Deciding Soccer Scores and Partial Orientations of Graphs

Dömötör Pálvölgyi*

Abstract

We show that deciding if a simple graph has a partial orientation of its edges such that all vertices have a prescribed in-, out- and undirected degree, is NP-complete. We prove a related question, that if we know that in a soccer-tournament who played who so far, but we do not know the outcomes, then deciding whether a score vector is legal or not, is NP-complete.

1 Introduction

The problem of deciding whether we can direct a graph with each vertex having a prescribed in- and out-degree is wellknown to be in P. It is another interesting question to determine the complexity of the problem where instead of a directed graph, we want to obtain a mixed graph, ie. a graph that can have both directed and undirected edges, and we prescribe the in-, out- and undirected-degree of each vertex. Let us denote the problem of deciding whether this can be done or not by Partial Orientation Problem. We show that Partial Orientation Problem is NP-complete.

The Elimination Problem is to decide whether a given team can still win the tournament at some point. This was shown to be NP-complete not so long ago independently by Bernholt et al. ([1]) and Kern and Paulusma ([2]). Later it was also generalized to various other point-systems by Kern and Paulusma ([3]), in this paper they solve completely for which score allocation rules the problem is NP-complete, assuming that we do not require that the score vector is reachable in a valid tournament. They suspect that deciding

*Dept of Comp. Sci. and Communication Networks Lab, ELTE, Budapest.
Supported by OTKA T67867.

if a score vector is reachable or not (if we know the remaining games) is a difficult problem. So let us denote the problem of deciding whether a given score vector is a possible result of a soccer-tournament or not (if we know which team played against which so far) by SCORE VECTOR PROBLEM . In this paper we prove that the SCORE VECTOR PROBLEM is NP-complete (in the case when teams get $1 < p \neq 2$ points for winning, 1 for drawing and 0 for losing a game). The proof is an easy consequence of our construction given to the PARTIAL ORIENTATION PROBLEM.

We denote the degree of a vertex v in a simple graph by $d(v)$. In the mixed graph the in-degree is denoted by $\rho(v)$, the out-degree by $\delta(v)$ and the number of the adjacent undirected edges by $\theta(v)$. Thus $d(v) = \rho(v) + \delta(v) + \theta(v)$. When we say orientation, we mean *three* possibilities: The two directions and the undirected case. Thus in the beginning we have a graph with unoriented edges and we want to orient them.

2 Partial Orientation Problem is NP-complete

We reduce 3-SAT to the PARTIAL ORIENTATION PROBLEM as follows: We construct a graph for each input formula to 3-SAT. For each x_i variable the graph will have a tree that is almost binary; its root has degree two, each vertex on an odd level has degree three and each vertex on an even level has degree two. The last level is an even one, and from each leaf there is an edge connecting the tree to the rest of the graph. (See Figure 1.) For the root we prescribe $\rho(r_i) = \delta(r_i) = 1$. For the orientation of each edge of the tree there will be exactly two possibilities. The direction of the two edges of r_i will determine the orientation of each other edge in the tree.

For each vertex w on an odd level of the tree we prescribe $\rho(w) = \delta(w) = \theta(w) = 1$ and for each vertex v on an even level we prescribe either $\rho(v) = \delta(v) = 1$ or $\rho(v) = \theta(v) = 1$ or $\theta(v) = \delta(v) = 1$. When we say that v is $\rho\delta$ (or $\rho\theta$ or $\delta\theta$), we mean that for the degree two vertex v the prescription is $\rho(v) = \delta(v) = 1$. One of the two grandchildren of a $\rho\delta$ vertex is always a $\rho\theta$, while the other is always a $\delta\theta$. Similarly, the $\rho\theta$ vertices have $\rho\delta$ and $\delta\theta$ grandchildren and $\delta\theta$ vertices have $\rho\delta$ and $\rho\theta$ grandchildren. The root, which has four grandchildren, has two $\rho\theta$ and two $\delta\theta$ grandchildren. This finishes the description of the tree. Note that since every edge in the tree is incident to a vertex of degree two, we have exactly two possible orientation for each edge. When we say that an edge is $\rho\delta$, we mean that its orientation cannot be undirected.

Eg., let us take one of r_i's children, w, and both of w's children, v_1 and v_2. The edge r_iw can be either directed towards w or away from w but it cannot

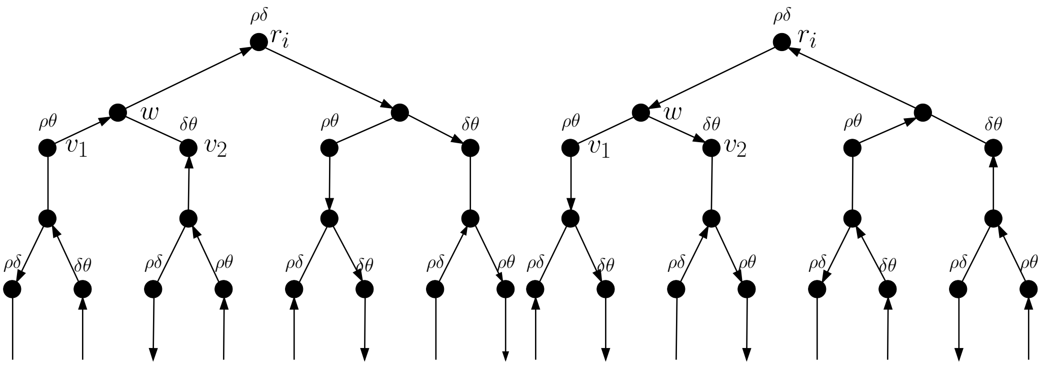

Figure 1: The two possible orientations of the tree associated with x_i.

be undirected. (See the two possibilities in Figure 1.) The edge connecting v_1 to its child can be undirected or directed away from v_1 but this is determined by the orientation of r_iw. The edge connecting v_2 to its child can be directed towards v_2 or be undirected and this is also determined by the orientation of r_iw. These edges determine the orientation of the edges under them and therefore the orientation of the whole tree depends on the choice at the root. This way we can achieve that from one decision at r_i we have an arbitrary number of edges directed to the same way from the leaves of the tree. Let us count how many.

Let us denote the number of the $\rho\delta$ edges (the ones that cannot be undirected) that are going from the $2l$th level to the $2l+1$th by $a(l)$ and the number of the other edges at the same level by $b(l)$. We have $a(0) = 2$ and $b(0) = 0$ and it is easy to see that the equations $a(l) = b(l-1)$ and $b(l) = 2a(l-1)+b(l-1)$ hold. Solving these we get $a(l+1) = b(l) = 4(2^l-(-1)^l)/3$. For each variable x_i, we need the tree associated with x_i to have $a(l) \geq$ twice the appearances of x_i (or $\overline{x_i}$) in the clauses, so the size of the tree is only polynomial. Note that half of the edges counted in $a(l)$ are directed towards the tree, and the other half away from the tree, whichever orientations we choose at r_i. We will call one of these orientations *true* and the other orientation *false*. For each clause that contains x_i we reserve an edge that is directed away from the tree in the *true* orientation and towards the tree in the *false* orientation. Similarly, for each clause that contains $\overline{x_i}$ we reserve an edge that is directed towards the tree in the *true* orientation and away from the tree in the *false* orientation. This can be done since $a(l)$ is sufficiently large.

For each clause C the graph will have a vertex v_c of degree 5. The prescription for each v_C is $\rho(v_C) = 3$ and $\delta(v_C) = 2$. The three edges reserved for clause C (adjacent to the leaves of the trees associated with the variables of C) are connected to the vertex v_C. The remaining two edges are connected to the degree two $\rho\delta$ vertices v_{C1} and v_{C2}.

Now we are done with the representation of our formula, we only need to somehow get rid of the edges that have only one incident vertex so far. To this end, we add the *mirrored reflection* of everything constructed so far to the graph. This means for every vertex v that belongs to a tree or a clause, we add a v' vertex that is connected to w' if and only if v is connected to w. We also connect v and v' if and only if v has an edge that was not connected to any other vertex yet. The prescription of v' is $\rho(v') = \delta(v), \delta(v') = \rho(v)$ and $\theta(v') = \theta(v)$. This finishes our construction.

Now we have to prove that this graph has a mixed orientation fulfilling the required prescriptions if and only if the original formula had a true assignment.

First, if the formula had a true assignment, then let us orient the edges of the trees associated with the true variables in their *true* orientation and orient edges of the trees associated with the false variables in their *false* orientation. Each v_C will have at least one edge entering from a tree, we can pick the two edges connecting it to v_{C1} and v_{C2} such that $\rho(v_C) = 3$. We do the opposite with each edge in the mirrored part of the graph, this guarantees a good orientation for the vv' type edges.

Similarly, if the graph has a good orientation, then let us pick the variables associated with the trees whose orientation is *true* to be true, and the rest to be false. Since $\rho(v_C) = 3$ and only two edges can enter v_C that are not coming from a tree, therefore one of the trees associated with a variable of C must have *true* orientation, thus each clause must have a true literal.

3 The Score Vector Problem

To prove that the Score Vector Problem is NP-complete, we associate a vertex of a graph to each of the teams. The graph is the same as in the previous construction, but instead of prescribing the degrees of a vertex v, we prescribe the score of the team associated with that vertex to be $p\delta(v) + \theta(v)$ (it would get this much if it had won $\delta(v)$, drew $\theta(v)$ and lost $\rho(v)$ games). Now we only have to notice that in our construction the score of each vertex that has degree at most three, determines the number of games that the team associated with that vertex won, drew and lost. Eg., if a vertex w has $p+1$ points and $d(w) = 3$, this is only possible if it has won one game, drew one game and lost one game (since $1 < p \neq 2$). Since none of the vertices adjacent to the v_C's drew any of their games, the v_C's must have 3 wins and 2 losses. Therefore our construction reduces 3-SAT to SCORE VECTOR PROBLEM if instead of the degrees we prescribe the scores.

Note that when $p = 2$, the construction fails because one win, one draw and one losing worth the same number of points as three draws. For this $p = 2$ case the problem is in P and the proof is a folklore; just take the original simple graph, double every edge and ask whether this graph can be (completely) directed such that for every vertex v the prescription is $\delta(v) =$ the score of v.

In a soccer tournament usually the teams have played the same number of matches at a given time, while in our construction the degrees vary. We can fix this by adding a few new vertices who have won all their matches and played some of the teams whose degree is less than the average. Also, in tournaments everyone plays with everyone else in a round, so at any point the who-played-who-so-far graph can be partitioned into perfect matchings. Our construction with a little modification can be transformed into a regular bipartite graph that has this property.

4 Acknowledgments and Concluding Remarks

I would like to thank my supervisor, Zoltán Király for early discussions on the subject. I would also like to thank Attila Bernáth for his useful advices. He also noticed that if instead of the 3-SAT problem we use the ONE-IN-THREE-SAT problem (meaning that in a 3-CNF we want exactly one literal to be true, also NP-complete), then we do not need the v_{Ci} vertices and thus we obtain a graph with maximum degree three, which is clearly optimal.

An interesting open question remains to determine the complexity of the problem when we only know the score (or the in-, out- and undirected degrees) of each vertex and the number of games it played and we have to decide whether it is a possible outcome of a real tournament or not. We conjecture these problems to be in P although we could not even solve it in the case when we know that everyone played with everyone exactly once (meaning the tournament is finished, ie. the graph is the complete graph). A similar question can be raised concerning the ELIMINATION PROBLEM.

References

[1] T. Bernholt, A. Gülich, T. Hofmeister, N. Schmitt, Football elimination is hard to decide under the 3-point-rule, in: Mathematical Foundations

of Computer Science 1999, Lecture Notes in Computer Science, Vol. 1672, Springer, Berlin, 1999, 410-418.

[2] W. Kern, D. Paulusma, The new FIFA rules are hard: complexity aspects of sports competitions, Discrete Applied Mathematics 108 (2001), 317-323.

[3] W. Kern and D. Paulusma, The computational complexity of the elimination problem in generalized sports competitions, Discrete Optimization 1 (2004), 205-214.