Discrete Applied Mathematics 190–191 (2015) 100–108

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Knocking out P_k-free graphs[☆]

Matthew Johnson, Daniël Paulusma[*], Anthony Stewart

School of Engineering and Computing Sciences, Durham University, South Road, Durham, DH1 3LE, UK

ARTICLE INFO

Article history:
Received 28 October 2014
Received in revised form 1 April 2015
Accepted 17 April 2015
Available online 8 May 2015

Keywords:
Parallel knock-out schemes
Hamilton cycle
Cographs
Split graphs
Computational complexity

ABSTRACT

A parallel knock-out scheme for a graph proceeds in rounds in each of which each surviving vertex eliminates one of its surviving neighbours. A graph is KO-reducible if there exists such a scheme that eliminates every vertex in the graph. The Parallel Knock-Out problem is to decide whether a graph G is KO-reducible. This problem is known to be NP-complete and has been studied for several graph classes. We show that the problem is NP-complete even for split graphs, a subclass of P_5-free graphs. In contrast, our main result is that it is linear-time solvable for P_4-free graphs (cographs).

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

We consider *parallel knock-out schemes* for finite undirected graphs with no self-loops and no multiple edges. These schemes, which were introduced by Lampert and Slater [17], proceed in rounds. In the first round each vertex in the graph selects exactly one of its neighbours, and then all the selected vertices are eliminated simultaneously. In subsequent rounds this procedure is repeated in the subgraph induced by those vertices not yet eliminated. The scheme continues until there are no vertices left, or until an isolated vertex is obtained (since an isolated vertex will never be eliminated). A graph is called *KO-reducible* if there exists a parallel knock-out scheme that eliminates the whole graph. The *parallel knock-out number* of a graph G, denoted by $\text{pko}(G)$, is the minimum number of rounds in a parallel knock-out scheme that eliminates every vertex of G. If G is not KO-reducible, then $\text{pko}(G) = \infty$.

Examples. Every graph G with a hamiltonian cycle has $\text{pko}(G) = 1$, as each vertex can select its successor on a hamiltonian cycle C of G after fixing some orientation of C. Also every graph G with a perfect matching has $\text{pko}(G) = 1$, as each vertex can select its matching neighbour in the perfect matching. In fact it is not difficult to see [2] that a graph G has $\text{pko}(G) = 1$ if and only if G contains a *[1,2]-factor*, that is, a spanning subgraph in which every component is either a cycle or an edge.

We study the computational complexity of the Parallel Knock-Out problem, which is the problem of deciding whether a given graph is KO-reducible. Our main motivation is the close relationship with cycles and matchings as illustrated by the above examples. We also consider the variant in which the number of rounds permitted is fixed. This problem is known as the k-Parallel Knock-Out problem, which has as input a graph G and ask whether $\text{pko}(G) \leq k$ for some fixed integer k (i.e. that is not part of the input).

Known Results. The 1-Parallel Knock-Out problem is equivalent [2] to testing whether a graph has a [1, 2]-factor, which is well-known to be polynomial-time solvable (see e.g. [4] for a proof). However, both the problems Parallel Knock-Out and

[☆] An extended abstract of this paper appeared in the proceedings of MFCS 2014 [15].

[*] Corresponding author.
E-mail addresses: matthew.johnson2@durham.ac.uk (M. Johnson), daniel.paulusma@durham.ac.uk (D. Paulusma), a.g.stewart@durham.ac.uk (A. Stewart).

http://dx.doi.org/10.1016/j.dam.2015.04.010
0166-218X/© 2015 Elsevier B.V. All rights reserved.

k-Parallel Knock-Out with $k \geq 2$ are NP-complete even for bipartite graphs [4]. On the other hand, it is known that Parallel Knock-Out and k-Parallel Knock-Out (for all $k \geq 1$) can be solved in $O(n^{3.5}\log^2 n)$ time on trees [2]. These results were later extended to graph classes of bounded treewidth [4]. It remains *open* whether a further generalization is possible to graph classes of bounded clique-width. Broersma et al. in [3] gave an $O(n^{5.376})$ time algorithm for solving Parallel Knock-Out on n-vertex claw-free graphs. Later this was improved to an $O(n^2)$ time algorithm for almost claw-free graphs (which generalize the class of claw-free graphs) [16]. The latter paper also gives a full characterization of connected almost claw-free graphs that are KO-reducible. In particular it shows that every KO-reducible almost claw-free graph has parallel knock-out number at most 2. In general, KO-reducible graphs (even KO-reducible trees [2]) may have an arbitrarily large parallel knock-out number. Broersma et al. [3] showed that a KO-reducible n-vertex graph G has $\mathrm{pko}(G) \leq \min\{-\frac{1}{2} + (2n - \frac{7}{4})^{\frac{1}{2}}, \frac{1}{2} + (2\alpha - \frac{7}{4})^{\frac{1}{2}}\}$ (where α denotes the size of a largest independent set in G). This bound is asymptotically tight for complete bipartite graphs [2]. Broersma et al. [3] also showed that every KO-reducible graph with no induced $(p+1)$-vertex star $K_{1,p}$ has parallel knock-out number at most $p-1$.

Our Results. To date the only graph classes of unbounded tree-width for which Parallel Knock-Out is known to be polynomial-time solvable are complete bipartite graphs [2] and almost claw-free graphs [16], and we aim to identify further such classes. In particular we want to address the open problem of whether Parallel Knock-Out is polynomial-time solvable on graph classes whose clique-width is bounded by a constant. This seems a very challenging problem, and in this paper we focus on graphs of clique-width at most 2 (which may have arbitrarily large tree-width). It is known that a graph has clique-width at most 2 if and only if it is a cograph [7]. Cographs are also known as P_4-free graphs (a graph is called P_k-free if it has no induced k-vertex path).

In Section 3 we give a linear-time algorithm for solving the Parallel Knock-Out problem on cographs. The first step of the algorithm is to compute the cotree of a cograph. It then traverses the cotree twice. The first time to compute to what extent "large" subgraphs can be reduced by themselves and how many free "firings" from outside are available. The second time to check whether the number of free external firings is sufficient to knock them out. In this way it will be verified whether the whole graph is KO-reducible. In Section 4 we prove that both the Parallel Knock-Out problem and the k-Parallel Knock-Out problem ($k \geq 2$) are NP-complete even for split graphs. Because split graphs are P_5-free, our results imply a dichotomy result for the computational complexity of the Parallel Knock-Out problem restricted to P_k-free graphs, as shown in Section 5, where we also give some (other) open problems.

2. Preliminaries

We denote a graph by $G = (V(G), E(G))$ and write $|G| = |V(G)|$ to denote the order of G. An edge joining vertices u and v is denoted by uv. If not stated otherwise a graph is assumed to be finite, undirected and simple.

Let $G = (V, E)$ be a graph. The *neighbourhood* of $u \in V$, that is, the set of vertices adjacent to u is denoted by $N_G(u) = \{v \mid uv \in E\}$. For a subset $S \subseteq V$, we let $G[S]$ denote the *induced* subgraph of G, which has vertex set S and edge set $\{uv \in E \mid u, v \in S\}$. A set $I \subseteq V$ is called an *independent set* of G if no two vertices in I are adjacent to each other. A subset $C \subseteq V$ is called a *clique* of G if any two vertices in C are adjacent to each other. A subset $D \subseteq V$ is a *dominating set* of a graph $G = (V, E)$ if every vertex of G is in D or adjacent to a vertex in D.

The *union* of two graphs G and H is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. If $V(G) \cap V(H) = \emptyset$, then we say that the union of G and H is *disjoint* and write $G + H$. We denote the disjoint union of r copies of G by rG.

For $n \geq 1$, the graph P_n denotes the *path* on n vertices, that is, $V(P_n) = \{u_1, \ldots, u_n\}$ and $E(P_n) = \{u_iu_{i+1} \mid 1 \leq i \leq n-1\}$. For $n \geq 3$, the graph C_n denotes the *cycle* on n vertices, that is, $V(C_n) = \{u_1, \ldots, u_n\}$ and $E(C_n) = \{u_iu_{i+1} \mid 1 \leq i \leq n-1\} \cup \{u_nu_1\}$. The graph K_n denotes the *complete graph* on n vertices, that is, the n-vertex graph whose vertex set is a clique. A graph is *complete bipartite* if its vertex set can be partitioned into two classes such that two vertices u and v are adjacent if and only if u and v belong to different classes. The graph $K_{p,q}$ is the *complete bipartite graph* with partition classes of sizes p and q, respectively; the graph $K_{1,q}$ is also called the *star* on $q+1$ vertices.

Let G be a graph and let $\{H_1, \ldots, H_p\}$ be a set of graphs. We say that G is $(H_1, \ldots, H_p)$*-free* if G has no induced subgraph isomorphic to a graph in $\{H_1, \ldots, H_p\}$. If $p = 1$ we may write H_1-free instead of (H_1)-free. A P_4-free graph is also called a *cograph*. A graph G is a *split graph* if its vertex set can be partitioned into a clique and an independent set. Split graphs coincide with $(2K_2, C_4, C_5)$-free graphs [10]; note that this implies that every split graph is $2K_2$-free and thus P_5-free.

We also need some formal terminology for parallel knock-out schemes. For a graph $G = (V, E)$, a *KO-selection* is a function $f : V \to V$ with $f(v) \in N(v)$ for all $v \in V$. If $f(v) = u$, we say that vertex v *fires at* vertex u, or that u *is knocked out* by a *firing* of v. If $u \in U$ for some $U \subseteq V$ then the firing is said to be *internal* with respect to U if $v \in U$; otherwise it is said to be *external* (with respect to U).

For a KO-selection f, we define the corresponding *KO-successor* of G as the subgraph of G that is induced by the vertices in $V \setminus f(V)$; if G' is the KO-successor of G we write $G \rightsquigarrow G'$. Note that every graph without isolated vertices has at least one KO-successor. A sequence

$$G \rightsquigarrow G^1 \rightsquigarrow G^2 \rightsquigarrow \cdots \rightsquigarrow G^s,$$

is called a *parallel knock-out scheme* or *KO-scheme*. A KO-scheme in which G^s is the null graph $(\emptyset, \emptyset)$ is called a *KO-reduction scheme*; in that case G is also called *KO-reducible*. A single step in a KO-scheme is called a *(firing) round*. Recall that the

Download English Version:

https://daneshyari.com/en/article/6872116

Download Persian Version:

https://daneshyari.com/article/6872116

Daneshyari.com