

Eliminating graphs by means of parallel knock-out schemes

Hajo Broersma^{a,b}, Fedor V. Fomin^c, Rastislav Kráľovič^{d,1}, Gerhard J. Woeginger^e

^aDepartment of Computer Science, University of Durham, Science Labs, South Road, DH1 3LE Durham, UK

^bCenter for Combinatorics, Nankai University, Tianjin 300071, China

^cDepartment of Informatics, University of Bergen, N-5020 Bergen, Norway

^dDepartment of Computer Science, Comenius University, 842 48 Bratislava, Slovak Republic

^eDepartment of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Received 28 September 2004; received in revised form 12 October 2005; accepted 20 April 2006

Available online 22 June 2006

Abstract

In 1997 Lampert and Slater introduced parallel knock-out schemes, an iterative process on graphs that goes through several rounds. In each round of this process, every vertex eliminates exactly one of its neighbors. The parallel knock-out number of a graph is the minimum number of rounds after which all vertices have been eliminated (if possible). The parallel knock-out number is related to well-known concepts like perfect matchings, hamiltonian cycles, and 2-factors.

We derive a number of combinatorial and algorithmic results on parallel knock-out numbers: for families of sparse graphs (like planar graphs or graphs of bounded tree-width), the parallel knock-out number grows at most logarithmically with the number n of vertices; this bound is basically tight for trees. Furthermore, there is a family of bipartite graphs for which the parallel knock-out number grows proportionally to the square root of n . We characterize trees with parallel knock-out number at most 2, and we show that the parallel knock-out number for trees can be computed in polynomial time via a dynamic programming approach (whereas in general graphs this problem is known to be NP-hard). Finally, we prove that the parallel knock-out number of a claw-free graph is either infinite or less than or equal to 2.

© 2006 Elsevier B.V. All rights reserved.

MSC: 05C75; 05C35; 05C85; 68R10

Keywords: Knock-out number; Parallel knock-out scheme; Hamiltonian cycle; Perfect matching; Tree; Claw-free graph; Computational complexity; Dynamic programming

1. Introduction

Lampert and Slater [4] introduced the following *parallel knock-out* procedure for graphs: on every vertex v of an undirected graph, there is a person standing. Every person selects precisely one other person that stands on an adjacent vertex. Then all the selected persons (vertices) are knocked out (deleted) simultaneously, and the whole procedure is repeated with the surviving vertices. The procedure terminates, as soon as there are survivors that do not have any neighbor left to knock out. For instance, on the path $v_1 - v_2 - v_3 - v_4 - v_5$ it may happen that the persons on v_1 and

¹ Research supported by Grant APVT-20-018902.

E-mail addresses: hajo.broersma@durham.ac.uk (H. Broersma), fomin@ii.uib.no (F.V. Fomin), kralovic@dcs.fmph.uniba.sk (R. Kráľovič), gwoegi@win.tue.nl (G.J. Woeginger).

v_3 both decide to knock out the person on v_2 , that v_2 and v_4 both decide to knock out v_3 , and that v_5 knocks out v_4 . Then v_1 and v_5 are the survivors of this round, and the procedure terminates.

Formally, let $G = (V, E)$ be an undirected, simple, loopless graph. We refer to [2] for undefined terminology. For convenience we allow (\emptyset, \emptyset) as a graph, and call it the *empty graph*. We denote by $N(v)$ the set of all neighbors of vertex v (not including the vertex v itself). A *KO-selection* is a function $f : V \rightarrow V$ with $f(v) \in N(v)$ for all $v \in V$. If $f(v) = u$, we will sometimes say that vertex v *knocks out* or *eliminates* vertex u , or that (in the language of chip firing games) vertex v *fires at* vertex u . For a KO-selection f , we define the corresponding *KO-successor* G^f of G as the subgraph of G that is induced by the vertices in $V - f(V)$; this situation will be denoted by $G \rightsquigarrow G^f$. Note that every nonempty graph G without isolated vertices has at least one KO-successor.

In this paper, we are mainly interested in the question whether for a given graph G , there exists a sequence of KO-selections and KO-successors such that

$$G \rightsquigarrow G_1 \rightsquigarrow G_2 \rightsquigarrow \dots \rightsquigarrow G_r = (\emptyset, \emptyset).$$

If no such sequence exists, then the *parallel knock-out number* of the graph G is infinite, and we write $\text{PKO}(G) = \infty$. Otherwise, we define the parallel knock-out number $\text{PKO}(G)$ of G as the smallest number r for which such a sequence exists. A graph G is called *KO-reducible* if and only if $\text{PKO}(G)$ is finite.

A sequence of KO-selections that transform a KO-reducible graph into the empty graph is called a *KO-reduction scheme*. A single step in this sequence is called a *round* of the KO-reduction scheme.

It has been announced by Lampert and Slater [4] that it is an NP-complete problem to decide whether a given graph is KO-reducible. The complexity of deciding whether $\text{PKO}(G) \leq k$ (or $\text{PKO}(G) = k$) for some fixed value $k \geq 2$ that is not part of the input has been studied in [1]. Both problems turn out to be NP-complete for any fixed $k \geq 2$. It is not difficult to check that a graph G has $\text{PKO}(G) = 1$ if and only if G contains a spanning subgraph consisting of a number of mutually disjoint cycles and matching edges, which is sometimes called a $[1, 2]$ -factor. The problem of deciding whether $\text{PKO}(G) = 1$ is polynomially solvable; the equivalent formulation yields a folklore problem appearing in many standard books in combinatorial optimization. The equivalence also shows that the concept of the parallel knock-out number is related to well-known concepts like perfect matching, 2-factor, and hamiltonian cycle.

2. Results of this paper

We derive a number of combinatorial and algorithmic results around parallel knock-out numbers. In Section 3 we observe that for families of sparse graphs (like planar graphs or graphs with bounded tree-width), the parallel knock-out number (if finite) grows at most logarithmically with the number n of vertices. Furthermore, we construct a family of bipartite graphs for which the parallel knock-out number grows proportionally to the square root of n . Our upper bound result on sparse graphs is basically tight for trees (up to a constant factor): Section 4 presents a corresponding lower bound construction. In Section 5 we characterize trees with parallel knock-out number at most 2. This involves a condition analogous to Hall's condition for the existence of matchings in bipartite graphs. Section 6 investigates the algorithmic behavior of the parallel knock-out number for trees: it can be computed in polynomial time via a dynamic programming approach. This seems to be one of the rare cases where a dynamic program for trees does not immediately carry over to the bounded tree-width classes: a key ingredient of our dynamic program for trees is the reduction to a number of polynomially solvable bipartite matching problems; for higher tree-widths, these bipartite matching problems have no natural polynomially solvable analogues. Section 7 turns to claw-free graphs: if G is claw-free and $\text{PKO}(G) < \infty$, then $\text{PKO}(G) \leq 2$. If G is claw-free and $\delta(G) \geq 2$, then $\text{PKO}(G) \leq 2$; the lower bound on the minimum degree cannot be weakened. We finish the paper with some open problems.

3. Upper and lower bounds

We start with a general upper bound on the parallel knock-out number (if finite) of relatively sparse graphs. In fact, the proof below shows that this is an upper bound on the number of rounds in any KO-reduction scheme for such graphs.

Lemma 1. *Let $\tau > 0$ be a fixed real number. Let G be a KO-reducible graph all of whose induced subgraphs H satisfy $|E(H)| \leq \tau |V(H)|$. Then $\text{PKO}(G) \leq \lceil 4\tau \rceil \cdot \log_2 |V|$.*

Proof. Consider a KO-reduction scheme for an n -vertex graph G that satisfies the conditions of the lemma. We claim that after $\lceil 4\tau \rceil$ rounds, the number of surviving vertices goes down by at least a factor of two. Suppose otherwise. Then for $\lceil 4\tau \rceil$ rounds, the number of vertices is at least $n/2$. In every round, each of these $n/2$ vertices fires along some edge, an edge can be used in at most one round, and every edge is used by at most two vertices. Hence, in every round at least $n/4$ edges are removed from the graph, and so after $\lceil 4\tau \rceil$ rounds the graph would be without edges. This proves our claim. The statement in the lemma now follows by induction, using that every induced subgraph of G that is the KO-successor of G in a KO-reduction scheme for G , satisfies the conditions of the lemma. \square

Lemma 1 can be used to get logarithmic upper bounds e.g., for planar graphs and for graphs of bounded tree-width. Our next result deals with an upper bound on the parallel knock-out number of complete bipartite graphs.

Lemma 2. *Let $1 \leq a \leq b$. Then the complete bipartite graph $K_{a,b}$ is KO-reducible if and only if $b \leq \frac{1}{2}a(a+1)$. If this inequality is satisfied then*

$$\text{PKO}(K_{a,b}) = \left\lceil a + \frac{1}{2} - \sqrt{\left(a + \frac{1}{2}\right)^2 - 2b} \right\rceil. \quad (1)$$

This implies $\text{PKO}(K_{a,b}) < \sqrt{2(a+b)}$.

Proof. In $K_{a,b}$ the a vertices on one side of the bipartition will be called the *left* vertices, and the other vertices will be called the *right* vertices. Suppose $K_{a,b}$ is KO-reducible. In each round of some KO-reduction scheme at least one of the still present $x \leq a$ left vertices will be eliminated and in the same round the x left vertices can eliminate at most x vertices from the right, so in total over a rounds the left vertices can eliminate at most $a + (a-1) + \dots + 1 = \frac{1}{2}a(a+1)$ right vertices. Hence we have proved one half of the first part of the lemma. The other half follows from the second part (i.e., from the fact that $\text{PKO}(K_{a,b})$ is finite if $b \leq \frac{1}{2}a(a+1)$). The situation after r rounds of some KO-reduction scheme is fully specified by the number a_r of surviving left vertices and by the number b_r of surviving right vertices. The initial situation is described by $a_0 = a$ and $b_0 = b$. We will denote the expression in the right-hand side of (1) by $F(a, b)$. We will first show that $\text{PKO}(K_{a,b}) \leq F(a, b)$ and then that $\text{PKO}(K_{a,b}) = F(a, b)$.

It can be verified that $F(a, a) = 1$ and $F(a, b) \geq 2$ if $b \geq a+1$. Hence $F(a, b) = 1$ implies $a = b$. Since $\text{PKO}(K_{a,a}) = 1$, these cases indeed satisfy $\text{PKO}(K_{a,b}) \leq F(a, b)$, and from now on we will assume $F(a, b) \geq 2$. Let $k = F(a, b) - 2$, and consider the following KO-reduction scheme: if $k \geq 1$, in the first k rounds all right vertices fire at the same left vertex, and all left vertices fire at pairwise distinct right vertices; for $r = 1, \dots, k$ this yields $a_r = a_{r-1} - 1$ and $b_r = b_{r-1} - a_{r-1}$, which is equivalent to $a_r = a - r$ and $b_r = b - r a + \frac{1}{2}r(r-1)$. It can be shown that $a_k < b_k \leq 2a_k - 1$ holds: in fact, k is chosen as the first value of r for which $b_r \leq 2a_r - 1$, and $F(a, b)$ is chosen accordingly. The KO-reduction scheme continues as follows: for $k \geq 0$, in the $(k+1)$ th round, all right vertices fire at the same left vertex, whereas the left vertices fire at $b_k - a_k + 1$ distinct right vertices. This yields $a_{k+1} = b_{k+1} = a_k - 1$. In the $(k+2)$ th round, the left and right vertices fire at each other in pairs. This shows $\text{PKO}(K_{a,b}) \leq F(a, b)$.

Next, consider $K_{a,b}$ with $1 \leq a < b \leq \frac{1}{2}a(a+1)$. For any integer t with $0 \leq t \leq a-1$, we define $g(a, t) = a + (a-1) + \dots + (a-t)$. It is routine to check that $F(a, g(a, t)) = t+1$, and that $F(a, b) = t+2$ for $g(a, t) < b \leq g(a, t+1)$ with $t \leq a-2$. Clearly, if $b > g(a, t)$, then $\text{PKO}(K_{a,b}) > t+1$, by similar arguments as before: in each round at least one of the x left vertices is eliminated and in the same round the x left vertices can eliminate at most x right vertices. It now follows that $t+1 < \text{PKO}(K_{a,b}) \leq F(a, b) = t+2$ if $g(a, t) < b \leq g(a, t+1)$, so $\text{PKO}(K_{a,b}) = F(a, b)$ for all b with $a \leq b \leq \frac{1}{2}a(a+1)$. \square

Our last result in this section gives another logarithmic upper bound for the parallel knock-out number for a class of sparse graphs, namely trees. Again this is an upper bound for the number of rounds in any KO-reduction scheme for trees, as can be seen from the proof below.

Lemma 3. *If an n -vertex tree T is KO-reducible, then $\text{PKO}(T) \leq \lceil \log_3 n \rceil$.*

Proof. The proof is done by induction on n . In fact, we prove the stronger claim that $\max(T) \leq \lceil \log_3 n \rceil$. The statement trivially holds for $n = 2$ ($\max(T) = 1$) and $n = 3$ (not KO-reducible). For the inductive argument, consider an arbitrary tree on n vertices, and let T_1, \dots, T_s be the (connected) components of some KO-successor of T in a KO-reduction scheme for T . Clearly, we have

$$\max(T) \leq 1 + \max\{\max(T_1), \dots, \max(T_s)\}. \tag{2}$$

Every vertex v in some component T_k ($1 \leq k \leq s$) must have eliminated some vertex $v' \notin V(T_k)$, and every such eliminated vertex v' itself must have fired at another vertex $v'' \notin V(T_k)$. It is easy to verify that distinct vertices u and v in T_k yield pairwise distinct vertices $u', v', u'',$ and v'' , using the fact that there are no cycles in T . Therefore, T_k contains at most $n/3$ vertices. By plugging this into (2), we complete the proof. \square

4. Trees with high parallel knock-out numbers

It is routine to check that for all KO-reducible paths P_n , $\text{PKO}(P_n) \leq 2$, and that equality holds for all odd $n \geq 7$. By this observation, one would perhaps be inclined to think that all KO-reducible trees have a bounded parallel knock-out number. This is not the case. In this section we will construct trees with arbitrarily high parallel knock-out numbers. The construction is illustrated in Fig. 1 and done inductively via the following two sequences $\langle Y_1, Y_2, \dots \rangle$ and $\langle Z_1, Z_2, \dots \rangle$ of rooted trees:

- The tree Y_1 consists of a root with one child (Y_1 is a rooted P_2).
- The tree Z_1 consists of a root with one child and one grandchild (Z_1 is a rooted P_3).
- For $\ell \geq 2$, the tree Y_ℓ consists of a root r and ℓ disjoint subtrees. The first $\ell - 2$ of these subtrees are copies of the rooted trees $Z_1, \dots, Z_{\ell-2}$; the last two of these subtrees are copies of $Z_{\ell-1}$; r is adjacent to the roots of the ℓ subtrees.
- For $\ell \geq 2$, the tree Z_ℓ consists of a root r and ℓ subtrees. These subtrees are copies of the rooted trees Y_1, \dots, Y_ℓ ; r is adjacent to the roots of the ℓ subtrees.

We are about to show that $\text{PKO}(Y_\ell) = \ell$, but we first prove some auxiliary results on possible sequences of KO-selections for Y_ℓ and Z_ℓ .

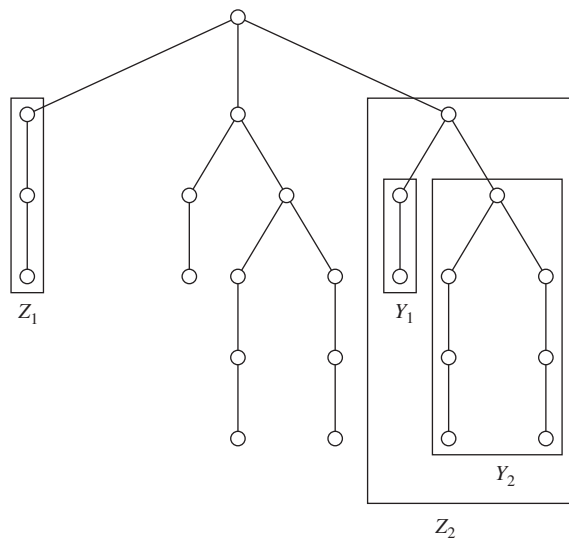


Fig. 1. The tree Y_3 .

Lemma 4. *For every $\ell \geq 1$, there exists a sequence of ℓ KO-selections that transforms Y_ℓ into the empty graph. For every $\ell \geq 1$, there exists a sequence of ℓ KO-selections that transforms Z_ℓ into a one-vertex tree that consists of the root of Z_ℓ .*

Proof. We prove both statements simultaneously by induction on ℓ . For $\ell = 1$, we let the two vertices in Y_1 eliminate each other simultaneously, and we let the root of Z_1 fire at its child, and let the child and the grandchild fire at each other.

We use the following scheme for Y_ℓ with $\ell \geq 2$. The ℓ subtrees $Z_1, \dots, Z_{\ell-2}, Z_{\ell-1}, Z_{\ell-1}$ that are attached at the children of the root use the sequence of KO-selections that exist by the inductive claim. In round k with $1 \leq k \leq \ell - 1$, the root of Y_ℓ fires at the root of the subtree Z_k (and makes this subtree disappear completely). In the final round ℓ , the root of Y_ℓ and the root of the last subtree $Z_{\ell-1}$ eliminate each other simultaneously.

We use the following scheme for Z_ℓ with $\ell \geq 2$. The ℓ subtrees Y_1, \dots, Y_ℓ that are attached at the children of the root use the sequence of KO-selections that exist by the inductive claim. In round k with $1 \leq k \leq \ell$, the root of Z_ℓ fires at the root of the subtree Y_k and makes this subtree disappear completely. After round ℓ , the root of Z_ℓ is the only survivor. \square

In the next lemma we use the term rooted maximal subtree T' of a tree T to indicate that the leaves (i.e., vertices with degree 1) of T' are also leaves in T .

Lemma 5. *Let T be a rooted tree that contains Y_ℓ (respectively Z_ℓ) as a rooted maximal subtree. Let r denote the root of Y_ℓ (respectively, of Z_ℓ). Assume furthermore that T is KO-reducible. Then in every KO-reduction scheme for T the following statements hold:*

- (a) *In the first ℓ rounds, the root r of Y_ℓ fires at its ℓ children.*
- (b) *In the ℓ th round, the root r of Y_ℓ is eliminated by one of its ℓ children.*
- (c) *The root r of Z_ℓ is not eliminated by any of its children.*
- (d) *If the root r of Z_ℓ is still alive in round $\ell + 1$, then it will fire at its father.*

Proof. We prove all four statements (a)–(d) simultaneously by induction on ℓ . In Y_1 , the leaf fires at the root in the first round. The leaf can only be eliminated, if the root fires at it in the first round. This proves (a) and (b) for $\ell = 1$. In Z_1 , the child and the grandchild must fire at each other in the first round. If the root survives the first round, it can only fire at its father in the second round. This proves (c) and (d) for $\ell = 1$.

Proofs of statements (a) and (b) for $\ell \geq 2$: The children of r form the roots of certain subtrees Z_k . By the inductive assumption of statement (c), these children are not eliminated by their own children. Hence, they all must be eliminated by r , and r must stay alive for the first ℓ rounds. By the inductive assumption of statement (d), the roots of Z_k ($1 \leq k \leq \ell - 2$) must be eliminated in the first $\ell - 2$ rounds. In rounds $\ell - 1$ and ℓ , the roots of the two subtrees $Z_{\ell-1}$ must be eliminated; this proves (a). Moreover, by (d), in round ℓ the last surviving child of r eliminates r ; this proves (b).

Proofs of statements (c) and (d) for $\ell \geq 2$: The children of r form the roots of certain subtrees Y_k . Now it is important that we have treated Y_ℓ first. By the inductive assumption of statements (a) and (b), these children only eliminate their own children, and eventually are eliminated by their own children. This proves (c). Since the root r is not eliminated by any of its children, it must be eliminated by its father. If r is still alive in round $\ell + 1$, it will fire at its last surviving neighbor, and this neighbor must be the father. This proves (d). \square

We now have the ingredients to show that there exist KO-reducible trees with arbitrarily high parallel knock-out number. This next result also shows that the upper bound in Lemma 3 is essentially best possible.

Theorem 6. *For arbitrarily large n , there exists KO-reducible n -vertex trees T with $\text{PKO}(T) = \Omega(\log n)$.*

Proof. According to Lemma 4, $\text{PKO}(Y_\ell) \leq \ell$, and according to Lemma 5(b), the root of Y_ℓ is still alive in round $\ell - 1$ of any KO-reduction scheme for Y_ℓ , so $\text{PKO}(Y_\ell) \geq \ell$. This yields $\text{PKO}(Y_\ell) = \ell$. It can be shown by induction that Y_ℓ has at most $(2 + \sqrt{2})^\ell / \sqrt{2}$ vertices, and that Z_ℓ has at most $(2 + \sqrt{2})^\ell$ vertices. \square

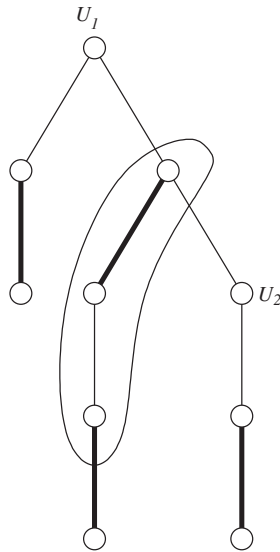


Fig. 2. Two vertices with the same triplet.

5. Trees with low parallel knock-out numbers

Recall that KO-reducible paths have parallel knock-out number at most 2. In this section we will characterize trees with parallel knock-out number at most 2. This involves a condition analogous to Hall’s condition for the existence of matchings in bipartite graphs. Here for a graph G and a matching M in G , a vertex $v \in V(G)$ is called M -saturated (or just saturated if M is understood) if v is matched to another vertex of G by an edge of M . A path P in G is called M -alternating if the edges of P are alternately in M and $E(G) \setminus M$.

We will start with an easy but useful observation, which in fact holds for arbitrary graphs.

Lemma 7. *Let T be a tree that has no matching saturating all leaves of T . Then $\text{PKO}(T) = \infty$.*

Proof. If T is KO-reducible, then in the first round of any KO-reduction scheme for T every leaf v fires at its unique neighbor u . If u does not fire at v in the first round, then after the first round v is an isolated vertex, which is not possible. Hence, if T is KO-reducible there is a matching containing one edge incident with each leaf of T . \square

Now suppose T is a KO-reducible tree, and choose a matching M of maximum cardinality subject to the condition that it saturates all leaves. The following statement is obvious:

$$\text{PKO}(T) = 1 \text{ if and only if } M \text{ is a perfect matching.} \tag{3}$$

Assuming that M is not a perfect matching we consider the set U of unsaturated (unmatched by M) vertices. Clearly, U is an independent set and by standard arguments from matching theory, the choice of M implies there are no M -alternating paths between pairs of vertices of U . For a vertex $u \in U$, a u -triplet is a P_3 (i.e., a path on three vertices) in $T - U$ with the property that one of the leaves of the P_3 is adjacent to u and the other leaf has degree at least 2 in T . Fig. 2 shows two vertices u_1 and u_2 that have the same (unique) triplet, as indicated.

Since u is not saturated by M , and since by the choice of M there are no M -alternating paths between u and another vertex that is unsaturated by M , this P_3 is an M -alternating path with two saturated leaves in T . Let $T(u)$ denote the set of u -triplets of $u \in U$ in $T - U$. Note that, with respect to a fixed matching M in a tree, a P_3 can be a u -triplet and a v -triplet for two distinct vertices u and v of U , but that in such cases u and v are adjacent to the same end vertex of the P_3 . We say that a u -triplet and v -triplet are M -disjoint if they do not share an edge of M and the end vertex of one of the triplets is not the starting vertex of the other triplet. For a subset $S \subseteq U$, a set $T(S)$ of S -triplets is a set of pairwise

M -disjoint u -triplets over all $u \in S$ in $T - U$. A tree T is called *Hall-perfect* if it has a matching saturating all leaves and for some maximum matching M with this property we have that either M is a perfect matching or

$$|T(S)| \geq |S| \text{ for all } S \subseteq U, \tag{4}$$

where U is the set of unsaturated vertices of T . (The term Hall-perfect is chosen because of the well-known condition $|N(S)| \geq |S|$ for every $S \subseteq X$ for the existence of a matching saturating all vertices of X in a bipartite graph with bipartition X, Y . This condition is often referred to in literature as Hall’s condition).

With the help of the above concepts we are now ready to characterize all trees with parallel knock-out number at most 2.

Theorem 8. *A tree T is Hall-perfect if and only if $\text{PKO}(T) \leq 2$.*

Proof. If T is a Hall-perfect tree, we can give a KO-reduction scheme with one or two rounds. By (3), one round suffices if and only if M is a perfect matching. Suppose M is a matching satisfying (4). By Hall’s Theorem on matchings in bipartite graphs, this implies we can assign $p = |U|$ distinct pairwise M -disjoint U -triplets to the vertices of U , one v_i -triplet for each $v_i \in U = \{v_1, \dots, v_p\}$. We can use the following KO-reduction scheme for each v_i and its associated v_i -triplet given by the vertices u_i, w_i, x_i of the P_3 , where u_i is a neighbor of v_i . In the first round, for each i , v_i fires at one of the other neighbors (v_i is not a leaf), u_i fires at w_i , and w_i fires at x_i ; the matching edges of M that are not part of a v_i -triplet are used to eliminate all other saturated vertices (except for all u_i): they pairwise eliminate each other by firing along the matching edge. In the second round the edges $v_i u_i$ are used to eliminate all remaining vertices. Note that two triplets are allowed to intersect at their x_i -vertices.

For the converse, suppose T can be eliminated in at most two rounds. If T needs only one round, we are done since by (3) this implies T has a perfect matching. Now suppose T needs exactly two rounds. Then in the second round the edges of a matching N between the remaining vertices after the first round are used to mutually eliminate their incident vertices. Let us consider the edges $u_i v_i$ of this matching N . Each u_i has fired at a vertex $x_i \neq v_i$ of T in the first round; at x_i starts a (directed) path $Q_i = x_{i_1} \dots x_{i_{t_i}}$ of length at least one with the property that x_{i_s} fires at $x_{i_{s+1}}$ in the first round ($s = 1, \dots, t_i - 1$), and $x_{i_{t_i}}$ fires at $x_{i_{t_i-1}}$. Similar paths $R_i = y_{i_1} \dots y_{i_{r_i}}$ exist for the vertices v_i . Clearly, Q_i and R_i do not intersect (since T is a tree), and none of u_j, v_j is on Q_i or R_i (since u_j and v_j survive the first round). We denote by $P_{u_i v_i}$ the (undirected) path consisting of $Q_i, R_i, u_i v_i$ and the edges $u_i x_{i_1}$ and $v_i y_{i_1}$. We denote by M^* the matching that pairwise matches the vertices of T that are on none of the paths $P_{u_i v_i}$ and that mutually eliminate each other in the first round. Finally, we denote by $\bar{P}_{u_i v_i}$ the subpath of $P_{u_i v_i}$ that is vertex disjoint from the other paths $P_{u_j v_j}$.

Suppose that the two rounds in the KO-reduction scheme are chosen in such a way that $|N|$ is as small as possible. We call this number the second round size of T . We make a number of useful observations.

- (i) $\bar{P}_{u_i v_i}$ has even length.

Proof: Otherwise $\bar{P}_{u_i v_i}$ has a perfect matching, and all vertices on this path can be eliminated in the first round without affecting the elimination scheme for the other vertices, contradicting the choice of N .

- (ii) If Q_i (respectively R_i) has odd length, then $P_{u_i v_i}$ has a matching such that u_i (respectively v_i) is the only unsaturated vertex and $v_i y_{i_1} y_{i_2}$ (respectively $u_i x_{i_1} x_{i_2}$) is a u_i -triplet (respectively v_i -triplet) with respect to this matching.

Proof: This follows from (i) and the observation that R_i (respectively Q_i) has length at least 2.

If $|N| = 1$, then (i) and (ii) together yield that T is Hall-perfect. Now assume $|N| \geq 2$ and that all trees with $\text{PKO} = 2$ and a smaller second round size are Hall-perfect.

Let F denote the forest obtained from $T - V(M^*)$ by deleting all edges of N . Clearly, in F there is at most one path between $\{u_i, v_i\}$ and $\{u_j, v_j\}$.

- (iii) We can choose the labels $\{u_i, v_i\}$ in such a way that there are no paths in F between any u_i and v_j with $i \neq j$.

Proof: Let $F(u_1)$ denote the component of F containing u_1 . Relabel (if necessary) all v_i in $F(u_1)$ by u_i and the corresponding u_i by v_i . Now let $F(v_1)$ denote the component of F containing v_1 (with the new labels). Relabel all u_i in $F(v_1)$ by v_i ; do the same for all $F(v_j)$ such that $u_j \in V(F(u_1))$. Then continue with components $F(u_i)$ with $u_i \in V(F(v_j))$, etc. Repeating the procedure for different components of F it is not difficult to see that we can obtain a labeling with the desired property.

By (iii) and since T contains no cycles, there exists a pair $\{u_i, v_i\}$ with the property that Q_i or R_i neither intersects with any other Q_j nor R_j . Without loss of generality, let $\{u_1, v_1\}$ be a pair such that Q_1 does not intersect with any other Q_j or R_j . By (i), $\overline{P}_{u_1v_1}$ has even length. Since Q_1 and u_1v_1 are vertex disjoint parts of it, the length of $\overline{P}_{u_1v_1}$ is at least 4. Denote by v'_1 the end vertex of $\overline{P}_{u_1v_1}$ that is not an end vertex of Q_1 . Let $T' = T - V(\overline{P}_{u_1v_1})$. Clearly, the KO-reduction scheme for T restricted to the vertices of T' is a KO-reduction scheme for T' . Since the components of T' have a smaller second round size than T , by the induction hypothesis, all components of T' are Hall-perfect, with a corresponding matching M' . Let M_1 be a matching in $\overline{P}_{u_1v_1}$ that only leaves v'_1 unsaturated. Extend M' with the edges of M_1 . Then the new matching has the desired properties to show that T is Hall-perfect. \square

6. A dynamic program for trees

For paths it is particularly easy to determine their parallel knock-out numbers: one can easily check that $\text{PKO}(P_n) = 1$ if n is even, $\text{PKO}(P_n) = 2$ if $n \geq 7$ is odd, and that $\text{PKO}(P_n) = \infty$ for $n = 1, 3, 5$. For general trees, it is already a lot more complicated, but it is still tractable.

In this section, we describe a polynomial time algorithm for computing the parallel knock-out number of a tree T . By Lemma 3, $\text{PKO}(T)$ is either infinite, or it is bounded from above by $\lceil \log_3 n \rceil$, where n denotes the number of vertices in T . Without loss of generality we assume that $n \geq 3$.

We root the tree T in an arbitrary vertex called ROOT. We denote by $T(v)$ the maximal subtree of T that is rooted at vertex v . If $v \neq \text{ROOT}$, there is some edge e_v that connects v to its father f_v . We are interested in the behavior of KO-reduction schemes inside of the subtrees $T(v)$: for $v \neq \text{ROOT}$, the only interaction between $T(v)$ and $T - T(v)$ occurs along the edge e_v , and there is at most one round during which this edge e_v can be used. If e_v is used, then it is either fired upwards (the child v fires at the father f_v), or downwards (the father fires at the child), or both ways (father and child simultaneously fire at each other).

For every vertex $v \neq \text{ROOT}$ and for every $r = 1, \dots, \lceil \log_3 n \rceil$, we define three boolean predicates $\text{UP}[v; r]$, $\text{DOWN}[v; r]$, and $\text{BOTH}[v; r]$: the predicate $\text{UP}[v; r]$ (respectively $\text{DOWN}[v; r]$, respectively $\text{BOTH}[v; r]$) is true, if there exists a KO-reduction scheme for $T(v)$, in which in round r the edge e_v is fired upwards (respectively downwards, respectively both ways). Moreover, for every vertex v (including the root), we introduce a boolean predicate $\text{NONE}[v]$, which is true if there exists a KO-reduction scheme for $T(v)$ which does not interact with vertices outside of $T(v)$; for $v \neq \text{ROOT}$ this means that the edge e_v is not used at all.

We compute the values of all these predicates by working upwards through the tree, starting in the leaves and ending in the root. By similar arguments as in the proof of Lemma 7, for every leaf v_L , we have $\text{BOTH}[v_L; 1] = \text{true}$ and $\text{DOWN}[v_L; 1] = \text{UP}[v_L; 1] = \text{NONE}[v_L] = \text{false}$. Moreover, for all $r \geq 2$ the three predicates $\text{UP}[v_L; r]$, $\text{DOWN}[v_L; r]$, and $\text{BOTH}[v_L; r]$ are false. For non-leaf vertices v , the computation of the predicates is described in the following four lemmas.

Lemma 9. *For every non-leaf $v \in V(T)$ and for every $r = 1, \dots, \lceil \log_3 n \rceil$, the value of $\text{DOWN}[v; r]$ can be determined in polynomial time.*

Proof. Let v be a non-leaf vertex with children v_1, \dots, v_d and father f_v . What does it mean that $\text{DOWN}[v; r]$ is true? Since the father f_v fires in round r along the edge e_v , vertex v is eliminated in round r . In the first r rounds, vertex v must have fired at r of its children. In the first $r - 1$ rounds, none of the children has fired at vertex v . In round r , some of the surviving children of v may fire at v . In later rounds, none of the children can fire at v .

We model this situation via a bipartite auxiliary graph: the left vertex class in this bipartite graph has d vertices that correspond to the children v_1, \dots, v_d . The right vertex class of the bipartite graph has d vertices that correspond to the possible ways the d edges between vertex v and its children are used in r rounds of a KO-reduction scheme, as follows:

- For $k = 1, \dots, r - 1$ there is one edge that is used downwards during round k . We label a corresponding vertex in the bipartite graph by (DOWN, k) .
- There is one edge along which v fires in round r . We label a corresponding vertex in the bipartite graph by the two labels (DOWN, r) and (BOTH, r) .
- The remaining $d - r$ edges may be fired upwards in round r , or they are not being used at all. We label $d - r$ corresponding vertices in the bipartite graph by the two labels (UP, r) and (NONE) .

The edges in the bipartite graph are defined as follows:

- If a vertex x in the right class has a label (DOWN, k) (respectively (UP, k), respectively (BOTH, k)), and if DOWN[v_i ; k] = true (respectively if UP[v_i ; k] = true, respectively if BOTH[v_i ; k] = true), then the bipartite graph has an edge between x and the vertex corresponding to v_i in the left class.
- Analogously, if a vertex x in the right class has a label (NONE) and if NONE[v_i ; k] = true, then the bipartite graph has an edge between x and the vertex corresponding to v_i in the left class.

There are no other edges in the auxiliary graph. It can be seen that DOWN[v ; r] is true if and only if the auxiliary graph contains a perfect matching. The existence of a perfect matching can be decided in polynomial time by standard methods. \square

Lemma 10. *For every non-leaf $v \in V(T)$ and for every $r = 1, \dots, \lceil \log_3 n \rceil$, the value of UP[v ; r] can be determined in polynomial time.*

Proof. Let v be a non-leaf vertex with children v_1, \dots, v_d and father f_v . If UP[v ; r] is true, then vertex v fires in round r upwards along the edge e_v . Therefore, vertex v must stay alive until it is eliminated in some round $s \geq r$. In the rounds $1, 2, \dots, r - 1$ and $r + 1, \dots, s$, vertex v must have fired at its children. In the first $s - 1$ rounds, none of the children has fired at vertex v . In round s , some of the surviving children of v may fire at v .

Hence, if we are given the value of s , then we can model this situation as a bipartite matching problem pretty much the same way as we did in the proof of Lemma 9. To find the value of UP[v ; r], we simply test all possible values for $s = r, r + 1, \dots, \lceil \log_3 n \rceil$. UP[v ; r] is true if and only if at least one of these bipartite auxiliary graphs has a perfect matching. \square

Lemma 11. *For every non-leaf $v \in V(T)$ and for every $r = 1, \dots, \lceil \log_3 n \rceil$, the value of BOTH[v ; r] can be determined in polynomial time.*

Proof. Once again, let v be a non-leaf vertex with children v_1, \dots, v_d and father f_v . If BOTH[v ; r] is true, then vertex v and its father f_v eliminate each other in round r . In the rounds $1, 2, \dots, r - 1$ vertex v must have fired at its children, whereas none of the children has fired back at v . In round r , vertex v does not fire at its children, whereas some of the surviving children of v may fire at v . This problem can be modeled and solved as a bipartite matching problem too. \square

Lemma 12. *For every non-leaf $v \in V(T)$, the value of NONE[v] can be determined in polynomial time.*

Proof. If NONE[v] is true, then vertex v is eliminated by one of its children in some round s with $1 \leq s \leq \lceil \log_3 n \rceil$. In the first $s - 1$ rounds, vertex v must have fired at its children, whereas none of the children has fired back at v . In round s , vertex v fires at a child, and some of the surviving children of v may fire at v . We test all possible values for s , and solve the corresponding bipartite matching problems. \square

If NONE[ROOT] is true in the end, then T is KO-reducible. To find the exact value of PKO(T), we remember the smallest number s in the proof of Lemma 12 for which a perfect matching exists. A perfect matching in a bipartite graph with α vertices and β edges can be found in $O(\beta\sqrt{\alpha})$ time (see, e.g., [7, p. 15]). Our algorithm faces matching problems with $O(n)$ vertices and $O(n^2)$ edges, and altogether there are $O(n \log^2 n)$ matching problems to be solved. This yields the following theorem.

Theorem 13. *The parallel knock-out number of an n -vertex tree T can be computed in $O(n^{3.5} \log^2 n)$ time.*

We note that, in contrast to the situation in many optimization problem areas, the dynamic program for trees that is presented here does not carry over to graphs of bounded tree-width, since the intrinsic polynomial problem of obtaining a perfect matching in an auxiliary bipartite graph has no polynomial time analogue if we move from trees to graphs with tree-width at least 2.

7. Claw-free graphs

We now turn to *claw-free* graphs, i.e., graphs that contain no $K_{1,3}$ as an induced subgraph. This is a well-studied class of graphs, especially with respect to algorithmic and structural properties. We refer to [3] for an excellent survey paper on claw-free graphs.

Claw-free graphs admit perfect matchings and 2-factors under rather mild conditions. We give two examples of known results. Sumner [9] and Las Vergnas [6] independently proved that every connected claw-free graph on an even number of vertices has a perfect matching. Ryjáček [8] proved that a connected claw-free graph on an odd number $n \geq 3$ of vertices has a spanning subgraph consisting of disjoint matching edges and odd cycles whenever at most one vertex has degree one.

It is natural to consider conditions that guarantee a low parallel knock-out number in a claw-free graph. Clearly, the above results together imply the following result involving the minimum degree $\delta(G)$ of the vertices of G .

Theorem 14. *Let G be a claw-free graph with $\delta(G) \geq 2$. Then $\text{PKO}(G) \leq 2$.*

It is easy to give examples showing that we cannot omit the degree condition in the above result. One could try to replace the minimum degree condition by the weaker condition that every vertex with degree 1 has a neighbor with a high degree, but this does not work either. Consider e.g., the claw-free graph G obtained from a complete graph K_k ($k \geq 2$) by adding $k - 1$ new vertices and $k - 1$ matching edges saturating all new vertices. One easily checks that $\text{PKO}(G) = \infty$.

We next point out and prove a perhaps more surprising result. Unlike the situation with trees, as was the case with paths, any claw-free graph has the property that either its parallel knock-out number is infinite or it is at most 2.

Theorem 15. *Let G be a KO-reducible claw-free graph. Then $\text{PKO}(G) \leq 2$.*

To prove the above theorem, we will show that we can find a suitable KO-reduction scheme for a KO-reducible claw-free graph, namely one in which after the first round a (claw-free) graph is left whose vertex set can be partitioned into sets inducing P_2 s and C_3 s. In order to show this, we need the following useful observation.

Lemma 16. *The vertex set of a claw-free graph without isolated vertices can be partitioned into sets inducing P_2 s, P_3 s and C_3 s.*

Proof. It clearly suffices to prove the lemma for connected graphs. Let G be a connected claw-free graph and let P be a partition of $V(G)$ into sets containing 1, 2 or 3 vertices, chosen in such a way that the subgraphs induced by the sets of P contain as few isolated vertices as possible. If there are no isolated vertices at all, we are done, so suppose v is an isolated vertex. Then, by the choice of P , all neighbors of v are vertices of degree 2 in the induced subgraphs on 3 vertices, and also by the choice of P , these subgraphs are P_3 s (if there was a C_3 among them, we could define two sets on 2 vertices inducing P_2 s instead of v and the C_3). We conclude that v and a set of 3 vertices in P containing a neighbor of v together induce a $K_{1,3}$ in G , a contradiction. \square

Proof of Theorem 15. Let G be a claw-free graph with $\text{PKO}(G) < \infty$. If $\text{PKO}(G) = 1$, we are done, so we may assume $\text{PKO}(G) \geq 2$. Let H denote the subgraph of G induced by the vertices that remain after the first round of a KO-reduction scheme for G . Then H is claw-free, hence contains a partition P as in the above lemma. We assume that the (first round of the) KO-reduction scheme has been chosen in such a way that among the induced subgraphs induced by the sets of P there are as few as possible P_3 s. We complete the proof by showing that there are in fact no P_3 s among them, hence all vertices of H can be eliminated in the second round. This last step in the proof is by contradiction.

Assume one of the subgraphs induced by a 3-set $\{u, v, w\}$ of P is a P_3 , with edges uv and vw , hence $uw \notin E(G)$. Assume v has fired at a vertex y in the first round. Since G is claw-free, we may assume $uy \in E(G)$. Now redefine the first round by letting v fire at w instead of y . If y survives the first round, we obtain a C_3 instead of P_3 , otherwise a K_2 . \square

8. Discussion

We derived a variety of combinatorial and algorithmic results on the parallel knock-out number of bipartite graphs, trees, and claw-free graphs. We answered a number of questions, but there remain many open problems on parallel knock-out numbers. We conclude the paper by listing some of these questions. The most important question is probably to settle the following conjecture:

The square-root conjecture for parallel knock-out schemes. Every n -vertex graph G satisfies either $\text{PKO}(G) = \infty$ (if it is not KO-reducible) or $\text{PKO}(G) < 2\sqrt{n}$ (if it is KO-reducible).

It has been announced by Lampert and Slater [4] that computing the parallel knock-out number is NP-hard. However, the special case of deciding whether $\text{PKO}(G) = 1$ is straightforward. The complexity of deciding whether $\text{PKO}(G) \leq k$ (or $\text{PKO}(G) = k$) for some fixed value $k \geq 2$ that is not part of the input has been studied in [1]. Both problems turn out to be NP-complete for any fixed $k \geq 2$.

We gave an $O(n^{3.5} \log^2 n)$ algorithm for computing the parallel knock-out number of an n -vertex tree. Is there a substantially faster algorithm for this problem, with a time complexity of, say, $O(n \log n)$ or $O(n^2)$? Can we avoid the computation of perfect matchings in auxiliary bipartite graphs while computing $\text{PKO}(T)$ for a tree T ? Can we then generalize such a method to graphs of bounded tree-width? In [1], it has been shown that both aforementioned decision problems can be formulated in monadic second order logic, implying that there is a polynomial algorithm to solve them for graphs with bounded tree-width.

We showed that the parallel knock-out number of a claw-free graph is either infinite or bounded by 2. Does the following generalization of this result hold: the parallel knock-out number of a $K_{1,k}$ -free graph is either infinite or bounded by $k - 1$?

Acknowledgments

We thank the two anonymous referees for their useful comments and suggestions, and we thank Matthew Johnson for pointing out an error in an earlier version of our proof of Theorem 8.

References

- [1] H.J. Broersma, M. Johnson, D. Paulusma, I.A. Stewart, The computational complexity of the parallel knock-out problem, Lecture Notes in Computer Science 3887 (2006) 250–261.
- [2] G. Chartrand, L. Lesniak, Graphs and Digraphs, Chapman & Hall, London, 1996.
- [3] R. Faudree, E. Flandrin, Z. Ryjáček, Claw-free graphs—a survey, Discrete Math. 164 (1997) 87–147.
- [4] D.E. Lampert, P.J. Slater, Parallel knockouts in the complete graph, Amer. Math. Monthly 105 (1998) 556–558.
- [6] M. Las Vergnas, A note on matchings in graphs, Cahiers Centre Etudes Rech. Opér. 17 (1975) 257–260.
- [7] L. Lovász, M.D. Plummer, Matching theory, Ann. Discrete Math. 29 (1986).
- [8] Z. Ryjáček, Factors and circuits in $K_{1,3}$ -free graphs, in: Combinatorics and Graph Theory, vol. 25, Polish Scientific Publishers, Banach Centre Publications, Warsaw, 1989, pp. 137–145.
- [9] D.P. Sumner, Graphs with 1-factors, Proc. Amer. Math. Soc. 42 (1974) 8–12.