On the Complexity of Unfrozen Problems

Adam Beacham Joseph Culberson

April 11, 2001

Abstract We consider questions such as what is the complexity of recognizing instances of (monotonic) NP-complete problems in which no variable is fixed (or *frozen*) by the set of solutions. Since this *unfrozenness* is also a monotonic property in NP, this leads to an inductive sequence of properties for each monotone NP-complete property. In some cases the sequence remains NP-complete, while in others it at some point enters P. Determining the boundaries is particularly challenging. We also consider the related questions of recognizing maximal properties. This study was motivated by results from statistical mechanics being applied to phase transitions of NP-complete problems, which show a correlation of hard instances with a sudden increase in frozen variables.

1 Introduction

Let Ω designate the set of all instances of a problem. A *property* is a subset $\mathbf{X} \subseteq \Omega$ and not-$\mathbf{X} = \Omega \setminus \mathbf{X}$. For graphs we use the notation $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with complement $\mathbf{G}^c$. We say a property is *monotone down with respect to edges*, or simply *monotone*, if $\mathbf{G} \in \mathbf{X} \Longrightarrow \forall \mathbf{E}' \subseteq \mathbf{E}, \mathbf{G}' = (\mathbf{V}, \mathbf{E}') \in \mathbf{X}$. We consider only those properties $\mathbf{X} \in$ NP that are monotone.

Empirical studies such as [1, 16, 7] have correlated a high frequency of very hard instances of NP-complete problems with phase transitions, or thresholds [2], in random classes. However, NP-complete problems without natural such hard regions do exist [10, 17]. It is known that every automorphism invariant monotone graph property has a sharp threshold [9]. The distinction between hard and easy instances appears to be related to whether or not there is a sudden appearance of a backbone [1, 7, 14, 15] in the generation process[3, 2].

We ask questions such as what is the complexity of recognizing instances which are unfrozen (i.e. have no backbone) for monotonic properties in NP. Since this unfrozenness is also a monotonic property in NP, this leads to an inductive sequence of properties for each monotone NP-complete property. In some cases the sequence remains NP-complete, while in others it at some point enters P. The cases do not correlate in an obvious way with the existence or not of hard instances at the threshold. Determining the boundaries is particularly challenging. By considering the complexity of solving backbone free instances, which we call *unfrozen,* we hope to gain insight into the role of frozen variables in the difficulty of the instance.

Maximal properties can be seen as an extension to the generation process in which frozen parts are skipped [1, 3, 7]. Test beds of hidden solution instances are typically generated by selecting a solution which forbids certain parts being generated and then adding (or deleting as appropriate) parts until some desired constraint density is achieved[1, 6]. These methods also sometimes produce the easy-hard-easy phenomenon associated with thresholds. In this paper we note that usually maximal properties are in P, but exhibit two cases where the instances remain hard in the complexity theoretic sense.

Maximal properties and unfrozen properties are also related to certain results in extremal graph theory [2] but typically such results are more concerned with the properties (e.g. number of edges) of the maximum over all maximal graphs.

We use the notation $\mathbf{G}+e$ ($\mathbf{G}-e$) to mean the graph with edge set $\mathbf{E}\bigcup\{e\}$ ($\mathbf{E}\setminus\{e\}$). We use the notation $\mathbf{G}[\mathbf{V}']$ for the induced subgraph of $\mathbf{G}$ on $\mathbf{V}'$, and $\mathbf{G}_1 \times \mathbf{G}_2$ to represent the *join*, that is the union of $\mathbf{G}_1$ and $\mathbf{G}_2$ plus all edges between them. For hypergraphs and satisfiability the only additional requirements are that the arity of the edges or clauses is bounded by a constant (so that the number of edges/clauses in a complete structure is polynomial in the number of vertices/variables). The notation for these cases is a straightforward extension.

For properties such as Hamiltonian Cycle (HC) monotonicity reverses direction. To satisfy our conditions we define the property with respect to the complementary structure, e.g. does $\mathbf{G}^c$ have an HC. When convenient, we revert to the more familiar model in which the property ceases to hold under deletion of edges.

We consider two sets of problems. The standard problems: independent set (IS), clique, vertex cover (VC), Hamiltonian Path/Cycle (HP/HC), coloring (COL), dominating set (DS), k-SAT, not-all-equal-k-SAT(NAE-k-SAT) and 3-ary hypergraph 2-coloring (H3K2). See [11] for definitions.

The other set consists of a special form of subgraph isomorphism which requires a little care to make it monotonic and automorphism invariant.

Definition: *k-complementary subgraph* (k-CSG): $\mathbf{G}$ has this property iff there is a subset $\mathbf{V}' \subseteq \mathbf{V}$ such that $|\mathbf{V}'| = k$ and $\mathbf{G}[\mathbf{V}'] \subseteq \mathbf{G}^c[\mathbf{V}\setminus\mathbf{V}']$. Note the subgraph isomorphism is not induced.

For $1 < k \leq \lfloor n/2 \rfloor$ this is monotonic, since as edges are added, the complement of any k-subset of $\mathbf{V}$ has decreasing edges, while any k vertex induced subgraph has increasing edges. We show in theorem 5.2 it is NP-complete.

2 Frozen Problems

We define a frozen set for any monotonic property $\mathbf{X}$ on graph $\mathbf{G}$ by

$$\mathbf{F}(\mathbf{X},\mathbf{G}) \quad = \quad \{e : \mathbf{G}+e \in \text{not-}\mathbf{X}\} \tag{1}$$

where the edge e can be replaced by the appropriate structures (hyper-edges, clauses) for other problems. Notice that if $\mathbf{G} \in$ not-$\mathbf{X}$, then every e is in $\mathbf{F}$.

We define a property *unfrozen with respect to* $\mathbf{X}$, denoted $\mathbf{U}(\mathbf{X})$, by $\mathbf{G} \in \mathbf{U}(\mathbf{X}) \iff (\mathbf{F}(\mathbf{X}, \mathbf{G}) = \emptyset)$. In lemma 3.1 we show that $\mathbf{U}(\mathbf{X})$ is also monotonic and in NP.

This leads to the inductive definition

$$\mathbf{U}^0(\mathbf{X}) = \mathbf{X} \quad (2)$$

$$\mathbf{U}^i(\mathbf{X}) = \mathbf{U}(\mathbf{U}^{i-1}(\mathbf{X})), i > 0 \quad (3)$$

Equivalently, $\mathbf{G} \in \mathbf{U}^i(\mathbf{X})$ iff $\forall e_1, \ldots, e_i, \mathbf{G} + e_1 + \ldots + e_i \in \mathbf{X}$.

Given a monotone property $\mathbf{X}$, we define the *maximally constrained* property $\max_c(\mathbf{X})$ by $\mathbf{G} \in \mathbf{X}$ and $\mathbf{G}$ is maximal with respect to $\mathbf{X}$; that is, for all $\mathbf{E}' \supset \mathbf{E}(\mathbf{G}), \mathbf{G}' \notin \mathbf{X}$. Equivalently, we can define maximally constrained by $\mathbf{G} \in \max_c(\mathbf{X}) \iff \mathbf{F}(\mathbf{X}, \mathbf{G}) = \mathbf{E}^c$.

Similar definitions apply to bounded arity SAT and hypergraph properties.

For SAT and Constraint problems, a different notion of frozenness is sometimes used [14, 15]. In this case the *backbone* is defined to be the set of variables forced to take on the same value under all satisfying assignments.[1] We define $\widehat{\mathbf{U}}^i(\mathbf{X})$ such that for any instance in this set, for any choice of i variables, and any setting of those variables, there is an extension of that setting that satisfies the instance.

There is frequently a close relation between $\widehat{\mathbf{U}}$ and $\mathbf{U}$. For 3-SAT each frozen clause corresponds to a set of three frozen variables, and vice versa. Only when there are exactly one or two frozen variables is there a distinction between the two notions of unfrozen. Also note that a frozen variable corresponds to a frozen unit clause. On the other hand, for any colorable k-COL instance, or satisfiable instance of NAE-k-SAT, the variables are never frozen. We discuss $\widehat{\mathbf{U}}$ only in the cases of k-SAT and H3K2.

3 Generic Complexity of Frozen Problems

Let us define some additional problems related to these frozenness conditions. The problem $\mathbf{BB}(\mathbf{G}, e, \mathbf{X})$ is to answer whether or not $e \in \mathbf{F}(\mathbf{G}, \mathbf{X})$. The problem $\mathbf{BB}(\mathbf{G}, \mathbf{X})$ requires an algorithm to return some e that is in $\mathbf{F}(\mathbf{G}, \mathbf{X})$ or $\emptyset$ if there is none.

The following indicates the complexity relationships that are determined solely by $\mathbf{X}$ being a monotonic property in NP.

Lemma 3.1 *If* $\mathbf{X}$ *is a monotonic property in* NP*, then*

1. $\mathbf{U}^i(\mathbf{X}) \in$ NP *and is monotone, for fixed* i*.*
2. $\mathbf{BB}(\mathbf{G}, e, \mathbf{X}) \equiv_T^P \mathbf{X}$

[1] In fact, in those papers a backbone is defined under all assignments causing a minimum violation of clauses.

3. $\mathbf{U}(\mathbf{X}) \propto_T^P \mathbf{BB}(\mathbf{G}, \mathbf{X}) \propto_T^P \mathbf{BB}(\mathbf{G}, e, \mathbf{X})$

4. $\max_c \propto_T^P \mathbf{X}$

Proof: For 1, monotonicity is trivial. To see that it is in NP, note that we need at most a polynomial number of certificates of polynomial size. For 2 we note that using an oracle for $\mathbf{X}$ we can test an e by simply adding it to $\mathbf{G}$. On the other hand, since every $e \in \mathbf{E}(\mathbf{G})$ is in $\mathbf{F}(\mathbf{G}, \mathbf{X})$ when $\mathbf{G} \in$ not-$\mathbf{X}$, we can test for insolubility of $\mathbf{G}$ using $\mathbf{BB}(\mathbf{G}, e, \mathbf{X})$.[2] For 3 and 4 the proofs are straightforward. ■

It is not so easy to determine the complexity of $\mathbf{U}$ or of $\max_c$. In fact, for IS and HC we will show that $\mathbf{U}^i$ is NP-complete for any fixed i, while for fixed k, $\mathbf{U}^i(k\text{-COL})$ and $\mathbf{U}^i(k\text{-SAT})$ are in P for i sufficiently large.

We find that for every property $\mathbf{X}$ in the standard set listed in section 1, $\max_c(\mathbf{X}) \in$ P. The obvious conjecture however does not seem to hold, since for k-CSG and for certain composite properties, we show reductions from complete properties that indicate $\max_c$ is not in P.

Determining the boundaries between P and NP or conditions when $\max_c$ is not in P are challenging open problems.

4 The Complexity of Some Unfrozen Properties

For some problems it is easily seen that the unfrozen versions are NP-complete.

Theorem **4.1** $\mathbf{U}^i(\mathbf{IS}) \in$ NP-complete *for all* $i \geq 0$.

Proof: Given a graph $\mathbf{G}$, construct $\mathbf{G}'$ by making $i+1$ copies of $\mathbf{G}$ and inserting all possible edges between the copies. Then $\mathbf{G}' \in \mathbf{U}^i(k\text{-IS})$ iff $\mathbf{G}$ contains a k-IS. ■

For k-SAT, we have the following results.

Theorem **4.2** $\mathbf{U}(k\text{-SAT})$ *and* $\widehat{\mathbf{U}}(k\text{-SAT})$ *are* NP-complete*, for fixed* $k \geq 3$.

Proof: A reduction from NAE-k-SAT[3]. ■

Theorem **4.3** $\mathbf{U}^i(k\text{-SAT})$ *is in* P *for* $i \geq 2^k - 1$*, and* $\widehat{\mathbf{U}}^i(k\text{-SAT})$ *is in* P *for* $i \geq k$.

For other cases the analysis is less simple.

Theorem **4.4** $\mathbf{U}^i(\mathbf{HP}) \in$ NP-complete, $\forall i \geq 0$.

[2]It is worth noting that even if we redefine $\mathbf{F}$ to be empty when $\mathbf{G} \in$ not-$\mathbf{X}$, the reduction can be carried out by starting with an empty graph and adding the edges in $\mathbf{G}$ in some order, testing each in turn to see if frozen. Thus, this relation holds even if we use e.g. the spine definition of [3].

[3]Many of the proofs can be found in Beacham's thesis available at http://web.cs.ualberta.ca/~joe/publications.html.

Proof: (outline) Let $\mathbf{G}$ be any graph, and construct a new graph $\mathbf{G}' = ((i+1)\mathbf{G}) \times K_{2i}$, where there are no edges between the $i+1$ copies of $\mathbf{G}$. Then $\mathbf{G} \in \mathbf{HP} \iff \mathbf{G}' \in \mathbf{U}^i(\mathbf{HP})$. [i.e., $\iff$ $\mathbf{G}'$ contains a hamiltonian path even after first deleting i arbitrary edges.]

Suppose $\mathbf{G} \notin \mathbf{HP}$. Then covering the vertices in each copy of $\mathbf{G}$ requires at least two disjoint paths. Thus, it takes $\geq 2(i+1)$ disjoint paths to cover the $i+1$ copies of $\mathbf{G}$. To create a Hamiltonian path in $\mathbf{G}'$, we would need to connect these disjoint paths together by visiting a vertex in the clique K_{2i} in between each subpath. But this would require at least $2i+1$ vertices in the clique. Hence, $\mathbf{G} \notin \mathbf{HP} \Longrightarrow \mathbf{G}' \notin \mathbf{HP} \Longrightarrow \mathbf{G}' \notin \mathbf{U}^i(\mathbf{HP})$.

Suppose $\mathbf{G} \in \mathbf{HP}$. We must show that there is a Hamiltonian path in $\mathbf{G}'$ even if an arbitrary set of i edges is first removed. The key idea is to note that we start with $i+1$ subpaths in the copies of $\mathbf{G}$. Then, each edge deleted from a copy of $\mathbf{G}$ can create at most one new subpath by splitting a current subpath. Thus, we can create at most $2i+1$ subpaths, which can be joined by the $2i$ vertices of the clique.

There are two other classes of edges that can be removed: edges in the clique, and edges with one endpoint in the clique and one endpoint in a copy of $\mathbf{G}$. The bulk of the proof is to show that even when some of the i deleted edges are from these classes, there yet remain connections enough to complete the path. Details omitted for reasons of space. ■

Remark: Note that by adding an extra vertex to the clique in the proof, we can modify the proof to obtain a reduction $\mathbf{HP} \propto_M^P \mathbf{U}^i(\mathbf{HC})$.

Remark: Note that any Hamiltonian path in the graph $\mathbf{G}'$ constructed in the proof of theorem 4.4 must contain a Hamiltonian path of at least one of the copies of $\mathbf{G}$ as a subpath. Hence, simply finding an $\mathbf{HP}$ in a graph $\mathbf{G} \in \mathbf{U}^i(\mathbf{HP})$ is as hard as finding a Hamiltonian path in an arbitrary graph.

THEOREM **4.5** $\mathbf{U}$(3-COL) $\in$ NP-complete.

Proof: (sketch) Reduction is from NAE-3-SAT. Each literal L in the NAE-3-SAT instance is represented by the graph gadget shown in figure 1(a). The vertex $\boxed{\mathbf{3}}$ is a single distinguished vertex that is assumed to be colored 3 in any 3-coloring of the graph; all literals are attached to this same vertex. We say the literal gadget has value $\mathbf{S}$ (same) if and only if a and b are colored the same color, and value $\mathbf{D}$ (different) otherwise. The original literal L will be assigned *true* if and only if its corresponding graph representation has value $\mathbf{S}$. This encoding does not depend on the actual colors used to color the gadget. For instance, $\mathbf{S}$ can be encoded as $c(a) = c(b) = 1$ or $c(a) = c(b) = 2$.

We then construct further gadgets that enforce desired properties. For example, the graph in figure 1(b) ensures that the literals x and $\overline{x}$ do not both have value $\mathbf{D}$. Similarly, gadgets can be designed that ensure x and $\overline{x}$ do not both have value $\mathbf{S}$, and not all three literals in a clause have identical values.

Together, these gadgets ensure that the constructed graph is 3-colorable if and only if the original NAE-3-SAT instance has a satisfying assignment.

The next step of the proof shows that if the graph is colorable, then it is also in $\mathbf{U}$(3-COL). By an exhaustive case analysis, one proves the following:

1. Any gadget is still colorable (and the coloring can be extended to the rest of the graph) if any edge is added with both endpoints inside the gadget. The reader is encouraged to check that figure 1(b) has this property.

2. If any edge is added between two gadgets, the colorings on those gadgets can be adjusted to accommodate the new edge, and this coloring is extensible to the rest of the graph.

To aid in the proof, one notes the following facts:

1. Due to the symmetry of NAE-3-SAT, we can flip the values of all the literals to generate a second solution. This means that if the graph is colorable, we have at least two distinct colorings: one where $x = \mathbf{S}$, and one where $x = \mathbf{D}$. (This means that adding the edge ab in figure 1(a) cannot render the graph uncolorable).

2. If proper care is taken, the gadgets can be designed so that for every vertex v in the gadget, there exists a valid encoding of a solution to the NAE-3-SAT instance such that v has a choice of two colors *independent* of the coloring on any other gadget.

Further details are omitted due to space constraints. ■

This result can be extended to k-COL by using the graph $\mathbf{G}' = \mathbf{G} \times K_{k-3}$ where $\mathbf{G}$ is the graph constructed in the proof, and K_{k-3} is a $(k-3)$-clique.

Note that the proof construction also shows that coloring an unfrozen graph is NP-hard, since the graphs are either unfrozen or not 3-colorable.

Trivially, if $i \geq \binom{k+1}{2}$ then $\mathbf{U}^i(k\text{-COL})$ contains no graph on more than k vertices. We improve this polynomial upper bound for fixed k as follows.

Lemma 4.1 *For* $k \geq 3$, $0 \leq i < \binom{k+1}{2}$

$$\mathbf{U}^i(k\text{-COL}) \subseteq \left\{ G \,\middle|\, \forall V' \subseteq V, |V'| = k+1 \Rightarrow |E(G[V'])| < \binom{k+1}{2} - i \right\}$$

Proof: The condition is necessary to avoid the creation of $k+1$-cliques. ■

Lemma 4.2 *If* $\mathbf{G} \in \mathbf{U}^{\binom{k}{2}}(k\text{-COL})$*, then all connected components of* $\mathbf{G}$ *contain* $\leq k$ *vertices.*

Theorem **4.6** *For* $k \geq 3$ *we have*

$$\mathbf{U}^{\binom{k}{2}}(k\text{-COL}) = \{G \,|\, \forall V' \subseteq V, |V'| = k+1 \Rightarrow |E(G[V'])| \leq k-1\}$$

Proof: (sketch) Lemma 4.1 provides the forward inclusion.

Let $\mathbf{G}$ be such that $\forall V' \subseteq V$, if $|V'| = k+1$ then $|E(G[V'])| \leq k-1$. Let E' be any set of $\binom{k}{2}$ edges not already in $\mathbf{G}$, and let $\mathbf{G}' = \mathbf{G} + E'$. By way

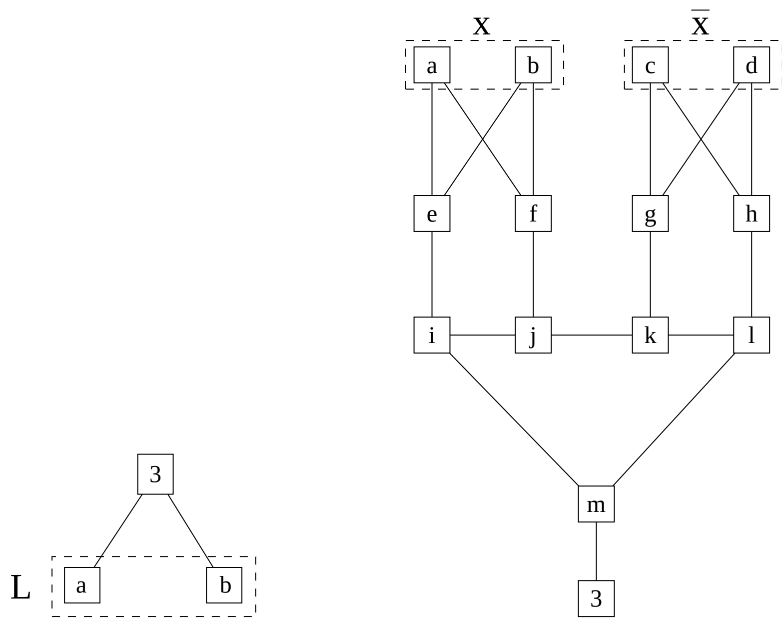

Figure 1: (a) A literal $\boldsymbol{L}$. (b) $\boldsymbol{x}$ and $\overline{\boldsymbol{x}}$ cannot both have value $\mathbf{D}$

of contradiction, assume that $\mathbf{G}'$ is uncolorable. Let $H \subseteq \mathbf{G}'$ be a minimal uncolorable subgraph of $\mathbf{G}'$. By the choice of $\mathbf{G}$, $n_H = |V(H)| > k+1$. As H is $k+1$-critical, its minimum degree is $\geq k$; hence, it must contain at least $kn_H/2$ edges. The proof is completed via induction on n_H (with help from lemma 4.2) by showing that H cannot have this many edges. ■

Theorem 4.6 tells us that $\mathbf{U}^{\binom{k}{2}}(k\text{-COL})$ may be solved in $O(n^{k+1})$ time by checking for the existence of a set of $k+1$ vertices that induce k or more edges in $\mathbf{G}$. However, since (by lemma 4.2) all the components of $\mathbf{G}$ are small, a dynamic programming algorithm can solve the problem in $O(k2^k n)$ time, which is linear for fixed k.

5 The complexity of $\max_c$

We observe that if $\mathbf{X}$ is any of the properties in the standard list in section 1 then the maximally constrained version of the property, $\max_c(\mathbf{X})$ is in the complexity class P. The proofs are easy exercises.

A quick glance through [11] shows that for many other “natural” graph properties, such as graph bipartition, the corresponding $\max_c$ property is in P. However, the obvious conjecture that $\max_c(\mathbf{X})$ is in P for any monotonic $\mathbf{X} \in$ NP seems likely to be false because of the following.

Theorem **5.1** $\max_c(k\text{-}\mathbf{CSG})$ *is isomorphism complete.*

Lemma 5.1 *Let* $\mathbf{G} = \mathbf{G}_1 \times \mathbf{G}_2$ *and* $k = |V_1| = |V_2|$. *If* $\mathbf{G} \in k\text{-}\mathbf{CSG}$ *then the* V' *such that* $\mathbf{G}[V'] \subseteq \mathbf{G}^c[(V_1 \cup V_2) \setminus V]$ *has either* $V' \subseteq V_1$ *or* $V' \subseteq V_2$.

Proof: Suppose V' is composed of m $(0 < m < k)$ vertices from V_1 and $k - m$ vertices from V_2. Then $\mathbf{G}[V']$ is a connected graph while $\mathbf{G}^c[V \setminus V']$ is disconnected, so $\mathbf{G}[V'] \not\subseteq \mathbf{G}^c[V \setminus V']$. ■

Lemma 5.2 *For* $\max_c(k\text{-}\mathbf{CSG})$, *if* $k < n/2$ *then the problem can be reduced in polynomial time to a case where* $k' = n'/2$.

Proof: If $\mathbf{G} \in \max_c(k\text{-}\mathbf{CSG})$, then there exist disjoint subsets of vertices $V', V'' \subseteq V$, $|V'| = |V''| = k$ such that $\mathbf{G}[V'] \cong \mathbf{G}^c[V'']$. Since $\mathbf{G}$ is maximal, every vertex in $V \setminus (V' \cup V'')$ must have degree $n - 1$. Thus, to check if a given graph $\mathbf{G}$ has the property $\max_c(k\text{-}\mathbf{CSG})$ where $k < n/2$, we can first remove $n - 2k$ vertices of degree $n - 1$ from $\mathbf{G}$ to reduce the problem to one where $k = n'/2$. If there are not that many vertices of large degree in $\mathbf{G}$, then we conclude $\mathbf{G} \notin \max_c(k\text{-}\mathbf{CSG})$. ■

Definition: R is the relation on the vertices V of a given graph $\mathbf{G}$ defined by $uRv \iff uv \notin E$. Set $vRv\ \forall v \in V$. The equivalence relation $\sim$ is the transitive closure of R.

Lemma 5.3 *If* $\mathbf{G}_1 \cong \mathbf{G}_2$ *and* $\mathbf{G} = \mathbf{G}_1 \times \mathbf{G}_2^c$, *then* $\exists$ *an equivalence class* $C \subseteq V$ *under* $\sim$ *such that* $\mathbf{G}[C] = \mathbf{G}_1$ *or* $\mathbf{G}[C] = \mathbf{G}_2^c$.

Proof: Clearly, if $u \in \mathbf{G}_1, v \in \mathbf{G}_2^c$, then $u \not\sim v$. Thus, for any equivalence class $\overline{u}$ of $\sim$, either $\overline{u} \subseteq V_1$ or $\overline{u} \subseteq V_2$. Therefore, the equivalence classes of $\sim$ partition both V_1 and V_2. Now, suppose $\mathbf{G}_1 \neq \mathbf{G}[\overline{u}]$ for any $u \in V$. Then $\mathbf{G}_1 = \mathbf{G}[\overline{u_1}] \times \mathbf{G}[\overline{u_2}] \times \cdots \times \mathbf{G}[\overline{u_m}]$ for some vertices $u_1, u_2, \ldots, u_m \in V_1$. But then $\mathbf{G}_2^c \cong \mathbf{G}_1^c$ is a disconnected graph, which implies $V_2 = \overline{v}$ for some $v \in V_2$. ■

Proof: (of theorem 5.1) (ISO $\propto_T^P \max_c(k\text{-}\mathbf{CSG})$) Given two graphs $\mathbf{G}_1, \mathbf{G}_2$ with $|V_1| = |V_2|$, and $|E_1| = |E_2|$, let $\mathbf{G} = \mathbf{G}_1 \times \mathbf{G}_2^c$. By lemma 5.1 $\mathbf{G} \in \max_c(k\text{-}\mathbf{CSG}) \Longrightarrow \mathbf{G}_1 \cong \mathbf{G}_2$.

($\max_c(k\text{-}\mathbf{CSG}) \propto_T^P$ ISO) By lemma 5.2, we need only consider the case where $k = n/2$. In this case $\mathbf{G} \in \max_c(k\text{-}\mathbf{CSG}) \iff \exists V' \subseteq V$ such that $\mathbf{G} = \mathbf{G}[V'] \times \mathbf{G}[V \setminus V']$ and $\mathbf{G}[V'] \cong \mathbf{G}^c[V \setminus V']$. In polytime compute the equivalence classes of $\mathbf{G}$ with respect to the relation $\sim$. By lemma 5.3, if $\mathbf{G} \in \max_c(k\text{-}\mathbf{CSG})$, then $\exists$ an equivalence class $\overline{v}$ of size $n/2$. Let $V' = \overline{v}$. Then $\mathbf{G} \in \max_c(k\text{-}\mathbf{CSG}) \iff$ every vertex in V' is adjacent to every vertex in $V \setminus V'$ and $\mathbf{G}[V'] \cong \mathbf{G}^c[V \setminus V']$, which requires the isomorphism test. ■

We also note

Theorem **5.2** $k\text{-}\mathbf{CSG} \in$ NP-complete *for* $k = n/2$.

Proof: **HC** $\propto_M^P$ k-**CSG**, details omitted. ∎

After much searching, we found the following, rather unsatisfying, NP-hard $\max_c$ problem. Let Δ be the maximum degree of a graph.

Theorem **5.3** $\max_c$(3-COL *and* $\Delta \leq 4$) *is* NP-*hard.*

Proof: 3-COL with $\Delta = 4$ is known to be NP-complete[11]. To convert an instance to our $\max_c$, simply add $K_{4,4}$'s by breaking an edge and joining the free ends to vertices of degree less than 4, until all vertices are of degree 4. 3-colorabillity is preserved, with maximality being on degree. ∎

We consider this unsatisfactory from the viewpoint that we obtain a hard maximal problem by maximizing on a polynomial property. The following theorem makes this more obvious by showing that several combinations of NP-complete properties remain polynomial when taken to the extreme.

Theorem **5.4** *For the pairs* **HC** *and* k-COL, **HC** *and* j-**IS**, k-COL *and* j-**IS**, $\max_c$ *of the pair is in* P *(for fixed* k*).* **IS** *can be replaced by* **Clique** *or* **VC**.

The $\max_c$(3-COL and $\Delta \leq 4$) problem does bear some relation to the generation of hidden solution instances and it might be worthwhile to experimentally consider hidden solution instances in which edges are added until the maximum degree bound is reached.

6 Discussion

In this paper we only look at one aspect of the frozen problem, namely the parts that are frozen out as we preserve a monotonic NP-complete property. In [7] for example, we also note that pairs of vertices may be frozen under coloring in the sense that the vertices always receive different colors under any coloring, although they are not an edge in the graph. We call these freely addable parts.

There is also the issue of preserving not-**X** under the deletion of parts. Again parts may be frozen or freely deletable. Corresponding to **U** we have **D** meaning any part may be deleted. Corresponding to $\max_c$ we have $\min_c$, which in k-coloring is a $k+1$-critical graph. Unlike $\max_c$, $\min_c$ seems almost always to be hard.

It is usually easy to identify frozen parts for either direction and freely addable parts using a program for testing **X**. But we know of no way to do this for freely deletable parts (e.g. edges not in any critical subgraph). These observations no doubt correlate to the fact that CO-NP appears harder to deal with in practice than NP. (We are at the edge of the polynomial hierarchy).

One very challenging problem is to determine for k-COL the polynomial unfrozen boundary; that is, for which $i, 2 \leq i < \binom{k}{2}$ is $\mathbf{U}^i$(k-COL) in P. For 3-COL we note that $\mathbf{U}^3$(3-COL) is in P and

Conjecture **6.1** $\mathbf{U}^2$(3-COL) $\in$ NP-complete.

We are fairly well convinced that this conjecture is correct through a reduction sequence 3-COL $\propto_M^P \widehat{\mathbf{U}}^2(\mathbf{H3K2}) \propto_M^P \mathbf{U}^2$(3-COL). The reduction on the left is a relatively straightforward theorem. For the one on the right, we start with the graph of Descartes [8]. From this we delete 3 judiciously chosen edges, then duplicate this structure, and pairwise merge one of each corresponding pair of vertices of the deleted edges, and join each other pair by an edge. This device then implements one hyperedge of the $\widehat{\mathbf{U}}^2(\mathbf{H3K2})$ instance. The difficulty comes in enumerating all the cases for adding two edges to the completed graph. We note that each edge of the original 3-COL instance is replaced on the right by a subgraph with more than 10 million vertices. Symmetry helps, but at the time of this submission not all cases have been listed and verified.

For the greater challenge we offer the less well supported

CONJECTURE **6.2** $\mathbf{U}^{\binom{k}{2}-1}$($k$-COL) $\in$ NP-complete*, for fixed k.*

In a graph in this class, all components of more than $k+1$ vertices must have girth at least $k+2$. Small components lead to polynomial time checking algorithms for fixed k, (roughly $O(N^{k^2}k^{k^3})$) and so we will need graphs of large girth. (This was the reason for starting with the Descartes graph above). Following the pattern of the previous construction, we might begin with large girth $k+1$-critical graphs, say a subgraph of those in [12], and construct unfrozen (to the $k-1$th) hypergraph 2-color instances. This approach is not likely to yield short proofs. We note furthermore that any many-one reduction will likely involve large girth k-chromatic graphs, among other properties. The difficulty is that the best known constructions of such graphs are not even primitive recursive[12].

It also remains open for which $i, 1 < i < 2^k - 1$, $\mathbf{U}^i$(k-SAT) becomes polynomial. Assuming P $\neq$ NP, is the switch directly between P and NP-complete, or are there intermediate problems for some values of i?

References

[1] D. Achlioptas, C. Gomes, H. Kautz, and B. Selman. Generating Satisfiable Problem Instances In *Proceedings of the Seventeenth National Conference on Articial Intelligence (AAAI-2000)*, pages 256–261, 2000.

[2] B. Bollobás. Modern Graph Theory. *Graduate Texts in Mathematics #184.* Springer Verlag New York. 1998.

[3] B. Bollobás, C. Borgs, J.T. Chayes, J.H. Kim, and D.B. Wilson. The scaling window of the 2-SAT transition, 1999. Paper in the xxx.lanl.gov e-Print archive. Available from http://xxx.lanl.gov/abs/math.CO/9909031.

[4] P. Cheeseman, B. Kanefsky, and W.M. Taylor. Where the really hard problems are. In *Proceedings of the 12th IJCAI*, pages 331–337, 1991.

[5] J.M. Crawford and L.D. Auton Experimental Results on the Crossover Point in Random 3SAT Artificial Intelligence 81 (1996)

[6] J.C. Culberson and F. Luo. Exploring the k–colorable landscape with iterated greedy. In David S. Johnson and Michael A. Trick, editors, *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, 1993*, volume 26 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 245–284. AMS, 1996.

[7] J. Culberson and I. P. Gent. Frozen Development in Graph Coloring. Theoretical Computer Science (to appear). Available as APES Research Report, APES-19-2000. http://www.apes.cs.strath.ac.uk/apesreports.html

[8] B. Descartes. Solution to Advanced Problem 4526, proposed by Peter Ungar. The American Mathematical Monthly, 61 (1954), pp. 352-353.

[9] E. Friedgut and G. Kalai. Every Monotone Graph Property Has a Sharp Threshold. Proc. Amer. Math. Soc. 124 (1996), pp. 2993-3002.

[10] Y. Gao and J. Culberson. The Phase Transition in NK Landscapes is easy. Genetic and Evolutionary Computation Conference, 2001. (to appear)

[11] M.R. Garey and D.S. Johnson. Computers and Intractibility: A Guide to the Theory of NP-Completeness W. H. Freeman and Company, 1979.

[12] I. Kříž. A hypergraph-free construction of highly chromatic graphs without short cycles. *Combinatorica*, 9(2):227–229, 1989.

[13] D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. In *Proceedings of AAAI-92*, pages 459–465, 1992.

[14] R. Monasson and R. Zecchina. Statistical mechanics of the random k-sat model. *Phys. Rev. E*, 56, 1997.

[15] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L Troyansky. Determining computational complexity from characteristic 'phase transitions'. *Nature*, 400:133–137, 1998. http://www.lpt.ens.fr/∼monasson/ http://www.ictp.trieste.it/∼zecchina.

[16] B. Selman, D. Mitchell, and H. Levesque Generating Hard Satisfiability Problems Artificial Intelligence 81 (1996), pp. 17-29

[17] B. Vandegriend and J. Culberson. The G_{nm} phase transition is not hard for the Hamiltonian cycle problem. *Journal of Artificial Intelligence Research*, 9:219–245, 1998. www.jair.org/contents/v9.html.