On a Game in Directed Graphs

Alan J. Hoffman* Kate Jenkins† Tim Roughgarden‡

Abstract

Inspired by recent algorithms for electing a leader in a distributed system, we study the following game in a directed graph: each vertex selects one of its outgoing arcs (if any) and eliminates the other endpoint of this arc; the remaining vertices play on until no arcs remain. We call a directed graph *lethal* if the game must end with all vertices eliminated and *mortal* if it is possible that the game ends with all vertices eliminated. We show that lethal graphs are precisely collections of vertex-disjoint cycles, and that the problem of deciding whether or not a given directed graph is mortal is NP-complete (and hence it is likely that no "nice" characterization of mortal graphs exists).

Keywords: Theory of computation; Graph algorithms; Distributed computing

1 Introduction

A fundamental and well-studied problem in distributed computation is that of electing a unique *leader* from the nodes of a network. Several recent algorithms for this problem (in both synchronous and asynchronous models) proceed by successively eliminating vertices (i.e., designating them ineligible for leadership) until only one vertex (the elected leader) remains [1, 2, 5]. Inspired by this algorithmic theme of "successive elimination", we define and study the following simple game in a directed network G with vertex set $V(G)$ and arc set $E(G)$. Roughly, the game consists of a sequence of "rounds", where in one round each vertex v selects one of its outgoing arcs (if there

*IBM Watson Research Center. Email: `hoffa@watson.ibm.com`.

†Akamai Technologies, Cambridge, MA. Research performed while at IBM Watson Research Center and the Center for Applied Mathematics, Cornell University, Ithaca, NY 14853. Email: `kjenkins@akamai.com`.

‡Department of Computer Science, Cornell University, Ithaca, NY 14853. Part of this work done while visiting IBM Watson Research Center. Supported in part by an NSF Graduate Fellowship, a Cornell University Fellowship, and ONR grant N00014-98-1-0589. Email: `timr@cs.cornell.edu`.

are any), say (v, w), and eliminates the vertex w. In the following round, the game continues in the directed graph induced by the remaining vertices; the game ends when no arcs remain. While recent leader election algorithms suggested this game to us, we do not claim that it models the leader election problem.

Formally, we define a *round* R in G to be a subset of $E(G)$ such that if v is a vertex with positive out-degree in G, there is precisely one arc in R with tail v. A vertex that is the head of no arc in R is called a *survivor* of R, and we denote the set of all survivors of a round R by $s(R)$. By a *game* in G, we mean a sequence $(G_1, R_1, G_2, R_2, \ldots, G_k)$ where:

- $G_1 = G$
- for $i = 1, \ldots, k-1$, R_i is a round in G_i
- for $i = 1, \ldots, k-1$, G_{i+1} is the subgraph of G_i induced by $s(R_i)$
- G_k has an empty arc set (and possibly an empty vertex set).

Games of this sort have previously been investigated in complete directed graphs, under the assumption that in each round a vertex selects an outgoing arc uniformly at random. In particular, the probability that some vertex survives a random game has been characterized (as a function of the number of vertices) [3, 6].

We call a game *lethal* if its final graph is empty. We call a directed graph G *lethal* if every game in G is lethal, and *mortal* if some game in G is lethal. Put differently, a random game in G is lethal with probability 1 if and only G is lethal, while it is lethal with probability 0 if and only if G is not mortal. We are interested in characterizing the sets of lethal and mortal directed graphs. In Section 2 we prove that a graph is lethal if and only if it is a collection of vertex-disjoint cycles. In Section 3 we give evidence that no similarly simple characterization of mortal graphs exists, by showing that the problem of deciding whether or not a given directed graph is mortal is NP-complete.

2 Lethal Graphs

We begin with an auxiliary definition and an easy lemma. Call a directed graph G *rapidly lethal* if G is lethal and in addition every game in G contains only one round (that is, all vertices must be eliminated after only one round of play).

Lemma 1 *A graph is rapidly lethal if and only if it is a collection of vertex-disjoint cycles.*

Proof: One direction is obvious. For the other, it suffices to show that each vertex in a rapidly lethal graph G has in-degree and out-degree 1. Since any game in G has the form $(G, R, \emptyset)$, any round R must contain n arcs with distinct heads and distinct tails (we must have $s(R) = \emptyset$), which immediately implies that each vertex must have in-degree and out-degree at least 1. Further, if any vertex v had in-degree 2 or more, we could define a round with at least one survivor (select two arcs with head v), a contradiction. Finally, a simple handshaking argument shows that every vertex must have out-degree 1. ■

We next give a proof of our first result, a characterization of lethal graphs.

Theorem 2 *A graph is lethal if and only if it is a collection of vertex-disjoint cycles.*

Proof: By the lemma, it suffices to show that any lethal graph is rapidly lethal. Suppose not, and consider a counterexample G having the fewest possible number of vertices and a (lethal) game in G with more than one round, say $(G_1, R_1, \dots, G_k, R_k, \emptyset)$ with $k \geq 2$. Let H denote the subgraph of G induced by $V(G_k)$ (the "last survivors") and let K denote the subgraph induced by $V(G) \backslash V(G_k)$. Since G is lethal, H must be lethal and hence by minimality of G and Lemma 1 is a collection of vertex-disjoint cycles. Let T denote the only possible round in H.

We next claim that K must be lethal. Assume the contrary, and let $(K_1, S_1, K_2, S_2, \dots, K_r)$ be a non-lethal game in K (thus K_r has a non-empty vertex set). For each vertex $v \in V(K)$ with out-degree 0 in K and strictly positive out-degree in G, select an arbitrary outgoing arc (v, w) (with w necessarily in $V(H)$). Let S' denote the (possibly empty) collection of these arcs. Then, we may extend the non-lethal game in K to a non-lethal game in G (intuitively by running games in H and K in parallel), namely $(G, S_1 \cup S' \cup T, K_2, S_2, \dots, K_r)$; this contradicts the assumption that G is lethal, proving K lethal.

Appealing once more to the minimality of G and Lemma 1, K must be a collection of vertex-disjoint cycles (call its only possible round S). Thus, G is a collection of vertex-disjoint cycles together with some arcs between H and K. However, if (v, w) is an arc in G with one endpoint in each of $V(H), V(K)$, then taking e to be the unique arc in $S \cup T$ with tail v, $(S \cup T \cup \{(v, w)\}) \backslash \{e\}$ is a round in G with precisely one survivor (namely, the head of e), contradicting the assumption that G is lethal. Thus G is the union of vertex-disjoint cycles and fails to be a counterexample. ■

We note that Theorem 2 gives a trivial linear-time algorithm for deciding whether or not a given graph is lethal. In addition, the proof of Theorem 2 is easily modified to give a polynomial-time algorithm for constructing a non-lethal game in a non-lethal graph.

3 Mortal Graphs

In this section we direct our attention toward the set of mortal graphs—graphs in which some game is lethal. Our main result gives strong evidence that no simple (more precisely, polynomial-time verifiable) characterization of mortal graphs exists (cf. Theorem 2).

Theorem 3 *The problem of deciding whether or not a given directed graph is mortal is NP-complete.*

Proof: The problem is clearly in NP (a lethal game can be checked in polynomial time). To show hardness, we reduce the problem of checking whether or not a given CNF formula ϕ is satisfiable to that of checking whether or not a given graph is mortal (see [4] for the relevant background). Suppose a given CNF formula ϕ has n clauses $C_1, \ldots, C_n$ and contains m Boolean variables $x_1, \ldots, x_m$. We may assume without loss of generality that no clause contains both a variable and its negation. Define a graph G as follows:

- *clause vertices:* for each clause C_j introduce $m+1$ vertices $u_{j,1}, \ldots, u_{j,m+1}$
- *variable vertices:* for each variable x_i introduce three vertices $v_{i,T}, v_{i,F}, w_i$
- *cleanup vertices:* introduce $2n(m+1)+m-2$ additional vertices
- introduce the arc $(v_{i,\alpha}, u_{j,k})$ (where $\alpha \in \{T, F\}$) if and only if giving variable x_i the truth assignment α satisfies clause C_j (independent of k)
- for every $i = 1, \ldots, m$, include the arcs $(w_i, v_{i,T})$ and $(w_i, v_{i,F})$
- introduce arcs from each cleanup vertex to each non-clause vertex (including to other cleanup vertices).

It is clear that the size of G is polynomial in the size of ϕ. As an example, a sketch of the graph arising from the formula $x_1 \vee x_2$ is shown in Figure 1.

We claim that G is mortal if and only if ϕ is satisfiable. First suppose ϕ is satisfiable, and that assigning variable x_i the truth value $\alpha_i \in \{T, F\}$ (for each i) satisfies ϕ; we will exhibit a lethal game in G consisting of $n(m+1)+2$ rounds. Construct the first round R_1 as follows:

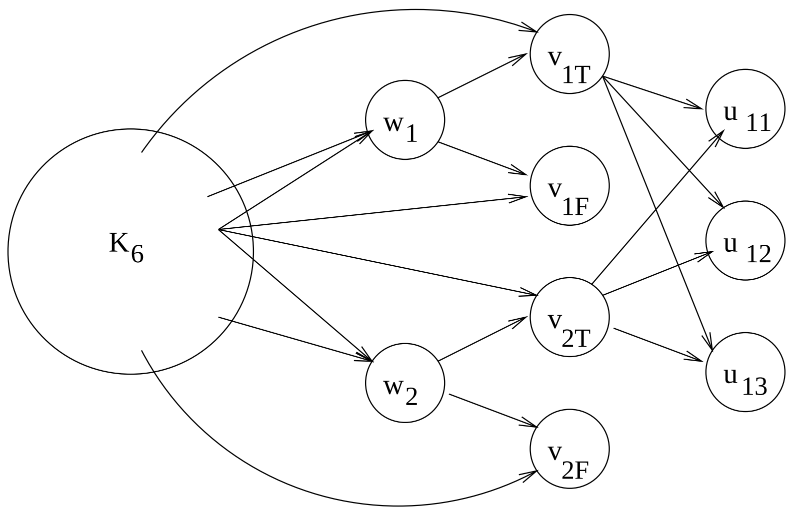

Figure 1: The mortal graph arising from $\phi = x_1 \vee x_2$ (not all arcs are shown).

- for each cleanup vertex, include an outgoing arc whose head is among $w_1, \ldots, w_m$; moreover, do this so that all of $w_1, \ldots, w_m$ will be eliminated in the first round
- for each i, include arc $(w_i, v_{i,\neg\alpha_i})$
- include an arbitrary outgoing arc for every other variable vertex.

Following round R_1, we will be left with a graph containing all of the cleanup vertices, variable vertices corresponding precisely to the satisfying assignment of ϕ, and some clause vertices. Construct each of the next $n(m+1)-1$ rounds as follows: for each variable vertex select an arbitrary outgoing arc (if there is one) and pick two cleanup vertices that will be precisely the vertices eliminated by all other selected arcs emanating from cleanup vertices (this is possible since we always have a complete digraph among the remaining cleanup vertices). Thus, following the first $n(m+1)$ rounds, there will no longer be any clause vertices: on one hand, every clause vertex can be eliminated by some remaining variable vertex since every clause is satisfied by the truth assignment; on the other hand, each remaining variable vertex has out-degree at most $n(m+1)$ and cannot eliminate any vertex more than once. The remaining graph thus consists of m variable vertices (now with out-degree 0) and m cleanup vertices (recall we began with $2n(m+1)+m-2$ and eliminated 2 in each of the rounds $2, 3, \ldots, n(m+1)$). To complete the lethal game, in the penultimate round select a matching from the cleanup vertices to the variable vertices (leaving only the clique of the final m cleanup vertices) and in the final round select, for example, a Hamiltonian cycle.

Finally, we show that if G is mortal then ϕ is satisfiable. The first key observation is that for any game in G and for each variable i, at least one of $v_{i,T}, v_{i,F}$ is eliminated in the first round (each w_i must select one of its two outgoing arcs in the first round). The second is that for any clause C_j, at most m of the $m+1$ clause vertices $u_{j,1}, \ldots, u_{j,m+1}$ will be eliminated in the first round (recall we assume that no clause of ϕ contains both a variable and its negation).

Now fix a lethal game in G; in particular all $m+1$ clause vertices belonging to a clause C_j are eventually eliminated and thus some variable vertex of the form v_{i,α_i} selects an outgoing arc with head of the form $u_{j,k}$ in some round subsequent to the first. It is then easy to see that assigning variable x_i the truth value $\alpha_i \in \{T, F\}$ whenever the variable vertex v_{i,α_i} survives the first round yields a well-defined partial truth assignment that can be extended to a satisfying assignment of ϕ in an arbitrary manner (i.e., if both $v_{i,T}$ and $v_{i,F}$ are eliminated in the first round, then x_i may be given an arbitrary truth value). ■

4 Acknowledgements

We thank Don Coppersmith and Greg Sorkin for introducing us to the described game in directed graphs.

References

[1] Y. Afek and E. Gafni. Time and message bounds for election in synchronous and asynchronous complete networks. *SIAM Journal of Computing*, 20(2):376–394, 1991.

[2] Y. Afek and E. Gafni. Distributed algorithms for unidirectional networks. *SIAM Journal of Computing*, 23(6):1152–1178, 1994.

[3] D. Coppersmith. Personal communication.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[5] G. Singh. Leader election in complete networks. *SIAM Journal of Computing*, 26(3):772–785, 1997.

[6] P. Winkler. Personal communication.