A Note on the Asymptotics and Computational Complexity of Graph Distinguishability

Alexander Russell
acr@cs.utexas.edu
Department of Computer Science
University of Texas at Austin
Austin, TX 78712

Ravi Sundaram
koods@delta-global.com
Delta Global Trading
141 Tremont Street, 12th Floor
Boston, MA 02111

Submitted: February 2, 1998; Accepted: March 24, 1998

Abstract

A graph G is said to be *d-distinguishable* if there is a d-coloring of G which no non-trivial automorphism preserves. That is, $\exists\chi : G \to \{1,\dots,d\}$,

$$\forall\phi \in \mathrm{Aut}(G)\setminus\{\mathbf{id}\}, \exists v, \chi(v) \neq \chi(\phi(v)).$$

It was conjectured that if $|G| > |\mathrm{Aut}(G)|$ and the $\mathrm{Aut}(G)$ action on G has no singleton orbits, then G is 2-distinguishable. We give an example where this fails. We partially repair the conjecture by showing that when "enough motion occurs," the distinguishing number does indeed decay. Specifically, defining

$$\mathfrak{m}(G) = \min_{\substack{\phi\in\mathrm{Aut}(G)\\ \phi\neq\mathbf{id}}} |\{v \in G : \phi(v) \neq v\}|,$$

we show that when $\mathfrak{m}(G) > 2\log_2|\mathrm{Aut}(G)|$, G is indeed 2-distinguishable. In general, we show that if $\mathfrak{m}(G)\ln d > 2\ln|\mathrm{Aut}(G)|$ then G is d-distinguishable.

There has been considerable interest in the computational complexity of the d-distinguishability problem. Specifically, there has been much musing on the computational complexity of the language

$$\{(G,d) : G \text{ is } d\text{-distinguishable}\}.$$

We show that this language lies in $\mathrm{AM} \subset \Sigma_2^P \cap \Pi_2^P$. We use this to conclude that if DIST is coNP-hard then the polynomial hierarchy collapses.

AMS Classification: Primary: 05C25; Secondary: 68Q15.

1 Introduction

An undirected graph G is *d-distinguishable* if there is a d coloring of G which no non-trivial automorphism preserves. Formally, we write $\exists\chi : G \to \{1,\ldots,d\}$,

$$\forall\phi \in \mathrm{Aut}(G) \setminus \{\mathbf{id}\}, \exists v, \chi(v) \neq \chi(\phi(v)),$$

where $\mathrm{Aut}(G)$ denotes the collection of automorphisms of the graph G and **id** denotes the identity map. One says that such a coloring "destroys the symmetries" of G. Every graph G is then $|G|$-distinguishable and a graph is 1-distinguishable exactly when it is *rigid*, i.e. $|\mathrm{Aut}(G)| = 1$. The smallest d for which G is d-distinguishable is dubbed the *distinguishing number* of G, denoted $\mathfrak{d}(G)$. An instantiation of this machinery, mentioned in [1], is the problem of coloring keys on a (circular) key chain so that one can uniquely identify each key. (In this case, one is interested in the distinguishing number of the dihedral groups.)

A paper of Albertson and Collins [1] gracefully develops the theory of distinguishability in several directions. They conjectured that if $|G| > |\mathrm{Aut}(G)|$ and the action of $\mathrm{Aut}(G)$ on G has no singleton orbits, then $\mathfrak{d}(G) = 2$. Though there are graphs for which this fails[1], the idea that *few colors suffice if every automorphism moves many vertices* can be substantiated. Specifically, for an automorphism $\phi \in \mathrm{Aut}(G)$, define the *motion* of ϕ as

$$\mathfrak{m}(\phi) = |\{v \in G : \phi(v) \neq v\}|.$$

The *motion* of a graph G is then

$$\mathfrak{m}(G) = \min_{\substack{\phi\in\mathrm{Aut}(G)\\ \phi\neq\mathbf{id}}} \mathfrak{m}(\phi).$$

We show that when $\mathfrak{m}(G) > 2\log_2|\mathrm{Aut}(G)|$, G is 2-distinguishable. More generally, when $\mathfrak{m}(G)\ln d > 2\ln|\mathrm{Aut}(G)|$, G is d-distinguishable.

Another natural question is that of the computational complexity of the graph distinguishability problem (see the discussion in [1]). Specifically, one would like to place the language

$$\textsc{Dist} = \{(G,d) : \mathfrak{d}(G) \leq d\},$$

as low in the natural hierarchy of complexity classes as possible. There is no obvious NP algorithm for this language; the only immediate conclusion is that DIST $\in \Sigma_2^P$. We show that $\textsc{Dist} \in \mathrm{AM} \subset \Pi_2^P \cap \Sigma_2^P$.

2 Graphs with Large Motion can be Distinguished with Few Colors

We now return to the first theorem advertised in the introduction, namely

Theorem 1. *If* $\mathfrak{m}(G) > 2\log_2|Aut(G)|$ *then G is 2-distinguishable.*

[1] Select a large rigid graph H and let G_H be the graph formed by the disjoint union of K_3 and 3 copies of H. Then $\mathrm{Aut}(G_H) = S_3 \times S_3$, G_H has no one cycles, $\mathfrak{d}(G_H) = 3$, and $|G_H|$ can be arbitrarily large.

Anticipating the proof, we define the *cycle norm* as follows: decomposing an automorphism ϕ into a product of disjoint cycles

$$\phi = (v_{11}v_{12}\dots v_{1l_1})(v_{21}\dots v_{2l_2})\dots(v_{k1}\dots v_{kl_k}),$$

the *cycle norm* of ϕ is the quantity

$$\mathfrak{c}(\phi) = \sum_{i=1}^{k}(l_i - 1).$$

The cycle norm is relevant to graph distinguishability in the following sense. Suppose that a graph G is randomly two-colored by independently assigning each vertex a color uniformly from $\{\text{red}, \text{black}\}$. Then the probability that every cycle of an automorphism ϕ is monochromatic is exactly $2^{-\mathfrak{c}(\phi)}$. When this event occurs, the automorphism ϕ preserves the coloring so chosen.

For convenience, the cycle norm of a graph G is defined

$$\mathfrak{c}(G) = \min_{\substack{\phi\in\mathrm{Aut}(G)\\ \phi\neq\mathbf{id}}} \mathfrak{c}(\phi).$$

Notice that for any automorphism, $\mathfrak{c}(\phi) \geq \mathfrak{m}(\phi)/2$. Of course, then $\mathfrak{c}(G) \geq \mathfrak{m}(G)/2$. With this observation, Theorem 1 above is an easy consequence of the following theorem:

Theorem 2. *If $\mathfrak{c}(G)\ln d > \ln|Aut(G)|$ then G is d-distinguishable.*

Proof. We study the behavior of a random d-coloring of G, the probability distribution given by selecting the color of each vertex independently and uniformly in the set $\{1,\dots,d\}$. Fix an automorphism $\phi \neq \mathbf{id}$ and consider the bad event that the random coloring χ is in fact preserved by ϕ: an easy calculation shows that

$$\Pr_{\chi}[\forall v, \chi(v) = \chi(\phi(v))] = (\frac{1}{d})^{\mathfrak{c}(\phi)} \leq (\frac{1}{d})^{\mathfrak{c}(G)}.$$

Collecting together these bad events, we have

$$\Pr_{\chi}[\exists\phi \neq \mathbf{id}, \forall v, \chi(v) = \chi(\phi(v))] \leq |\mathrm{Aut}(G)|(\frac{1}{d})^{\mathfrak{c}(G)}.$$

The hypothesis of the theorem is exactly that this quantity is less than one, in which case there exists a coloring χ for which $\forall\phi \neq \mathbf{id}, \exists v, \chi(v) \neq \chi(\phi(v))$, as desired. □

For a delightful survey of the probabilistic method, of which the above is an example, see [2].

It is interesting to notice that the above theorem is actually tight in the case of the dihedral groups $D_3, D_4, \dots$ mentioned in the introduction (and in [1]). (The answers are $\mathfrak{d}(D_3) = 3, \mathfrak{d}(D_4) = 4, \mathfrak{d}(D_5) = 2, \mathfrak{d}(D_6) = 2, \dots$.)

3 DIST $\in$ AM

Though we will discuss the definition of AM in some detail, for a general introduction to complexity theory and detailed discussions of the polynomial time hierarchy and AM, we refer the reader to [9] and [4, 5].

The polynomial time hierarchy is the "polynomial time bounded variant" of the Kleene hierarchy of recursive function theory. One defines $\Sigma_0^P = \Pi_0^P = \mathrm{P}$, and, in general, $L \in \Sigma_k^P$ if there is a polynomial p and $D \in \Pi_{k-1}^P$ for which

$$L = \{w : \exists \pi, |\pi| \leq p(|w|), \langle w, \pi \rangle \in D\}.$$

Finally, define the class Π_k^P to consist of all languages L for which $\overline{L} \in \Sigma_k^P$. Above, the notation $\langle \cdot, \cdot \rangle$ refers to some canonical pairing function. With these definitions, $\mathrm{NP} = \Sigma_1^P$, $\mathrm{coNP} = \Pi_1^P$, and the classes Σ_k^P and Π_k^P form a neat hierarchy containing P and lying inside PSPACE.

Considering the quantifier alternation in the definition of the distinguishability problem, it is not surprising that DIST $\in \Sigma_2^P$, as an easy argument shows. Our goal here is to show that DIST $\in \mathrm{AM} \subset \Sigma_2^P \cap \Pi_2^P$.

AM is the class of languages for which there are *Arthur–Merlin* games (see [3]). Intuitively, an Arthur–Merlin game for a language L is played by two players, *Arthur*, equipped with a random coin and only modest (polynomial-time bounded) computing power and *Merlin*, who is computationally unbounded. Both Arthur and Merlin are supplied with a word x, and the goal of the game is for Arthur to determine if $x \in L$. Arthur, based on his coin flips, may ask Merlin a constant number of questions, and, having heard Merlin's answers, must then decide to *accept* that $x \in L$ or *reject* this statement. Of course, a natural question for Arthur to ask is, "$x \in L$?" Unfortunately, rather than being the trustworthy advisor we might hope, Merlin actually has a vested interest in seeing that Arthur accept the predicate. An Arthur–Merlin game, then, is a strategy for Arthur to follow in his questioning of Merlin so that:

- When $x \in L$, regardless of Arthur's coin tosses (which may determine the questions he asks of Merlin under this strategy), Merlin can answer satisfactorily, convincing Arthur to accept that $x \in L$.

- When $x \notin L$, regardless of way in which Merlin answers, the discussion ends with Arthur rejecting that $x \in L$ a constant fraction of the time. (The probability distribution is taken over Arthur's coin tosses.)

The number of questions which Arthur is allowed to ask may depend on the language, but not the specific input. Furthermore, the entire conversation must have length polynomial in the length of the input. In the above model, Arthur's coin flips are *public*– Merlin can see them.

Hopefully, it is clear from this vague definition that every language in NP has an (easy) Arthur-Merlin game. We will show that there is an Arthur-Merlin game for the language DIST. First, a formal definition:

Definition 1. *For a function $M : \{0,1\}^* \to \{0,1\}^p$, and random variables $X_1, X_2, \dots, X_R \in \{0,1\}^p$, let*

$$M_X = (M(X_1), M(\langle X_1, X_2\rangle), \dots, M(\langle X_1, \dots, X_R\rangle)).$$

AM *consists of those languages L for which there exists a constant R, a polynomial p, and a polynomial time bounded Turing machine A so that:*

- $x \in L \Rightarrow \exists M : \{0,1\}^* \to \{0,1\}^{p(|x|)}$,

  $$\Pr_{\{X_i\}}[A(x, X_1, \dots, X_R, M_X) \textit{ accepts}] = 1,$$

  where the X_i are independent uniform random variables on $\{0,1\}^p$.

- $x \notin L \Rightarrow \forall M : \{0,1\}^* \to \{0,1\}^{p(|x|)}$,

  $$\Pr_{\{X_i\}}[A(x, X_1, \dots, X_R, M_X) \textit{ accepts}] \leq \frac{1}{2},$$

  where the X_i are independent uniform random variables on $\{0,1\}^p$.

We start by showing that the language of rigid graphs is in AM. Let

$$\textsc{Rigid} = \{G : |\mathrm{Aut}(G)| = 1\}.$$

Theorem 3. Rigid $\in$ AM

Proof. The proof is an easy adaptation of the result of [7, 8] that the language

$$\mathrm{NGI} = \{(G_1, G_2) : G_1 \not\cong G_2\}$$

is in *AM*. In the formulation of AM given above, Merlin observes Arthur's coin tosses. This scenario is aptly dubbed the "public" coin model. In fact, in the formalization above, Arthur's questions to Merlin are exactly his coin tosses (the random variables X_i in the above definition). Since Arthur is deterministic aside from his coin tossing, any question he might wish to have answered can be anticipated and duly answered by Merlin. In the alternative model, involving "private" coin tosses, Arthur's questions may not completely reveal the coins he has tossed so far. It is rather remarkable that the two models are in fact equivalent [8]. We shall allow ourselves the flexibility of a private coin in our constructions. Our goal is to show that Rigid $\in$ AM. Given input $G = ([n], E)$, consider the following protocol:

1. Arthur generates a random permutation $\sigma \in S_n$, and sends Merlin the graph $G_\sigma = ([n], E_\sigma)$, where
   $$E_\sigma = \{(\sigma(u), \sigma(v)) : (u, v) \in E\}.$$
2. Arthur expects Merlin to respond with an element of S_n. Given any other response, Arthur rejects. Upon receiving $\tau \in S_n$. Arthur accepts exactly if $\tau = \sigma$.

Notice that when G is rigid, there is a unique isomorphism between G and G_σ, so that Merlin does indeed have a strategy which always convinces Arthur to accept. Suppose instead that G is non-rigid so that $|\mathrm{Aut}(G)| > 1$. In this case, there are exactly $|\mathrm{Aut}(G)|$ isomorphisms between G and G_σ and, furthermore, conditioned on Arthur asking the question G_σ to Merlin, each of these isomorphisms is equally like to be the one used by Arthur to construct G_σ. Hence no strategy of Merlin can induce accepting behavior in Arthur for more than a $|\mathrm{Aut}(G)|^{-1} \leq \frac{1}{2}$ fraction of Arthur's coin tosses. □

Theorem 4. $\textsc{Dist} \in \mathrm{AM}$.

Proof. Let $(G = ([n], E), k)$ be the common input, and consider the following protocol:

1. Arthur expects Merlin to send him $\chi : G \to [k]$, a k-coloring of G. On any other message, Arthur rejects.

2. Arthur builds a new graph G' follows. Starting with G, Arthur adds for every vertex v of G a fresh $(n+\chi(v))$-clique, called K_v. Each vertex v, aside from maintaining its old connections inside G is attached to each vertex of K_v. An easy argument shows that the isomorphisms of G' are in one-to-one correspondence with isomorphisms of G which fix χ. Specifically, if χ destroyed all of the symmetries of G, G' is rigid. Arthur now uses the protocol described above for $\textsc{Rigid}$.

It is now easy to check that this protocol satisfies the requirements in the definition of AM. □

Based on constructions like those of [12, 10, 11], one has $\mathrm{AM} \subset \Sigma_2^P \cap \Pi_2^P$, completing the claim in the introduction.

One naturally wonders at the relationship of $\textsc{Dist}$ to more familiar classes such as NP and $\mathrm{\textsc{co}NP}$. In this direction, applying the machinery of [6], we can argue that it is unlikely that $\textsc{Dist}$ is $\mathrm{\textsc{co}NP}$-hard. Specifically, from [6], we have the following theorem:

Theorem 5. *If* $\mathrm{\textsc{co}NP} \subset \mathrm{AM}$, *then the polynomial hierarchy collapses to* Σ_2^P, *specifically* $\Sigma_k^P \subset \Sigma_2^P$ *for all* k.

In our case, were $\textsc{Dist}$ to be $\mathrm{\textsc{co}NP}$-complete, $\mathrm{\textsc{co}NP} \subset \mathrm{AM}$, and we could apply the above theorem. Complementing, this shows that the language

$$\textsc{Robust} = \{(G,k) : \forall \chi : G \to [k], \exists \gamma \in \mathrm{Aut}(G) \setminus \{\mathbf{id}\}, \gamma \text{ preserves } \chi\}$$

is unlikely to be NP hard.

4 An Open Problem

An outstanding open question is whether the language $\textsc{Dist}$ is in fact NP-hard.

5 Acknowledgments

We thank Karen Collins for originally introducing us to the problem during her engaging talk at M.I.T. and several helpful discussions.

References

[1] Michael O. Albertson and Karen L. Collins. Symmetry breaking in graphs. *Electronic Journal of Combinatorics*, 3, 1996. R18.

[2] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. John Wiley & Sons, Inc., 1992.

[3] László Babai and Shlomo Moran. Arthur-merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

[4] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Computer Science*. Springer-Verlag, Berlin, 1988.

[5] José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity II*, volume 22 of *EATCS Monographs on Computer Science*. Springer-Verlag, Berlin, 1990.

[6] Ravi Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25:127–132, 1981.

[7] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal of Computing*, 18(1):186–208, February 1989.

[8] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research*, 5:73–90, 1989.

[9] Johannes Köbler, Uwe Schöning, and Jacobo Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Progress in Theoretical Computer Science. Birkhäuser, Boston, 1993.

[10] Clemens Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17:215–217, 1983.

[11] Alexander Russell and Ravi Sundaram. Symmetric alternation captures BPP. *Computational Complexity*, 6, 1996.

[12] Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 330–335, Boston, Massachusetts, 25–27 April 1983.