems
Graph algorithms
Cartesian product
Complementary product
Complementary prism

ABSTRACT

For two graphs G and H, a set R of vertices of G, and a set S of vertices of H, the complementary product $G(R)\Box H(S)$ is defined as the graph with vertex set $V(G)\times V(H)$ where two vertices (u,v) and (x,y) are adjacent exactly if either $u=x$, $u\in R$, and $vy\in E(H)$; or $u=x$, $u\in V(G)\setminus R$, and $vy\notin E(H)$; or $v=y$, $v\in S$, and $ux\in E(G)$; or $v=y$, $v\in V(H)\setminus S$, and $ux\notin E(G)$.

We show that for a fixed connected graph H and a fixed non-empty and proper subset S of its vertex set, it can be decided in polynomial time whether, for a given input graph F, there exists a graph G such that F is isomorphic to $G(V(G))\Box H(S)$. Furthermore, if such a graph G exists, then one such graph can be found in polynomial time. Our result gives a positive answer to a question posed by Haynes, Henning, Slater, and van der Merwe [5].

1. Introduction

We consider only finite, simple, and undirected graphs and use standard terminology.

In [5] Haynes, Henning, Slater, and van der Merwe, introduce the so-called *complementary product* as a generalization of the well-known *Cartesian product* of graphs. For two graphs G and H, a set R of vertices of G, and a set S of vertices of H, the *complementary product* $G(R)\Box H(S)$ is defined as the graph with vertex set $V(G)\times V(H)$ where two vertices (u,v) and (x,y) are adjacent exactly if

- either $u=x$, $u\in R$, and $vy\in E(H)$,
- or $u=x$, $u\in V(G)\setminus R$, and $vy\notin E(H)$,
- or $v=y$, $v\in S$, and $ux\in E(G)$,
- or $v=y$, $v\in V(H)\setminus S$, and $ux\notin E(G)$.

Note that $G(R)\Box H(S)$ coincides with the Cartesian product of G and H if $R=V(G)$ and $S=V(H)$.

In [5] the authors concentrate on graphs of the form $G(V(G))\Box K_2(S)$ with $|S|=1$ for which they study degrees, distances, independence, and domination. They call the graph $G(V(G))\Box K_2(S)$ *the complementary prism of* G and denote it by $G\bar{G}$. Some few further papers [2,8,9] study similar aspects of complementary prisms.

At the end of [5] Haynes et al. ask whether complementary prisms can be recognized in polynomial time. We give a positive answer to their question. For a fixed graph H and a fixed subset S of its vertex set, we consider the following more general problem, which comprises the recognition of complementary prisms as a special case.

* Corresponding author.
E-mail addresses: marcia@inf.ufg.br (M.R. Cappelle), lucia.penso@uni-ulm.de (L. Penso), dieter.rautenbach@uni-ulm.de (D. Rautenbach).

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.11.006

Complementary Product with $H(S)$

Instance: A graph F.
Task: Determine whether there is a graph G such that F is isomorphic to $G(V(G))\square H(S)$ and determine such a graph if it exists.

Our main result is the following.

Theorem 1. *For a fixed connected graph H and a fixed non-empty and proper subset S of its vertex set,* Complementary Product with $H(S)$ *can be solved in polynomial time.*

It makes sense to relate our result to complexity issues of the Cartesian product. In [10,11] Sabidussi and Vizing showed that every connected graph has a factorization into prime factors with respect to the Cartesian product that is unique up to a permutation of the factors. A first polynomial time algorithm to determine this factorization was proposed by Feigenbaum et al. [4]. Considerable simplifications and running time improvements were obtained by Winkler [12], Feder [3], and Aurenhammer et al. [1] culminating in a linear time algorithm by Imrich et al. [7]. For disconnected graphs, factorization with respect to the Cartesian product is at least as hard as the graph isomorphism problem. For elegant proofs and an excellent survey we refer the reader to [6].

In view of Complementary Product with $H(S)$ and Theorem 1, the above discussion has several implications. If $S \in \{\emptyset, V(H)\}$, then the complementary product $G(V(G))\square H(S)$ reduces to a Cartesian product. Therefore, to require that S is a non-empty and proper subset of the vertex set of H is a reasonable restriction. If $H = \bar{K}_2$ and $|S| = 1$, then a polynomial time algorithm for Complementary Product with $H(S)$ would lead to a polynomial time algorithm for the graph isomorphism problem. In fact, two given graphs G_1 and G_2 are isomorphic if and only if $G_1 \cup \bar{G}_2$ is of the form $G(V(G))\square\bar{K}_2(S)$ with $|S| = 1$. Hence is makes sense to require that H is connected.

In order to prove Theorem 1, we describe a corresponding efficient algorithm in the next section.

2. Algorithm for Complementary Product with $H(S)$

Throughout the rest of the paper let H be a fixed connected graph and let S be a fixed non-empty and proper subset of its vertex set. We assume that $V(H) = \{v_1, \dots, v_h\}$ and $S = \{v_1, \dots, v_s\}$. Let $\bar{S} = V(H) \setminus S$. Since H is connected, we may assume that $v_s v_{s+1}$ is an edge of H. We denote the set $\{1, 2, \dots, n\}$ by $[n]$.

Let the graph F be an instance of Complementary Product with $H(S)$.

Our approach is to assume that F equals $G(V(G))\square H(S)$ for some graph G and to try to efficiently reconstruct G by spending polynomial effort for each of a polynomial number of partial solutions.

Let $V(G) = \{u_1, \dots, u_n\}$.

For $i \in [n]$, the i-th *row of* $G(V(G))\square H(S)$ is the subgraph of $G(V(G))\square H(S)$ induced by $\{u_i\} \times V(H)$, which is isomorphic to H. Similarly, for $j \in [h]$, the j-th *column of* $G(V(G))\square H(S)$ is the subgraph of $G(V(G))\square H(S)$ induced by $V(G) \times \{v_j\}$, which is isomorphic to G if $j \in [s]$ and to $\bar{G}$ if $j \in [h] \setminus [s]$. An edge of $G(V(G))\square H(S)$ between vertices in different rows is *vertical* and an edge of $G(V(G))\square H(S)$ between vertices in different columns is *horizontal*. Let $w(i, j) = (u_i, v_j)$.

In order to solve Complementary Product with $H(S)$ for F we need to identify the vertex of F that corresponds to $w(i, j)$ for each $(i, j) \in [n] \times [h]$, or to find out that no such identification leads to the desired factorization. We say that a vertex of F has been *identified*, once we decided which vertex $w(i, j)$ it should corresponds to.

Throughout the exposition of the algorithm we illustrate the information extracted until a certain point using rectangular tables as in Figs. 1–4. The rows and columns of these tables correspond to the rows and columns of $G(V(G))\square H(S)$ and every entry is initially filled with a "?". Whenever a vertex of F is identified to correspond to some $w(i, j)$, we replace the "?" in row i and column j with "$w(i, j)$".

Our algorithm proceeds along to the following steps.

- We guess which h vertices of F correspond to the first row $(w(1, 1), \dots, w(1, h))$.
  Clearly, there are $O(n^{O(h)})$ such guesses.
- We identify all neighbors of vertices in the first row that are in rows 2 to n.
  This step will require $O(n^{O(h)})$ time.
- If a vertex of F that has still not been identified has at least two identified neighbors in one column, then it must lie in the same column. We iteratively identify such vertices until every unidentified vertex has at most one identified neighbor in each column. We argue that at this point all but $O(h)$ many vertices either in the first s columns or in the last $h - s$ columns have been identified.
  This step will require $O(n^{O(h)})$ time.
- At this point we may assume, by symmetry, that all but $O(h)$ many vertices in the last $h - s$ columns have been identified. We guess all these $O(h)$ unidentified vertices.
  Clearly, there are $O(n^{O(h)})$ such guesses.