On the Graph Turnpike Problem

Tomás Feder* Rajeev Motwani†

1 Introduction

In the *graph turnpike problem without distinctness*, we are given a graph $G = (V, E)$ with positive weights $w_e > 0$ on each edge e of G. The aim is to map each vertex v of G to a point $f(v)$ in the real line so that $d(f(u), f(v)) = |f(u) - f(v)| = w_e$ for each edge $e = uv$ of G.

In the *graph turnpike problem with distinctness*, we additionally require $f(u) \neq f(v)$ for $u \neq v$.

In this paper we prove results on this problem, including its NP-completeness (even with small weights), and an $O(m + n \log n)$ algorithm when the graph contains a vertex adjacent to all other vertices, where $n = |V|$ and $m = |E|$.

The usual turnpike problem has all pairwise distances given, but does not specify which pair of vertices w_e corresponds to, that is, the $\binom{n}{2}$ values w_e are given as multisets. This problem is much harder and has been studied in [10] and [4].

There are two other problems that can be viewed as special cases of the graph turnpike problem. The *bandwidth* of an n-vertex graph G is the minimum over all permutations $v_1, \ldots, v_n$ of the vertices of the maximum over all edges (v_i, v_j) of $|i - j|$ [3, 7, 8]. Thus the bandwidth problem can be viewed as approximating solutions to the turnpike problem when the edges are restricted to weights $w_e = 1$ and the points $f(v)$ are restricted to the points $1, 2, \ldots, n$. The other problem is the *low-distortion graph embedding* problem, where the aim is to embed a graph in a metric space, to approximate as best as possible the distance between vertices [2, 6, 9]. This corresponds to the case where the graph of the graph turnpike problem is a clique, as all distances matter, not just particular adjacencies, and the target metric space for low-distortion embedding is the real line.

2 NP-completeness for Cycles

Theorem 1 *The graph turnpike problem without distinctness for graphs G that are cycles is NP-complete.*

Proof. In the NP-complete *partition problem* we are given a colection of n weights $w_i > 0$ and the aim is to partition these weights into two subsets that add up to the same. Form a cycle G of length n with weights w_i on the n edges. The aim for the turnpike problem is then to orient the edges with weights w_i in either direction so that when we traverse the whole cycle in the real line we return to a chosen starting point for the cycle. This is the same as assigning signs to the w_i so that they add up to zero, which is in turn the same as requiring the weights assigned positive sign to add up to the same as the weights assigned negative weights, which is the partition problem.

*268 Waverley Street, Palo Alto, CA 94301. Email: tomas@theory.stanford.edu

†Department of Computer Science, Stanford University, Stanford, CA 94305. Supported in part by NSF Grants IIS-0118173, EIA-0137761, and ITR-0331640, and grants from Microsoft, SNRC, and Veritas. Email: rajeev@cs.stanford.edu

This proves NP-completeness. ◻

Theorem 2 *The graph turnpike problem with distinctness for graphs G that are cycles is NP-complete.*

Proof. We carry out the same reduction as for the graph turnpike without distinctness problem from the partition problem. Now we must make sure that when a solution exists, no partial path of the cycle adds up to zero. This we guarantee by inserting a large weight M after each w_i in the cycle, with M larger than the sum of the w_i, and inserting in addition a single weight nM anywhere in the cycle. A solution to partition will necessarily put all n weights M in one side and the single weight nM in the other, guaranteeing distinctness. ◻

3 Easiness and Hardness of Approximation

Theorem 3 *If G has a vertex v that is adjacent to every vertex $u \neq v$ in G, then both the graph turnpike problem with and without distinctness are polynomial time solvable, in time $O(m)$ without distinctness and in time $O(m + n \log n)$ with distinctness. Thus adding at most $n - 2$ edges with corresponding weights makes the problem makes the problem tractable on general graphs G.*

Proof. Consider first the turnpike problem without distinctness. If $G - v$ has more than one component, it suffices to consider each component separately together with v. If $G - v$ has edges ab and bc, then the triangles abv and bcv can be mapped uniquely once bv is mapped, so we may determine the distance between a and c, identify a with c if this distance is zero, and otherwise add the edge ac. Repeatedly performing this operation when $G - v$ is connected transforms G into a clique which can be mapped in a unique way up to choice of $f(v)$ and orientation, since triangles determine betweenness for the three points assigned to the vertices of the triangle. This completes the algorithm for the turnpike problem without distinctness. The process of adding vertices in a component takes time $O(m)$, giving a linear time algorithm for the graph turnpike problem without distinctness.

For the turnpike problem with distinctness, whenever we have that the distance between a and c has to be zero there is no solution, otherwise we obtain a clique for each component of $G - v$ that is mapped in a unique way up to choice of $f(v)$ and orientation. We then assign a Boolean variable to each clique for its choice of orientation, and impose a 2-satisfiability clause between any two such Boolean variables if some choice of orientations for the two corresponding cliques is not allowed by distinctness. We finally solve the 2-satisfiability instance to obtain the orientations for the solution. In addition to the $O(m)$ time algorithm without distinctness, testing whether points coincide takes $O(n \log n)$ time for sorting, and the 2-satisfiability instance solves the problem of testing whether a graph is bipartite which can be done in $O(m)$ time, for a total $O(m + n \log n)$ time.

If all vertex degrees are at most one, then the problem is easy, as the problem decomposes into isolated edges and vertices that can be easily mapped. So we may assume some vertex v has degree 2, and adding $n - 2$ edges makes v adjacent to all other edges, giving polynomiality. ◻

We prove a counterpart to the last statement of the Theorem. Say that a problem A is the *weakly NP-complete* version of an NP-complete problem B if A consists of $n^{1-\epsilon}$ instances of B of size n^ϵ that have a solution, for some constant $0 < \epsilon < 1$, and the aim in A is to find a solution to at least one of these instances of B.

Theorem 4 *The graph turnpike problem with and without distinctness remain weakly NP-complete even if we are allowed to guess additional $n^{1-\epsilon}$ edges with their corresponding weights.*

Proof. We let G be $3n^{1-\epsilon}$ disjoint cycles of length $n^\epsilon/3$. At least one of these cycles will have no edge added and will remain NP-complete as before, proving weak NP-completeness. $\square$

4 Chordal and Bounded Treewidth Graphs Without Distinctness

Theorem 5 *If G is chordal, then the graph turnpike problem without distinctness is polynomial time solvable.*

Proof. Remove from G a vertex v whose neighbors form a clique K, and solve the problem for $G-v$ recursively. The solution thus found for K is unique up to translation and choice of direction, and we may thus determine if v can be mapped as well by placing v with regards to K, completing the recursion. $\square$

Theorem 6 *If G has bounded treewidth, then the graph turnpike problem without distinctness is polynomial time solvable for instances whose weights are integers bounded by a polynomial in n.*

Proof. A bounded treewidth graph recursively has a separator subset of constant size, and we may consider all possible ways of mapping this separator recursively since the weights are bounded by a polynomial in n. $\square$

5 Instances with Small Integer Weights

Theorem 7 *For the graph turnpike problem without distinctness, if all weights are 1 then the problem is equivalent to testing bipartiteness and is thus polynomial, while if all weights are 1 or 2 then the problem is NP-complete.*

Proof. If all weights are 1, then an odd cycle cannot be mapped, so the instance must be bipartite to have a solution, and in that case we can choose two points at distance 1 from each other to map the two sides of the bipartition.

If all weights are 1 or 2, we consider an instance H of 3-colorability. We first encode two vertices x and y being at integer distance d from each other in the mapping by using a path $x = z_0, z_1, \ldots, z_d = y$ with edges of weight 1 and adding edges of weight 2 joining z_i to z_{i+2}. For each vertex v in H we include a vertex v in G. For any two vertices u, v of H we join u and v by a path consisting of four edges of weight 1. Thus all vertices of H in G will map to points at distance 0,2, or 4 from each other, and we may assume they will thus map to the points 0,2,4. Finally for each edge uv in H we join u and v in G by a path consisting of an edge of weight 4 (encoded with 1 and 2 as described for $d = 4$ above) followed by two edges of weight 1. This requires u, v to map to points distance 2 or 4 if uv is an edge of H. Thus all vertices of H are vertices of G that map to 0,2, or 4, and if uv is an edge of H then u, v map to distinct values out of 0,2,4. Therefore 3-colorability of H is encoded by G with the three colors represented by 0,2,4, proving NP-completeness. □

Theorem 8 *For the graph turnpike problem with distinctness having integer weights* $1, 2, \ldots, L$*, if* $L = 1$ *then the problem is equivalent to testing whether* G *is a union of disjoint paths, if* L *is a constant then the problem is polynomial time solvable, and for* L *polynomially bounded the problem is NP-complete.*

Proof. If $L = 1$, then no vertex of G can have three or more neighbors if there is a solution, as two of these neighbors would coincide. Furthermore, if a vertex v has two neighbors, these neighbors must go to the two sides of v. This proves that G must be a union of disjoint paths.

If L is a constant, then any interval of size L separates the vertices in either side and contains at most L vertices, so by dynamic programming we can guess this L-vertex separator having a solution for the vertices to the left of this separator, giving polynomiality.

If L is polynomially bounded, we do a reduction from 4-colorability of H. Choose two large integers k and r, include in G a path $x_0, x_1, \ldots, x_r$ with edges of weight k, and join x_0, x_r with an edge of weight kr. We may then assume that x_i will map to ik. We reserve different x_i for the different vertices of H, and also reserve additional x_i for the different edges of H, so that for each edge of H we reserve some $x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}$.

If we have reserved x_i for vertex v in H, then we require that v in G map to one of $x_i - 3, x_i - 1, x_i + 1, x_i + 3$ corresponding to the four possible colors for v in H. This is done by joining x_i to v by a path consisting of an edge of weight 1 followed by an edge of weight 2. If we have reserved $x_j, \ldots, x_{j+4}$ for the edge uv in H, then we assume x_j is reserved for u and x_{j+4} is reserved for v as well, encoding the four possible colors around x_j and x_{j+4} similarly as we did around x_i before with paths starting at x_j or x_{j+4} consisting of an edge of weight 1 and an edge of weight 2. As for u we have reserved both a vertex around x_i and around x_j, we join these two vertices by an edge of weight $(j - i)k$, so the two encodings of the possible colors will coincide. Similarly, as for v we have reserved both a vertex around $x_{i'}$ and around x_{j+4}, we join these two vertices by an edge of weight $(j + 4 - i')k$, so the two encodings of the possible colors will coincide. Finally, join u around x_j and v around x_{j+4} by a path consisting of seven edges of respective lengths $k, 2, k, 2, k, 2, k$. As each of the three occurences of 2 in this path can correspond to 2 or -2, this guarantees that the color out of the four possible colors around x_j and the color out of the four possible colors around x_{j+4} differ in either direction by 2,4, or 6 and are thus different. This completes the encoding of 4-colorability of H and proves NP-completeness. □

6 Uniquely Mappable and Clicable Instances

We say that an instance of the turnpike problem with or without distinctness is *uniquely mappable* if there exists at most one solution up to translation and choice of orientation. We say that an instance of the turnpike problem with or without distinctness is *clicable* if whenever we repeatedly perform the operation of adding the edge *ac* for triangles *abd* and *bcd* in the case with distinctness, or perform either this operation or identify a with c in the case without distinctness, we eventually transform G into a clique K.

Theorem 9 *If* G *is clicable then* G *is uniquely mappable.*

Proof. The operations of adding ac or identifying a and c are justified by the fact that mapping bd determines the mapped position for a and c due to the triangles abd and bcd. The resulting clique K is uniqely mappable since triangles determine betweenness. □

Theorem 10 *There exists a unique mappable clicable G regardless of weights for the graph turnpike problem with or without distinctness that has $2n-3$ vertices. For the graph turnpike problem without distinctness, if G is uniquely mappable regardless of weights, then G has at least $1.5(n-1)$ vertices for $n \geq 3$.*

Proof. Let G be a tree T plus a vertex v adjacent to all of G. The graph G is clicable and thus uniquely mappable, and G has $2n-3$ vertices.

To obtain the lower bound for $n \geq 3$, if G has no triangles, then we may choose a vertex v and identify all its neighbors. This makes v of degree one in a graph with at least three vertices, and v can be mapped in either direction, contrary to G being uniquely mappable. The lower bound for $n = 3$ follows from the existence of a triangle. Suppose we have identified a set of vertices U in G of size $n \geq 3$ so that the subgraph of G induced by U satisfies the bound for n. If some vertex u of $G - U$ is adjacent to at least two vertices of U, then $U \cup \{u\}$ satisfies the bound for $n+1$. If G has a triangle uvw with $u \in U$ and $v, w \in G - U$, then $U \cup \{v, w\}$ satisfies the bound for $n+2$. Otherwise we may identify all the neighbors in $G-U$ of each vertex $u \in U$, and the edges joining U to $G-U$ will form a matching whose edges can be given all the same weight and oriented in either direction but all the same direction, once U and $G-U$ have been separately mapped, contrary to uniqueness. This completes the induction. □

7 Approximation Algorithm

Theorem 11 *Let G be an instance of the graph turnpike problem with integer weights $1 \leq w_i \leq c = 2^k - 1$ that has a solution. Then G can be $(c+1)$-colored in polynomial time and thus a solution with new integer weights $1 \leq w'_i \leq c$ can be found, giving an approximate solution for the original problem since $1/c \leq w'_i/w_i \leq c$.*

Proof. Write $w_i = 2^k_i(2r_i + 1)$ for integers $0 \leq k_i < k$ and $r_i \geq 0$. The edges of weights w_i that have the same k_i form a bipartite graph, as an odd cycle would add up to an odd multiple of 2^{k_i}. Let S_j, T_j be the two sides of the biparition for $0 \leq k_i = j < k$. Each vertex has two choices for each such j, namely whether to belong to S_j or to T_j. This gives a total of 2^k choices over all such j, and each such choice gives an independent set. Thus G is 2^k-colored, or $(c+1)$-colored for $c = 2^k - 1$. Assigning the $c+1$ colors to the points $0, 1, 2, \ldots, c$ on the line gives a solution with new weights $1 \leq w'_i \leq c$ and the desired approximation ratio. □

If the w_i with $1 \leq w_i \leq c = 2^k - 1$ are not required to be integer, then the points in the assumed solution can be moved to the nearest half-integer, so that each w_i is replaced by either $\lfloor w_i \rfloor$ or $\lceil w_i \rceil$. As we do not know whether the floor or the ceiling of each w_i will occur in a solution, we do not know how to find an approximate solution in this non-integer setting. We prove a weaker result.

Theorem 12 *Let G be an n-vertex, instance of the graph turnpike problem with real weights $1 \leq w_i < 1+1/k$ for some constant integer $k \geq 1$ that has a solution. Then G can be $O(n^{1/(k+1)})$-colored in polynomial time and thus a solution with new weights $1 \leq w'_i \leq O(n^{1/(k+1)})$ can be found, giving an approximate solution for the original problem.*

Proof. Erdös et al. [5] showed that if G contains no cycle of length $2k+1$, then G has an independent set of size at least $c(k)n^{k/(k+1)}$ for some constant $c(k)$ dependent on the constant k. Such an independent set can be constructed from the proof in polynomial time. Thus G can be $O(n^{1/(k+1)})$-colored. The proof is completed by noticing that if $1 \leq w_i < 1 + 1/k$ then no cycle of length $2k+1$ can exist in G. Given the coloring, we may assign values at increments of 1 on the line to the colors, giving $1 \leq w'_i \leq O(n^{1/(k+1)})$. $\square$

Theorem 13 *Let G be an n-vertex, instance of the turnpike problem with real weights $1 \leq w_i < k$ for some constant integer $k \geq 2$ that has a solution. Then G can be $O(n^{1-1/k})$-colored in polynomial time and thus a solution with new weights $1 \leq w'_i \leq O(n^{1-1/k})$ can be found, giving an approximate solution for the original problem.*

Proof. It is known that if G contains no clique of size $k+1$, then G can be $O(n^{1-1/k})$-colored in polynomial time. The proof is completed by noticing that if $1 \leq w_i < k$ then G contains no clique of size $k+1$. Given the coloring, we may assign values at increments of 1 on the line to colors, giving $1 \leq w'_i \leq O(n^{1-1/k})$. $\square$

References

[1] The turnpike problem, http://geomblog.blogspot.com/2004/03/turnpike-problem.html

[2] M. Badoiu, J. Chuzhoy, P. Indyk and A. Sidiropoulos, Low-distortion embeddings of general metrics into the line, In Proc. ACM Symp. Theory of Computing (2005).

[3] F.R.K. Chung, Labelings of graphs, Selected topics in graph theory, 3, Academic Press (1988), 151–168.

[4] T. Dakić, On the turnpike problem, doctoral thesis, School of Computing Science, Simon Fraser University (2000).

[5] P. Erdös, R. Faudree, C. Roussheau and R. Schelp, On cycle-complete graph Ramsey numbers, J. Graph Theory 2 (1978), 53–64.

[6] J. Fakcharoenphol, S. Rao and K. Talwar, A tight bound on approximating arbitrary metrics by tree metrics, in Proc. 35th Ann. ACM Symp. Theory of Computing (2003).

[7] U. Feige, Approximating the bandwith via volume respecting embeddings, J. Computer and System Sciences 60 (2000), 510–529.

[8] A. Gupta, Improved bandwidth approximation for trees and chordal graphs, J. Algorithms 40 (2001), 24–36.

[9] A. Gupta, Embedding tree metrics into low dimensional Euclidean spaces, Discrete Comput. Geom. (1999).

[10] P. Lemke, S.S. Skiena and W.D. Smith, Reconstructing sets from interpoint distances, DIMACS Technical Report 2002-37 (2002).