Spanning trees with a given number of leaves

Thinh D. Nguyen
Moscow State University
kosmofarmer@gmail.com

Abstract. *We prove the hardness of yet another problem in graph theory with a flavor of computer games, namely* Terra Mystica. *Much like works of Demaine and others, we abstract the computer game to have a mathmematical model which is in the form of combinatorial optimization. A reduction from* 3Sat *shows that our claim holds.*

Keywords: *edge, graph, leaf, MST, spanning, tree, vertex*

1. Introduction

In recent decades, many computer games are generalized as mathematical games. Using tools from game theory and graph theory, most of them are proven to be hard, in the sense that one cannot hope to efficiently find a winning strategy, or construct the required constructions. In this section, we give the description of the game and necessary notions from graph theory.

1.1. Graph theory notions

A graph $G = (V, E)$ is an undirected graph with the *vertex set* $V = V(G)$ and *edge set* $E = E(G)$. In a graph, every edge (u, v) $(u, v \in V)$ is undirected. A *tree* is a connected graph without any cycle. And, a cycle in a graph G is a sequence of vertices $(v_1, v_2, \ldots, v_k)$ in which, for all $1 \leq i < k$, (v_i, v_{i+1}) is an edge of G, and also (v_k, v_1) is an edge of G. Given a graph G, a *spanning tree* T of it is one of its subgraphs which happens to be a tree and contains every vertex of G. Of course, G can have many spanning trees. The neighborhood of a vertex u is the set of its *adjacent* vertices, where two vertices u, v are adjacent if (u, v) is an edge. The cardinality of a vertex's neighborhood is its *degree*. In a tree, a vertex is called a *leaf*, if it has only one neighbor.

A well-known result states that every n-vertex tree has exactly $n - 1$ edges. Thus, the number of vertices in a tree determines the number of its edges. But, the number of its leaves can vary vastly from 2 to $n - 1$ (consider a *path* and a *star*, respectively). Now, we can have the first definition in this article about an optimization problem on graphs

Definition 1.1. k-leaf MST = { $<G, k>$| G has an MST T with k leaves}

1.2. Terra Mystica

Now, we are going to give the definitions that are pertinent to our soon-to-be-defined problem. When playing *Terra Mystica*, it might be useful to predict how many spades you will get throughout the game, and use this information to decide where to build, such that you stand a good chance of having the longest network. Let us abstract that into the following problem:

- Let a graph G be given, $G = (V,\ E)$
- Associate a terraforming cost with each vertex, $C : V \to \{0,\ 1,\ 2,\ 3\}$
- You are given some number of spades s

Your goal is to find a set of vertices V^* such that:

- You can afford to build on all of them, $\sum_{v \in V^*} C(v) \leq s$
- They are connected: the graph $(V^*, V^* \times V^* \cap E)$ is connected
- It is the largest such set: for all V' which satisfies the two previous conditions, $|V^*| \geq |V'|$

Definition 1.2. Above, we give the optimization version of the abstract mathematical game. The decision version takes a threshold parameter k and asks "Is there a vertex set T with size $|T| \geq k$ and which satisfies the first two criteria?" We denote this decision problem by Terra Mystica.

In the following sections, we describe two approaches to prove the hardness of Terra Mystica.The first is to reduce from 3Sat to k-leaf MST and then to Terra Mystica. The second one also uses MST but combine with some more idea such as paths on an MST.

2. From 3SAT to MST to Terra Mystica

Claim 2.1. 3SAT $\leq_p$ k-LEAF MST $\leq_p$ TERRA MYSTICA

Proof. For an instance of 3SAT with n variables $x_1, \ldots, x_n$ and m clauses, create a graph as follows:

- For each variables x_i create two vertices v_i^0, v_i^1 with cost 1. Create an edge between each two vertices of these $2n$ vertices, i.e. the $2n$ vertices induce a clique.
- For each variable x_i, create a vertex v_i with cost 0, and two edges $(v_i, v_i^0), (v_i, v_i^1)$.
- For each clause c_l with variables x_i, x_j, x_k, create a vertex u_l with cost 0, and three edges $(u_l, v_i^{b_{li}})$ (if the literal is positive $b_{li} = 1$, otherwise $b_{li} = 0$), $u_l, v_j^{b_{lj}}$, $u_l, v_j^{b_{lk}}$.

Then we ask if there is a connected vertex set T with size at least $2n + m$ and the sum of the cost is no more than n. Now, if the instance of 3SAT has a feasible solution where $x_i = b_i$, we can make T consist of all v_i's, u_l's and $v_i^{b_i}$'s.

On the other hand, if T exists, because you can afford at most n, at most n vertices among v_i^0, v_i^1 are chosen, thus all $n + m$ vertices with cost 0 must be chosen. This means:

- Since v_i is chosen, at least one vertex of v_i^0 and v_i^1 is chosen. Because at most n vertices among v_i^0, v_i^1 are chosen, exactly one of v_i^0 and v_i^1 is chosen. We assign x_i accordingly.
- Since u_l is chosen, at least one corresponding literal has value 1, thus the clause is satisfied.

So the 3SAT instance has a feasible solution.

In both directions above, a connected vertex set T satisfying all the conditions exists if and only if a tree subgraph spanning T exists and also satisfies all the conditions. We have established both reductions in the claim. $\square$

2.1. Drawbacks of this approach

One of the main drawbacks of this approach is that it seems difficult to make the constructed graph G a *planar* one. TERRA MYSTICA is a computer game played in a 2D world, more precisely, on the faces of a hexagon tessellation of the plane - or rather, a finite subsets of such hexes. We will see that in the second approach, one can easily adapt the construction to this additional requirement.

3. Second approach

We reduce from k-MST with edge weights in $\{1, 2, 3\}$, for the definition of k-MST, refer to [3]. Let us recap the decision version of our problem: *A graph $G = (V, E)$ is given, together with an edge cost function $W : E \to \{1, 2, 3\}$, a target size t and a cost limit c are the rest of the input instance. An instance is a YES-instance if there exists a t-vertex tree with total edge cost at most c.* Note that in order for this second approach to work, we utilize the fact at the end of the proof of claim 2.1.

Claim 3.1. *k-MST* $\leq_p$ TERRA MYSTICA

Proof. To turn a k-MST instance into an instance of the TERRA MYSTICA network decision problem, add a vertex v_e on the middle of each edge e; replace $e = \{v, w\}$ with two edges $\{v, v_e\}$ and $\{w, v_e\}$. Let $C(v_e) = W(e)$, and let $C(v) = 0$ for all $v \in V$. Also attach a path of length $|E|$ and total cost 0 to each vertex in V. Let $k = t(|E|+1)+(t-1)$ and let $s = c$.

If we can find a k-MST, we can easily turn it into a TERRA MYSTICA network: simply use v_e in the network for all edges e used in the tree, carry over the vertices and flood-fill into the attached paths. This obeys the cost constraint because they are equal and the flood-filled vertices are free. Each vertex turned into $|E| + 1$ vertices, all of which are in the solutions, as are $t - 1$ edges, so the toal solution size is $t(|e| + 1) + (t - 1)$, exactly what is required. Trees are connected, so the TERRA MYSTICA network is as well.

On the other hand, let a (connected) TERRA MYSTICA network be given of size at least $t(|E| + 1) + (t - 1)$ and cost at most $s = c$. If it includes at most $t - 1$ of the vertices carried over from the k-MST instance, and at most all the v_e's, then its total size is at most

$$(t - 1)(|E| + 1) + |E| = t|E| - |E| + t - 1 + |E| = t|E| + t - 1 = t(|E| + 1) - 1 < t(|E| + 1) + (t - 1)$$

This is because each of the $|E|$-long paths is only accessible through the vertices carried over from V. But this is impossible, as we assumed the solution size is at least $t(|E| + 1) + (t - 1)$.

If it has more than t of the vertices from V, we can drop some of them while keeping the solution connected: t vertices plus t paths of length $|E|$ plus $(t - 1)$ of the v_e vertices to connect them also form a solution; we may need to flood-fill into partially included attached paths. If the solution includes more than $t - 1$ of the v_e vertices, we can eliminate some by computing any (e.g. a minimum-weight) spanning tree only on the vertices and edges in the solution (treating the v_e vertices as edges). The solution includes *at least* $t-1$ of the v_e vertices, else it would not be connected.

Then the vertices from V and the edges corresponding to the v_e vertices form a t-vertex tree in G with cost at most c. So if we have a solution to the TERRA MYSTICA network problem, we also have a solution to the k-MST problem. □

3.1. Bring back the required planarity

As mentioned earlier in this article, TERRA MYSTICA is played on the faces of a hexagon tessellation of the plane. The utilized k-MST problem in this second approach is still hard for planar graphs with edge weights still in $\{1, 2, 3\}$. The reduction in this section preserves planarity. Hence, with this reduction we can abstract away fewer parts of TERRA MYSTICA than the first approach.

One step is still missing: we need to embed a planar graph on a hexagonal-face grid. A river may be put somewhere with more effort while keeping the reduction from falling apart.

4. Conclusion

From time to time, one needs to ask one's self whether mathematics in this century is really improving ever since the previous one. Practically, one is gotten used to tapping into Internet regularly, for at least over the few last decades, and sometimes one really finds novel maths tricks which one manages to nourish into one's own theorems. We emphasize again that if one has ever loved reading maths books of Sharygin, one would question the capability of some day proving some mathematical conjecture.

References

[1] Michael R. Garey, David S. Johnson, **Computers and Intractability: A Guide to the Theory of NP-completeness**

[2] The free encyclopedia Wikipedia, *https://en.wikipedia.org/wiki/NP-completeness*

[3] The free encyclopedi Wikipedia, *https://en.wikipedia.org/wiki/K-minimum_spanning_tree*