UNIVERSITY OF CALIFORNIA
SANTA CRUZ

AUTOMATED RESONANCE ASSIGNMENT VIA BOOLEAN SATISFIABILITY AND QUADRATIC PROGRAMMING

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Master of Science

in

COMPUTER SCIENCE

by

Benjamin C. Sherman

June 2019

The Dissertation of Benjamin C. Sherman
is approved:

Seshadhri Comandur, Chair

Luca DeAlfaro

Phokion Kolaitis

Lori Kletzer
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

AUTOMATED RESONANCE ASSIGNMENT VIA BOOLEAN SATISFIABILITY AND QUADRATIC PROGRAMMING

by

Benjamin C. Sherman

The community of researchers using nuclear magnetic resonance (NMR) spectroscopy to study the structure and dynamics of proteins are interested in solving the following problem. Let G and H be undirected graphs such that H is isomorphic to at least one subgraph of G. For each vertex $v \in V(H)$, compute the set of vertices to which v is mapped by a subgraph isomorphism from H to a subgraph of G. This thesis introduces variations of this problem, reviews prior work, and proposes algorithmic and heuristic methods that outperform the incumbent state of the art on published benchmarks.

To Zayd

Acknowledgments

This thesis is joint work with Professors Dimitris Achlioptas and Nikolaos Sgourakis. This research started as a conversation between Nik and Dimitris. Nik is a chemist who was familiar with prior work on this problem and Dimitris is a computer scientist. I got involved when Dimitris asked me to come up with an efficient SAT encoding for subgraph isomorphism. At that time, Nik and Dimitris were trying to solve the maximum-likelihood problem discussed in Chapter 3 using stochastic local search to no avail. Dimitris had the idea to use SAT, punted it to me, and here we are. All novel algorithms in this thesis are due to Dimitris and I.

I want to thank everyone else in the Sgourakis lab who helped me pretend to be a chemist for 3 years, especially Andrew McShan, Sarah Overall, Danai Moschidi, Vivianne de Paula, David Flores-Solis, and Santrupti Nerli. I would like to thank Gwen Lin, my high school counselor, for accidentally putting me into AP computer science my senior year and refusing to let me drop it because all the chemistry classes were full. I would like to thank Michael Ferraro for teaching that AP computer science class beautifully. Without Dylan Anderson I could not have survived middle and high school. Without Anissa Zaitsu I could not have survived college. Without my parents and my brothers, I would not be who I am (for better or worse). Thank you all for helping me get here.

Chapter 1

Introduction

Nuclear magnetic resonance spectroscopy (NMR) is a methodology for measuring magnetic properties of nuclei. NMR is widely used in structural biology to identify the three-dimensional structure and dynamics of proteins by measuring the magnetic resonances of nuclei [11]. Roughly speaking, protein NMR experiments consist of placing a protein in a strong magnetic field and measuring the magnetic resonance of the entire protein as a decaying waveform. This waveform can then be transformed into a frequency spectrum, which will contain peaks at the resonant frequency of distinct nuclei. In this way, NMR is capable of easily identifying the set of nuclear resonances in a protein, but before these data can be used to draw conclusions about a protein, the observed resonances must be attributed to specific nuclei. The difficulty of this task, which is called the resonance assignment problem, is arguably the principal limitation of NMR in protein studies. The difficulty of the problem is suggested by the mere fact that a protein may contain hundreds or thousands of nuclei and, naively, there are $n!$ explanations for the generation of n resonances from n nuclei, and it is not obvious that one can confidently discriminate among assignments.

One of the main considerations for the design of experimental protocols in protein NMR is the ease of the resulting resonance assignment problem. Recently, there has been a shift within the protein NMR community towards a strategy known as methyl labelling. In a methyl-labeled NMR experiment, only the nuclei residing in molecular compounds called methyls can make a non-negligible contribution to the NMR spectra. This serves to significantly de-noise the spectra and decrease the number of nuclei involved in the assignment problem while still measuring enough nuclear resonances to be valuable. This thesis is concerned with the automation of the so-called methyl assignment problem, i.e., the problem of assigning the resonances measured in a methyl-labeled NMR experiment to the methyl-nuclei of a protein.

In addition to the set of resonances, there are other types of information available to aid in the methyl assignment process. A rough model, or template structure of the protein in question is typically available, allowing for the identification of the set of methyl-nuclei that may contribute to the spectra. This can be obtained via X-ray crystallography, wherein a protein is studied in a crystallized form. Additionally, the methyls of a protein may be partitioned according to the type of residue in which they appear, which is typically either alanine (ALA), isoleucine (ILE), leucine (LEU), or valine (VAL). Moreover, the range in which a nucleus resonates is determined by the type of residue in which it resides. Therefore we can restrict the assignment of resonances in certain regions of the spectrum to methyls residing in a specific type of residue. Finally, NMR can measure the transfer of magnetization between nuclei, a phenomenon known as a nuclear Overhauser effect (NOE). An NOE is only possible when the nuclei are

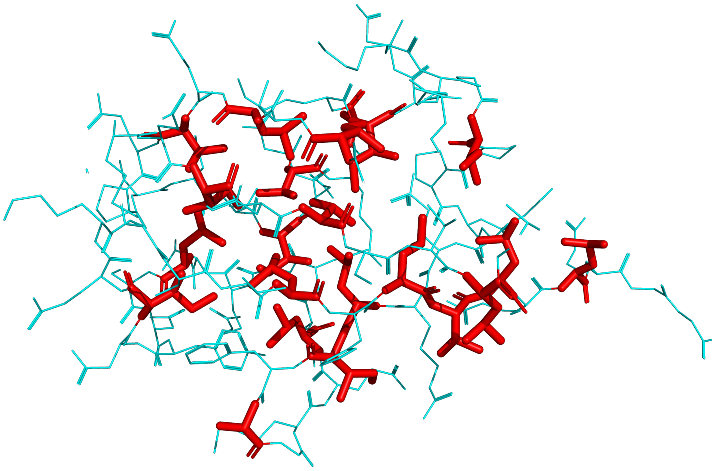

Figure 1.1: X-Ray structure of ubiquitin, used as the template structure for assigning the resonances of the same molecule. Methyls are shown in red.

close in space, so NOE data allow us to infer that specific pairs of methyl-resonances must have been generated by a pair of methyls that are close to each other in our template structure. The methyl assignment is now constrained as follows: each resonance should be assigned to a methyl residing in a residue that is consistent with the position of that resonance in the spectrum, and pairs of resonances exhibiting an NOE should be assigned to methyls that are near each other in the template structure.

In order to fully appreciate the difficulty of the methyl assignment problem, it is necessary to adopt a computational perspective, which arises naturally from the following graph theoretic description. Let the graph G represent a crude estimation of the protein's structure; the vertices of G are the methyls in the protein and the edges of G connect any two methyl-vertices that are close enough to exhibit an NOE during the experiment. Let the graph H represent a model of the experimental data; each H-vertex is a spectral peak and there is an edge between every pair of spectral peaks between

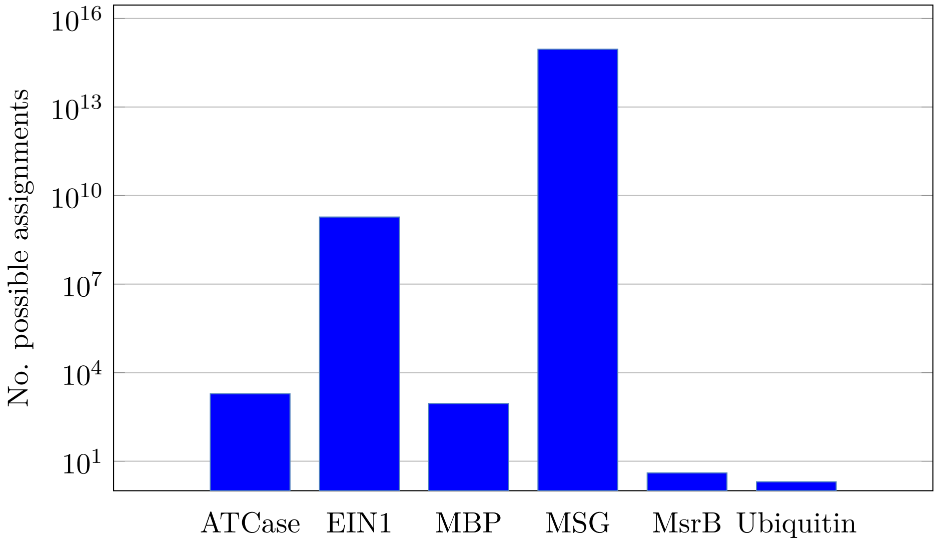

Figure 1.2: Number of possible methyl assignments for each of the molecules in the dataset outline in Table 5.1. Values were obtained by invocation of `sharpsat` [21], a tool that counts satisfying assignments to CNF formulas.

which there was an NOE observed. This description of the problem implies the following: the correct assignment of peaks to methyls corresponds to an isomorphism between H and the subgraph of G that arises by deleting an edge between any two methyls in G between which no NOE is experienced. Unfortunately, the problem of determining whether G contains a subgraph isomorphic to H is NP-complete [7], suggesting that such isomorphisms can not be found efficiently. Even worse, in difficult instances of methyl assignment, there are far too many subgraphs of G that are isomorphic to H to enumerate, even after restricting vertex assignments using residue-type information (see Figure 1.2). At this point, it may seem that, even given a computational engine capable of detecting subgraph isomorphisms, the methyl assignment problem is dramatically under-constrained.

1.1 Previous Work

In practice, most resonance assignment problems are solved manually. The typically process is to guess an assignment for a single resonance, consider the immediate consequences of that guess, and then make another guess. If it becomes clear that a mistake has been made, start to backtrack. In some sense, postdoctoral researchers are performing a version of the David-Putnam procedure by hand, a process which typically takes a matter of weeks to months. The way practitioners of methyl NMR have made this work is with the aid of secondary experiments, most notable an experiment known as mutagenisis [19, 20]. Mutagenisis in this case is the process of mutating a specific methyl out of the protein in study, then rerecording the NMR spectra to observe which peak has vanished, thus discovering the assignment of a single resonance. After performing some number of mutagenisis experiments, a human will be able to easily determine the assignment for an H-vertex given the assignments in its neighborhood. The problem with mutagenisis is that each vertex assignment purchased requires the preparation of another sample.

Early work on automation of the methyl assignment problem did not fit exactly within the graphical paradigm used herein. The earliest notable tool for automating the methyl assignment was `MAPXS` [27]. `MAPXS` was the first of many variations on the following monte carlo method: initialize a random assignment of methyls to resonances, and swap pairs of assignments to improve the score of your assignment according some objective function before terminating arbitrarily. Later modifications of this include `MAPXS-II` [28], `FLAMEnGO` [1], and `FLAMEnGO2.0` [2], with each development essentially constituting a new choice of objective function. These methods do, in one way or another, penalize assignments which do not correspond to a subgraph isomorphism, but none enforce subgraph isomorphism as a requirement for the final assignment. The tool MAGIC [13] represents a development from prior works in that it successfully optimizes its objective function in all its benchmarks.

The first method to operate in the graphical paradigm is MAGMA [14]. MAGMA defines the graphs H and G as in this work, but asserts that the correct assignment must correspond to a maximum-common-edge subgraph (MCES) between the graphs rather than a subgraph isomorphism. The significance of this distinction is that it accounts for the case where no G-subgraph is isomorphic to H, which is possible only in the event of an error in the construction of H. As it turns out, such errors are both common and very difficult to detect. MAGMA automates the problem by computing the set of G-vertices to which each individual H-vertex is assigned by any MCES. The fact that such sets may be computed in a reasonable amount of time should be surprising, since there are a vast number of MCESes, and the computation of any one is NP-hard.

1.2 The Contribution of This Thesis

The contribution of this thesis is a SAT-based method which computes the same sets as MAGMA, i.e., the possible assignments for each vertex over all MCESes. The issue of erroneous edges will be overcome with the use of MARCO [10], a tool that can compute minimal sets of constraints whose removal from an unsatisfiable formula causes the formula to become satisfiable. The method presented will be shown to terminate much faster MAGMA on the most difficult benchmarks. Moreover, this thesis presents a method for exact optimization over the set of MCESes using integer-quadratic programming, where the definition of optimality is directly motivated by the physics of the NMR experiment. The list of possible assignments may be thereby restricted to assignments occuring in a near-optimal MCES. This leads to a boost in the number of uniquely assigned vertices in most benchmarks, without losing the correct assignment for a single vertex in any of the benchmarks.

Chapter 2

Finding Methyl Assignments

2.1 Preliminary Definitions

The foundation of this work is a graph theoretic description of the methyl assignment problem. While the basic concept of a graph is assumed, a few definitions will be stated in the interest of establishing consistent notation.

Definition 1. *For any graph $\mathcal{G}$, let $V(\mathcal{G})$ denote the vertices of $\mathcal{G}$, and let $E(\mathcal{G})$ denote the edges of $\mathcal{G}$.*

Definition 2. *The neighborhood of a vertex v in a graph $\mathcal{G}$ is defined as*

$$N_{\mathcal{G}}(v) := \{u \in V(\mathcal{G}) \mid \{u, v\} \in E(\mathcal{G})\}$$

Definition 3. *The degree of a vertex v in a graph $\mathcal{G}$ is defined as*

$$d_{\mathcal{G}}(v) := |N_{\mathcal{G}}(v)|$$

Definition 4. *Let G be a graph such that $V(G)$ is the set of methyls in a protein and $E(G)$ is the set of pairs of methyls in $V(G)$ that are close enough to experience a NOE.*

Definition 5. *Let H be a graph such that $V(H)$ is the set of methyl resonances in an NMR spectra and $E(H)$ is the set of pairs of resonances in $V(H)$ that have experienced a NOE.*

Definition 6. *Let $\gamma : V(H) \cup V(G) \rightarrow \{ALA, ILE, LEU, VAL\}$ be a function which maps each methyl to the type of residue in which it resides, and maps each resonance to the spectral residue who's spectral region it lies within.*

The object of desire in the methyl assignment problem is the correct methyl assignment, i.e., a function $\pi^* : V(H) \rightarrowtail V(G)$ such that $\pi^*(h) = g$ if and only if h is the magnetic resonance of the nucleus in the methyl g. Unfortunately, it seems likely that there is no algorithm for determining π^*. The definition of the problem allows for the rejection of assignments, but not for discrimination between assignments which cannot be rejected. The goal of the method can thus be described as the determination of $\pi * (h)$ for as many $h \in V(H)$ as possible, using a mixture of algorithmic and heuristic methods.

Definition 7. *A one-to-one function $\pi : V(H) \rightarrowtail V(G)$ is said to be a methyl assignment if and only if*

$$\forall h \in V(H) : \gamma(h) = \gamma(\pi(h))$$

Definition 8. *A methyl assignment π is said to be a valid if and only if*

$$\forall \{i, j\} \in E(H) : \{\pi(i), \pi(j)\} \in E(G)$$

The principal concern of this chapter is the means for the discovery of valid methyl assignments. This requires a formal introduction of the Boolean satisfiability problem (SAT), since modern SAT solvers are the computational workhorse of this method. Again, basic definitions will be restated in the interest of establishing consistent notation.

Definition 9. *A Boolean formula in conjunctive normal form (CNF formula) is defined as follows*

1. *A Boolean variable can be any symbol.*

2. *The negation of a Boolean variable X is the syntactic expression $\neg X$*

3. *A literal is either a Boolean variable or a negation of a Boolean variable.*

4. *A clause is a set of literals.*

5. *A CNF formula is a set of clauses.*

Definition 10. *A CNF formula F over the Boolean variables B is said to be satisfiable if and only if there exists a satisfying assignment $A : B \to \{True,\ False\}$ such that every clause clause of B contains a variable x such that $A(x) = True$ or a negation $\neg x$ such that $A(x) = False$.*

2.2 Using SAT for Methyl Assignment

The problem of determining whether a CNF formula is satisfiable is among the most studied problems in modern mathematics, serving as the canonical NP-complete problem [7]. A problem being NP-complete typically suggests that large instances cannot be solved efficiently. However, modern SAT solvers based on the conflict-driven-clause-learning (CDCL) algorithm [17] are capable of deciding the satisfiability of CNF formulas with millions of clauses in a reasonable amount of time [5].

In light of the emergence of practically effective SAT solvers, encoding instances of NP-complete problems as CNF formulas and then translating the output of a SAT solver to derive a solution to the original problem has become a popular strategy. SAT is widely used as an encoding system for hardware verification [3], operations research [6], and automated theorem proving [12]. In all such applications, a CNF formula is defined such that the satisfiability of the formula is equivalent to some statement about the original problem.

In this work, a CNF formula F will be used to encode the methyl assignment problem in a way such that F is satisfiable if and only if a valid methyl assignment exists. Moreover, if F is satisfiable, every satisfying assignment to F will correspond to a distinct valid methyl assignment. After encoding the problem in this way, valid methyl assignments can be obtained by a quick invocation of a SAT solver. The main influence of the encoding used herein is [23], which gives a SAT encoding for the Hamiltonian cycle problem. The Hamiltonian cycle problem, like subgraph isomorphism, is NP-complete. A few modifications to the ideas presented in [23] yield an effective encoding for subgraph isomorphism, and thereby an effective encoding for valid methyl assignments.

Definition 11. *Let X_{hg} be a Boolean variable informally denoting the proposition that vertex $h \in V(H)$ is assigned to vertex $g \in V(G)$.*

The methyl assignment encoding will be written in terms of the Boolean variables X_{hg}. Clauses will be added to the encoding formula to enforce that the following is true of any satisfying assignment A to the X_{hg} variables:

1. $\forall h \in V(H) : \big|\{X_{hg} \mid A(X_{hg} = \text{True}\}\big| = 1$
2. $\forall g \in V(G) : \big|\{X_{hg} \mid A(X_{hg} = \text{True}\}\big| \leq 1$
3. $\forall h \in V(G), \forall g \in V(G) : \gamma(h) \neq \gamma(g) \implies A(X_{hg}) = \text{False}$
4. $\forall h, h' \in E(H) : A(X_{hg}) \wedge A(X_{h'g'}) \implies \{g, g'\} \in E(G)$

Constraint 1 enforces that every resonance receives exactly one assignment, and constraint 2 enforces that no methyl has more than one resonance assigned to it; 1 can be restated as saying that each resonance receives at least one one assignment and at most one assignment. To force at least one X_{hg} to be true, simply create a clause containing all X_{hg} variables for a specific h or g. The at-most-one constraint is much trickier. The naive method for encoding the at-most-one constraint for a set of literals X is to add a clause $\{\neg x, \neg y\}$ for all $\{x, y\} \subseteq X$. This naive encoding results in $O(|V(G)|^3)$ clauses being added to the formula.

A more efficient encoding of the at-most-one-constraint is the commander encoding [8]. The commander encoding entails partitioning X into several sets, and creating a Boolean variable called the commander for each set. Clauses are then added such that each commander variable is True if and only if exactly one of the variables in the set it commands is true. Then an at-most-one-constraint is recursively added to the set of commander variables using the commander encoding. The base case for this recursion is that there are few enough variables that performing the naive at-most-one encoding is relatively cheap. At the cost of introducing $O(|V(G)|)$ commander variables, the commander encoding reduces the number of clauses introduced to enforce 1 and 2 from $O(|V(G)|^3)$ to $O(|V(G)|^2)$, and dramatically improves the performance of a SAT solver in deciding the satisfiability of methyl assignment formulas. Pseudocode for the commander encoding is given in Algorithm 1.

Constraint 3 can be enforced implicitly by simply not instantiating the variable X_{hg} where $\gamma(h) \neq \gamma(g)$. Constraint 4 can be enforced with the addition of $|E(H)|O(|V(G)|^2)$ clauses.

A few comments on Algorithm 2 are due. Observe that the variables X_{hg} are only instantiated and used for g in the set D_h, defined on line 3. The set $D_h \subseteq V(G)$ is defined so that D_h is guaranteed to contain every methyl g such that h is assigned to g by a valid methyl assignment. The two filters used to extract D_h from $V(G)$ are the residue type information and vertex degree. Both optimizations cause a dramatic speedup in the construction of formulas and the performance of the solver.

Definition 12. *Let F be the formula returned by Algorithm 2, given the graphs H and G.*

Theorem 1. *F is satisfiable if and only if there exists a valid assignment π. Moreover, every satisfying assigning to F corresponds to a unique valid methyl assignment π.*

Proof. It will be shown that one can construct a valid assignment π from a satisfying assignment A to the formula, and vice versa.

(I) Given a satisfying assignment A to F, define a methyl assignment π so that $\pi(h) =$

Algorithm 1 Commander variable encoding

```
function COMMANDER($L$)    ▷ $L$ is a set of literals
    $F \leftarrow \{\}$
    $L_1, L_2, \ldots, L_k \leftarrow$ partition $L$ into sets of size $\leq 3$
    $c_1, c_2, \ldots, c_k \leftarrow$ create $k$ Boolean variables
    for $i = 1, 2, \ldots, k$ do
        $F \leftarrow F \cup \{\{\neg c_i\} \cup L_i\}$    ▷ Commander implies group
        for $l \in L_i$ do
            $F \leftarrow F \cup \{\{\neg l, c_i\}\}$    ▷ Every literal implies commander
        end for
        for $x, y \in L_i$ do
            $F \leftarrow F \cup \{\{\neg x, \neg y\}\}$    ▷ At most one in group
        end for
    end for
    if $k \leq 3$ then
        for $c_i, c_j \in \{c_1, c_2, \ldots, c_k\}$ do
            $F \leftarrow F \cup \{\{\neg c_i, \neg c_j\}\}$
        end for
    else
        $F \leftarrow F \cup$ COMMANDER($\{c_i\}_{i=1}^{k}$)
    end if
    return $F$
end function
```

Algorithm 2 Construction of a SAT Formula

```
$F \leftarrow \{\}$
for $h \in V(H)$ do
    $D_h \leftarrow \{g \in V(G) \mid \gamma(h, g) \wedge d_H(h) \leq d_G(g)\}$
    $F \leftarrow F \cup \textsc{Commander}(\{X_{hg} \mid g \in D_h\})$
    $F \leftarrow F \cup \{\{X_{hg} \mid g \in D_h\}\}$
end for
for $g \in V(G)$ do
    $F \leftarrow F \cup \textsc{Commander}(\{X_{hg} \mid D_h \ni g\})$
end for
for $i \in V(H)$ do
    for $j \in D_i$ do
        for $i' \in N_H(i)$ do
            $C \leftarrow \{\neg X_{hg}\} \cup \{X_{i'j'} \mid j' \in N_G(j) \cap D_{i'}\}$
            $F \leftarrow F \cup \{C\}$
        end for
    end for
end for
return $F$
```

g if and only if $A(X_{hg}) =$ True. Assuming the correctness of Algorithm 1, it is clear that π is a one to one function respecting methyl type constraints. Therefore π is not a valid methyl assignment only if there exists an edge $\{h, h'\} \in E(H)$ such that $\{\pi(h), \pi(h')\} \notin E(G)$. Assume this is true. Per line 13 of Algorithm 2, F contains a clause

$$C = \{\neg X_{h,\pi(h)}\} \cup \{X_{h'g'} \mid g' \in N_G(\pi(h)) \cap D_{h'}\}$$

Because h' is assigned to a vertex not in $N_G(\pi(h))$, C is unsatisfied. Therefore F is not satisfied by A, which is a contradiction. Therefore π is a valid methyl assignment.

(II) Given a valid methyl assignment π, define A so that $A(X_{hg}) =$ True if and only if $\pi(i) = j$. The assignment made by A to the variables introduced by the commander encoding may be safely ignored, as these will have a value implied by the assignment to the X_{hg} variables. By the definition of a valid methyl assignment, it is clear that, conditioned on the correctness of Algorithm 1, the clauses introduced to force exactly one X_{hg} to be true for each h at most one to be true for each g will be satisfied by A. Therefore A does not satisfy F only if one or more clauses of the form

$$C = \{\neg X_{hg}\} \cup \{X_{h'g'} \mid g' \in N_G(g) \cap D_{h'}\}$$

is not satisfied. Every such clause such that $\pi(h) \neq g$ is satisfied because $A(X_{hg})$

= False. If $\pi(h) = g$, then for the clause to not be satisfied by A it must be that $\pi(h') \notin N_G(g)$, in which case π is not a valid methyl assignment, which is a contradiction. Therefore A satisfies F.

□

From this point on, the notion of satisfying assignments will largely be ignored. Instead, the output of a SAT solver will be either a valid methyl assignment π, or UNSAT if the formula is unsatisfiable.

Chapter 3

Navigating Methyl Assignments

The SAT solver has been established as a search engine for valid methyl assignments. When the solver returns a valid methyl assignment π, one may like to conclude that $\pi = \pi^*$, i.e., that for each vertex $h \in V(H)$, $\pi(h)$ is the correct assignment to h. The fact that π is a valid methyl assignment says only that π is consistent with the data. This does not mean that π is the only valid methyl assignment.

Definition 13. *For every $h \in V(H)$, let the set $S(h)$ be defined as*

$$S(h) = \{\pi(h) \mid \pi \textit{ is a valid methyl assignment}\}$$

A key discovery of `MAGMA` [14] is that $|S(h)|$ is typically 1. In other words, one does not need to introduce any heuristic in order to determine a provably correct, unique assignment for most vertices of H. The details of how `MAGMA` computes $S(h)$ are unknown, a SAT solver can be used to compute these same sets in the following way. Consider a single Boolean variable X_{hj}. If $F \cup \{\{X_{hg}\}\}$ is unsatisfiable, it can be concluded that there does not exist a valid methyl assignment $\pi : \pi(h) = g$. By naively iterating over all X_{hg} variables and testing the satisfiability of $F \cup \{\{X_{hg}\}\}$, the set of possible assignments for every individual vertex can be determined $O(|V(G)|^2)$ invocations of the solver.

3.1 Enumerating Possible Assignments

Algorithm 3 computes the sets $S(h)$ for every vertex of H through iterative use of a SAT solver. The idea of the algorithm is to progressively build the sets $S(h)$ as more satisfying assignments are witnessed. At every step, the algorithm selects a vertex h at random and adds temporary clauses to the formula to forbid h from being assigned to any vertex to which it has been assigned previously. If the solver concludes that this formula is unsatisfiable, then we can conclude that every possible assignment of h has been witnessed and recorded. Otherwise, the solver returns a new satisfying assignment which will witness new assignments for at least one vertex.

Though SAT solvers are good at finding valid methyl assignments, it is the ability of a solver to conclusively discredit theories about π^* rather than its ability to produce candidates for π^* which make it truly valuable in this setting. Observe that it is the capacity of the solver to prove the unsatisfiability of a formula which allows for the conclusion that every assignment of a vertex has been witnessed, which truly demonstrates the power of the SAT solver: a person may be able to find a single valid methyl assignment in a matter of hours if they are lucky, but if a person was asked to prove that there does not exist any valid methyl assignment in which h is mapped to g, they would be utterly hopeless.

Algorithm 3 Enumeration of Possible Assignments

```
1:  $F \leftarrow$ Invoke Algorithm 2
2:  $U \leftarrow V(H)$
3:  for $h' \in V(H)$ do
4:      $S(h) \leftarrow \{\}$
5:  end for
6:  while $U \neq \emptyset$ do
7:      $h \leftarrow$ random selection from $U$
8:      $T \leftarrow \{\{\neg X_{hg}\} \mid g \in S(h)\}$          ▷ Forbid previous assignments
9:      $\pi \leftarrow$ SOLVE$(F \cup T)$          ▷ Get methyl assignment from solver
10:     if $\pi = UNSAT$ then          ▷ $S(h)$ is complete
11:         $U \leftarrow U - \{h\}$
12:         $F \leftarrow F \cup \{\{X_{hg} \mid g \in S(h)\}\}$          ▷ Help the solver
13:     else
14:         for $u \in U$ do
15:             $S(u) \leftarrow S(u) \cup \{\pi(u)\}$          ▷ Update all sets $S(h)$
16:         end for
17:     end if
18: end while
19: return $S$
```

Line 12 of Algorithm 3 is an optimization. A clause is added to F which enforces that one X_{hg} variable is true for $g \in S(h)$. Because every literal X_{hg} such that $g \notin S(h)$ is by definition falsified by every satisfying assignment, these clauses do not affect the set of satisfying assignments; they merely help the solver reach its subsequent conclusions faster.

3.2 NOE Probability

The physics of a methyl NMR experiment actually suggest that some valid methyl assignments more likely than others. While the mere definition of a valid methyl assignment suffices to force a unique assignment for most resonances, there are still resonances with more than one possible assignment. A reliable way to discriminate among these assignments can be obtained by using the physics of the experiment to formulate a maximum likelihood problem.

The possibility of a credible maximum-likelihood approach comes from the following fact: the probability of two nuclei experiencing an NOE is a decreasing function of the distance between those nuclei. In other words, edges of H of more likely to appear between a pair of resonances that were generated by methyls close in space. Roughly, the probability of an NOE being observed between two methyls m_1, m_2 can be defined as

$$p(m_1, m_2) \propto \Delta(m_1, m_2)^{-6} \tag{3.1}$$

where $\Delta(\cdot, \cdot)$ is specifically defined to measure the distance between methyl nuclei. Though it is not strictly true, assume that H is generated in the following way. Let $V(H)$ be the set of methyl resonances. The edges of H are formed by the following sequence of independent trials: for every pair of vertices $h, h' \in V(H)$, place and edge between h, h' with probability $\propto \Delta(\pi^*(h), \pi^*(h'))^{-6}$. Under this generative model, the following statement becomes true:

$$\Pr[H = \mathcal{H}] \propto \prod_{\{u,v\} \in E(\mathcal{H})} \Delta(\pi^*(u), \pi^*(v))^{-6} \tag{3.2}$$

Though this generative model is not physically accurate, it is an essential piece of every heuristic approach to methyl assignment [27, 28, 1, 2], and its reliability will be demonstrated in the experimental section of this work. This generative model suggests the following maximum-likelihood problem, which defines π_{opt}, the optimal methyl assignment.

$$\pi_{opt} = \arg\min_{\pi} \prod_{\{u,v\} \in E(H)} \Delta(\pi(u), \pi(v))^{-6} \tag{3.3}$$

$$= \arg\min_{\pi} \sum_{\{u,v\} \in E(H)} \log \left[\Delta(\pi(u), \pi(v))^{-6} \right] \tag{3.4}$$

3.3 Quadratic Programming

The maximum likelihood problem can be formulated as a integer-quadratic program (IQP). In fact, the problem can be written as an integer-linear program (ILP), which should in general be easier to solve than the quadratic counterpart. However, the complexity of the linear encoding appears to make the ILP more difficult to solve than the IQP for an industry grade solver.

Definition 14. *Let the weight of* $\{x, y\} \in E(G)$ *be defined as*

$$w(\{x, y\}) := \Delta(x, y)^{-6}$$

Definition 15. *Let* Y_{hg} *be a variable taking values in* $\{0, 1\}$*, informally denoting the proposition that* $h \in V(H)$ *is assigned to* $g \in V(G)$*.*

Definition 16. *Let* Z_{uv} *be a variable taking values in* $\{0, 1\}$*, informally denoting the proposition that* $u \in E(H)$ *is assigned to* $v \in E(G)$*.*

Definition 17. *An assignment* $A : \{Y_{hg}\} \to \{0, 1\}$ *is a feasible solution to the methyl assignment IQP if and only if all of the following are true:*

1. $\forall h \in V(H) : \sum_{g \in S(h)} A(Y_{hg}) = 1$
2. $\forall h \in V(H) : \sum_{g \notin S(h)} A(Y_{hg}) = 0$
3. $\forall g \in V(G) : \sum_{h : S(h) \ni g} A(Y_{hg}) \leq 1$
4. $\forall \{i, j\} \in E(H), \forall \{i', j'\} \in E(G) :$

$$A(Z_{\{i,j\}\{i',j'\}}) = A(Y_{ii'})A(Y_{jj'}) + A(Y_{ij'})A(Y_{ij'})$$

5. $\forall e \in E(H) : \sum_{e' \in E(G)} Z_{ee'} = 1$

Definition 18. *The cost of a feasible solution* A *to the methyl assignment IQP is defined as*

$$c(A) := \sum_{e \in E(H)} \sum_{e' \in E(G)} \log(w(e')) \times Z_{ee'}$$

$c(\pi)$ will be used to refer to $c(A)$ where A is the feasible solution corresponding to the valid methyl assignment π.

It is clear that, in the same way as for the SAT encoding, every feasible solution to the methyl assignment IQP corresponds to a unique valid methyl assignment. At this point, it may seem as though SAT is irrelevant, given that an IQP can express the same constraints as SAT. While the constraints are expressible by an IQP, on many datasets the IQP solver cannot optimize the objective function $c(A)$ in a reasonable amount of time if it is not provided the sets $S(h)$ to restrict the search space. When given the sets $S(h)$ as input, each optimization is almost instantaneous. Of course, the maximum-likelihood methyl assignment is not guaranteed to be the correct assignment, nor is it in fact the correct assignment in any of the benchmarks on which this method has been tested.

While the optimal assignment may be incorrect, the correct assignment is near-optimal. Moreover, across the benchmarks tested, every resonance is assigned to its true methyl by at least one valid methyl assignment which is very nearly to optimal. This can be exploited by recomputing an analogue of the sets $S(h)$: after of computing possible assignments for each resonance over the set of valid methyl assignments, compute the set of possible assignments for each resonance over the set of near-optimal valid methyl assignments.

Definition 19. *For $\varepsilon > 0, h \in V(H)$, let*

$$S_\varepsilon(h) = \{\pi(h) \mid \pi \in \Pi,\ c(\pi) \leq (1+\varepsilon)c(\pi_{opt})\}$$

The sets $S_\varepsilon(h)$ can be computed as follows. Invoke the IQP solver to find $c(\pi_{opt})$. Add a constraint to the IQP enforcing that $c(A) \leq (1+\varepsilon)c(\pi_{opt})$. At this point, valid methyl assignments that are not nearly optimal will not correspond to feasible solutions of the IQP. Therefore, the set of assignments being considered by the solver of the IQP has been restricted to those which are nearly optimal. The algorithm for computation of $S(h)$ can be naturally translated for enumeration of possible assignments in this restricted domain expressed by the IQP, which is done in Algorithm 4

As on line 12 of Algorithm 3, line 16 of Algorithm 4 is an optimization. The added constraints do not change the set of feasible solutions.

Algorithm 4 Enumeration of near-optimal assignments

```
$\pi_{opt} \leftarrow$ Solve IQP
for $h \in V(H)$ do
    $S_\varepsilon \leftarrow \{\pi_{opt}(h)\}$
end for
Add constraint $c(A) \leq (1+\varepsilon)c(\pi_{opt})$ to IQP
$U \leftarrow \{h \in V(H) \mid |S(h)| > 1\}$
while $U \neq \emptyset$ do
    $h \leftarrow$ random selection from $U$
    for $g \in S_\varepsilon(h)$ do
        Temporarily add constraint $Y_{hg} = 0$ to IQP
    end for
    $\pi \leftarrow$ Get methyl assignment from IQP solver
    Remove temporary constraints
    if $\pi =$ INFEASIBLE then                              ▷ $S_\varepsilon(h)$ is complete
        $U \leftarrow U - \{h\}$
        Add constraint $\sum_{g \in S_\varepsilon(h)} Y_{hg} = 1$ to IQP
    else
        for $u \in U$ do
            $S_\varepsilon(u) \leftarrow S_\varepsilon(u) \cup \{\pi(u)\}$              ▷ Update all sets $S_\varepsilon(h)$
            if $S_\varepsilon(u) = S(u)$ then           ▷ All assignments have been witnessed
                $U \leftarrow U - \{u\}$
            end if
        end for
    end if
end while
```

Chapter 4

Maximum Common Edge Subgraph

4.1 Mostly Valid Assignments

As mentioned previously, `MAGMA` accounts for the possibility that H is not isomorphic to any subgraph of G, i.e., that no valid methyl assignment exists. This can only happen if an edge of H is placed between two vertices that did not in fact exhibit an NOE. Call such an edge fake. On an input containing fake edges, the SAT-based method described thus far is useless, as the SAT solver will simply return UNSAT. The idea deployed effectively by `MAGMA` is to enumerate vertex assignments over the set of invalid methyl assignments that respect as many edges of H as possible.

Definition 20. *The edge score of a methyl assignment π is defined as*

$$e(\pi) := \left|\{\{u, v\} \in E(H) \mid \{\pi(u), \pi(v)\} \in E(G)\}\right|$$

Definition 21. *A methyl assignment π is mostly valid if $\nexists \pi' : e(\pi') > e(\pi)$.*

Definition 22. *For every $v \in V(H)$, let*

$$M(v) := \{\pi(v) \mid \pi \textit{ is a mostly valid methyl assignment}\}$$

The output of MAGMA on instances where no valid methyl assignments exists are the sets $M(h)$ for every H-vertex, which can be described as an analogue of the sets $S(h)$, only defined over the set of mostly valid assignments rather than the set of valid assignments. The SAT-based method can be adapted to compute $M(h)$ using a tool called MARCO [10], which can compute sets of clauses whose removal from an unsatisfiable CNF formula causes the formula to become satisfiable.

4.2 Minimal Correcting Sets

Definition 23. *A correcting set of an unsatisfiable CNF formula F is a set $M \subset F$ such that $F - M$ is satisfiable.*

Definition 24. *A minimal correcting set of an unsatisfiable CNF formula F is a correcting set M of F such that no proper subset of M is a correcting set of F.*

MARCO was designed principally to compute minimal unsatisfiable subsets (MUS) of CNF formulas, also known as minimal unsatisfiable cores, but MARCO is also capable of computing minimal correcting sets (MCS). In this case, MARCO be employed to find minimum sets of edges of H whose removal allows for valid methyl assignments, but this cannot be accomplished by a direct use of MARCO to compute MCSes of the methyl assignment CNF formula. While the aim is discover minimal subsets of $E(H)$ whose removal makes the problem satisfiable, any MCS of the formula may include clauses responsible for enforcing the unique assignment of vertices, and the removal of such a clause should not be entertained. The solution is to rewrite the CNF formula as a group-oriented CNF (GCNF) formula.

Definition 25. *A group-oriented CNF (GCNF) formula is a tuple*

$$\Gamma = (F, J_1, J_2, \ldots, J_k)$$

where F, J_i are CNF formulas. A GCNF formula $(F, J_1, J_2, \ldots, J_k)$ is satisfiable if and only if $F \cup J_1 \cup J_2 \cup \cdots \cup J_k$ is satisfiable.

Definition 26. *A correcting set of a GCNF formula $(F, J_1, J_2, \ldots, J_k)$ is set $M \subseteq [k]$ such that*

$$F \cup \left\{ \bigcup_{i \notin M} J_i \right\}$$

is satisfiable.

Definition 27. *A minimal correct set (MCS) of a GCNF formula Γ is a correcting set M of Γ such that no proper subset of M is a correcting set of Γ.*

In order to determine minimal subsets of $E(H)$ whose removal makes the problem satisfiable, we will construct a GCNF formula $\Gamma = (F, J_1, J_2, \ldots, J_{|E(H)|})$, where F will be the set of clauses responsible for enforcing a one-to-one vertex assignment, and each J_i will be the clauses responsible for enforcing that a particular edge of H is mapped to an edge of G. Algorithm 5 is an adaption of Algorithm 2 which does precisely this.

Given a GCNF formula constructed using this encoding, `MARCO` is able to quickly enumerate the MCSes of Γ of minimum size, that is, all MCSes of Γ that are the size of the smallest MCS of Γ. The output of `MARCO` will from this point be viewed as a series of edge sets, $E_1, E_2, \ldots, E_k$, with each $E_i \subseteq E(H)$; each E_i is a minimal set of edges such that there exists a valid methyl assignment from $H - E_i$ to G.

Algorithm 5 Construction of a GCNF Formula

```
$F \leftarrow \{\}$
for $h \in V(H)$ do
    $D_h \leftarrow \{g \in V(G) \mid \gamma(h) = \gamma(g) \wedge d_H(h) \leq d_G(g)\}$
    $F \leftarrow F \cup \textsc{Commander}(\{X_{hg} \mid g \in D_h\})$
    $F \leftarrow F \cup \{\{X_{hg} \mid y \in D_h\}\}$
end for
for $g \in V(G)$ do
    $F \leftarrow F \cup \textsc{Commander}(\{X_{hg} \mid g \in D_h\})$
end for
$i \leftarrow 0$
for $h, h' \in E(H)$ do
    $i \leftarrow i + 1$
    $J_i \leftarrow \{\}$
    for $g \in D_h$ do
        $C \leftarrow \{\neg X_{hg}\} \cup \{X_{h'g'} \mid g' \in N_G(g) \cap D_{h'}\}$
        $J_i \leftarrow J_i \cup \{C\}$
    end for
end for
return $(F, J_1, J_2, \ldots, J_{|E(H)|})$
```

4.3 Combining Maximum Subgraphs

Definition 28. *For $E \subseteq E(H), h \in V(H)$*

$$S^E(h) = \{\pi(h) \mid \pi \textit{ is a valid methyl assignment from } H - E \textit{ to } G\}$$

Theorem 2. *Let $E_1, E_2, \ldots, E_k$ be the output of* `MARCO`*. For $h \in V(h)$,*

$$M(h) = \bigcup_{E_i} S^{E_i}(h)$$

Proof. It suffices to show that every mostly valid methyl assignment π is also a valid methyl assignment from $H - E_i$ to G, for precisely one E_i, and vice versa. Let $E(\pi) \subseteq E(H)$ be the set of edges that π maps to edges of G. Because $E(\pi)$ is of maximum size, $E(H) - E(\pi) = E_i$ for some i. Therefore π respects all edges in $E(h) - E_i$, meaning that π is a valid methyl assignment from $H - E_i$ to G. The converse follows in much the same way. □

Definition 29. *Let $E_1, E_2, \ldots, E_k$ be the output of* `MARCO`*. For $\varepsilon > 0,\ h \in V(H)$,*

$$M_\varepsilon(h) := \bigcup_{E_i} S_\varepsilon^{E_i}(h)$$

Theorem 2 means that the sets $M(h)$ can be computed by enumerating the sets E_i, then computing S^{E_i} for all such sets. Moreover, in all benchmarks considered, for every $h \in V(H)$, $\pi^*(h) \in S_\varepsilon^{E_i}$ for at least one E_i and a choice of ε small enough increase the number of uniquely assigned vertices.

Algorithm 6 Enumerating possible assignments over all MCES

```
for $h \in V(H)$ do
    $M_\varepsilon(h) \leftarrow \{\}$
end for
$(F, J_1, J_2, \dots, J_{|E(H)|}) \leftarrow$ Invoke Algorithm 5
$E_1, E_2, \dots, E_k \leftarrow$ Invoke MARCO
for $i = 1, 2, \dots, k$ do
    $S^{E_i} \leftarrow$ Invoke Algorithm 3 on $H - E_i,\ G$
    $S_\varepsilon^{E_i} \leftarrow$ Invoke Algorithm 4 on $H - E_i,\ G$
    for $h \in V(H)$ do
        $M_\varepsilon(h) \leftarrow M_\varepsilon(h) \cup S_\varepsilon^{E_i}(h)$
    end for
end for
```

Chapter 5

Comparison With MAGMA

The publication of MAGMA [14] contains results on a series of benchmarks [9, 22, 1, 24, 25]. The method presented herein has been run on the same set of benchmarks. The experiments were run on a 2017 Dell XPS 13 9350, with an Intel Core i5-6200U processor and 8GB RAM. MAGMA was rerun on the same benchmarks on the same machine for comparison. The SAT solver used for testing purposes in every case was cryptominisat [18]. The IQP solver used for the enumeration of near optimal assignments was Gurobi [4]. For every target, the ε used for the definition of near-optimality was 0.002.

Figures 5.1 and 5.2 report the precision and runtime of the method proposed in this thesis compared to those of MAGMA. Note that all of the molecules in MAGMA's benchmark contain fake edges except for Ubiquitin. For ATCase R_2, EIN1, MBP, MSG, and MsrB, Algorithm 6 was used to compute near optimal assignments over the minimal correcting sets of F. For Ubiquitin, since F has no minimal correcting sets, Algorithm 3 and 4 were invoked successively.

Protein	$\lvert V(G)\rvert$	$\lvert E(G)\rvert$	$\lvert V(H)\rvert$	$\lvert E(H)\rvert$	|MCS|	No. MCS
ATCase R_2 [24]	36	156	34	91	9	3
EIN1 [25]	100	391	84	145	1	3
MBP [9]	73	254	70	144	4	10
MSG [22]	159	491	141	230	1	1
MsrB [9]	22	60	21	37	1	1
Ubiquitin [1]	27	45	18	20	0	0

Table 5.1: List of datasets in the `MAGMA` benchmark. For each dataset, the number of vertices and edges of H and G are given, as well as the size of the minimum size minimal correcting set and the number of such minimal correcting sets.

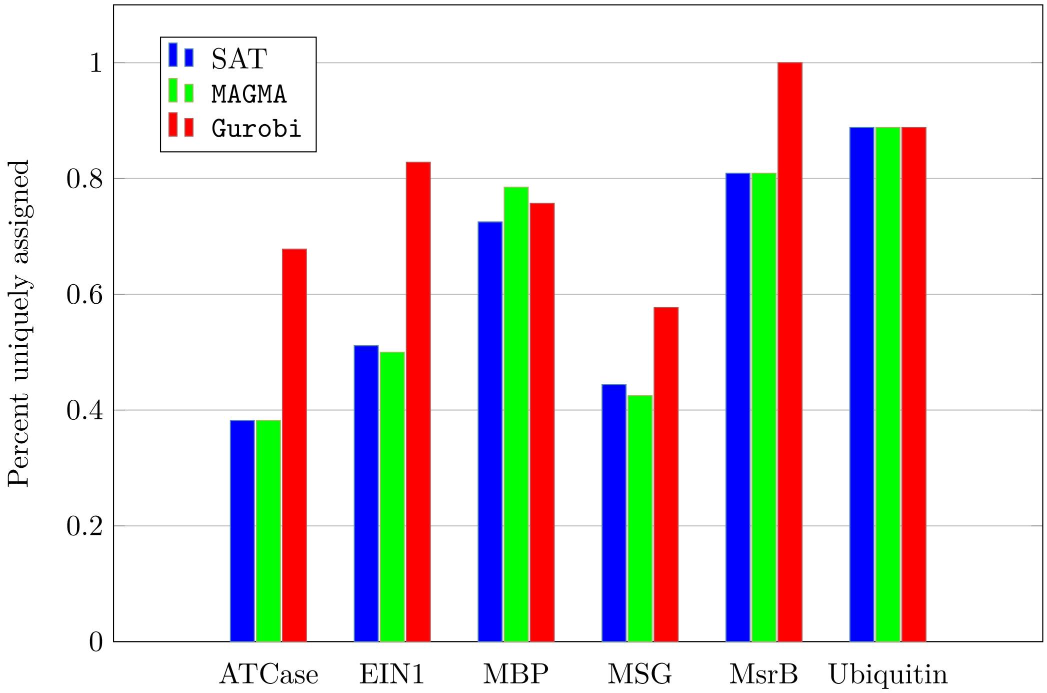

Figure 5.1: Fraction of vertices uniquely assigned by using SAT to compute possible assignments, using `MAGMA` to compute possible assignments, or using `Gurobi` to compute near optimal assignments.

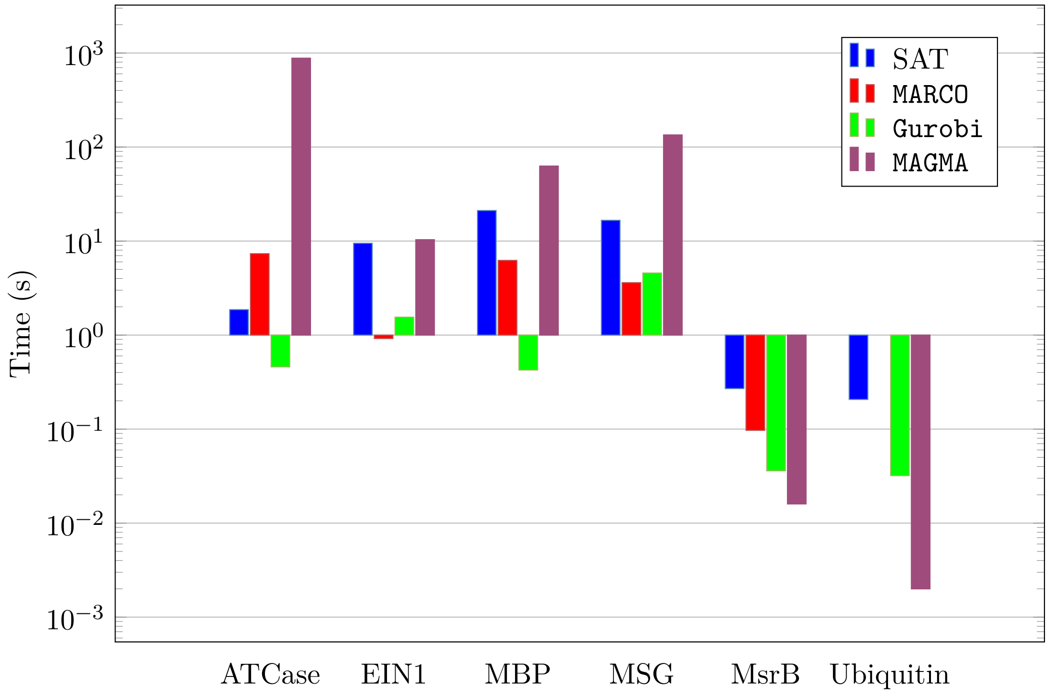

Figure 5.2: Runtime of MARCO, SAT, MAGMA, and Gurobi.

One thing that may seem strange is that there is actually a discrepancy in the fraction of vertices uniquely assigned by SAT and by MAGMA in the cases of EIN1, MBP, and MSG, though the output for both methods purports to be the sets $M(h)$. While the details of MAGMA are unknown, it seems likely that the discrepancy can be attributed to the fact that MAGMA runs on the connected components of H and G individually to derive restricted lists of assignments for each vertex before solving the global methyl assignment problem. For some molecules, MAGMA cannot derive the sets $M(h)$ for the global problem without performing this first step. Of course, the lists derived from the problem after splitting by connected component may differ from the sets $M(h)$, which would likely explain the minor discrepancies here.

This method uniquely assigns more vertices than MAGMA in every case except

EIN1 and Ubiquitin, which is a tie. The most staggering boost is in the case of ATCase, where this method uniquely assigns nearly twice as many vertices as MAGMA. This method also terminates faster than MAGMA on all molecules except MsrB and Ubiquitin. The speedup factor is over 100x for ATCase. This method terminates on MSG in less than half the time of MAGMA. Given that MSG is the largest molecule in the benchmark, and ATCase is the molecule with by far the most fake edges, it appears that SAT scales better in general. Additionally, this method is not operating on connected components of the graphs individually, but rather solving the global problem in one go, which MAGMA cannot do for some targets. Observe that MAGMA terminates an order of magnitude faster on the two smallest molecules, MsrB and Ubiquitin; it appears that SAT, MARCO, and Gurobi may in some sense be overkill for these molecules seeing as the number of (mostly) valid methyl assignments for those molecules are 4 and 2 (see figure 1.2), respectively, and this method will perform $O(|V(G)|^2)$ invocations of cryptominisat and Gurobi regardless.

Bibliography

[1] F. A. Chao, J. Kim, Y. Xia, M. Milligan, N. Rowe, and G. Veglia. FLAMEnGO 2.0: an enhanced fuzzy logic algorithm for structure-based assignment of methyl group resonances. *J. Magn. Reson.*, 245:17–23, Aug 2014.

[2] F. A. Chao, J. Kim, Y. Xia, M. Milligan, N. Rowe, and G. Veglia. FLAMEnGO 2.0: an enhanced fuzzy logic algorithm for structure-based assignment of methyl group resonances. *J. Magn. Reson.*, 245:17–23, Aug 2014.

[3] Aarti Gupta, Malay K. Ganai, and Chao Wang. Sat-based verification methods and applications in hardware verification. In *Proceedings of the 6th International Conference on Formal Methods for the Design of Computer, Communication, and Software Systems*, SFM'06, pages 108–143, Berlin, Heidelberg, 2006. Springer-Verlag.

[4] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2018.

[5] Marijn J. H. Heule and Oliver Kullmann. The science of brute force. *Commun. ACM*, 60(8):70–79, 2017.

[6] Andrei Horbach. A boolean satisfiability approach to the resource-constrained project scheduling problem. *Annals of Operations Research*, 181(1):89–107, Dec 2010.

[7] Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, pages 85–103, 1972.

[8] Will Klieber and Gihwon Kwon. Efficient cnf encoding for selecting 1 from n objects. In *Fourth Workshop on Constraints in Formal Verification*, 2007.

[9] O. F. Lange, P. Rossi, N. G. Sgourakis, Y. Song, H. W. Lee, J. M. Aramini, A. Ertekin, R. Xiao, T. B. Acton, G. T. Montelione, and D. Baker. Determination of solution structures of proteins up to 40 kDa using CS-Rosetta with sparse NMR data from deuterated samples. *Proc. Natl. Acad. Sci. U.S.A.*, 109(27):10873–10878, Jul 2012.

[10] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and João Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2):223–250, 2016.

[11] D. Marion. An introduction to biological NMR spectroscopy. *Mol. Cell Proteomics*, 12(11):3006–3025, Nov 2013.

[12] Ralph Eric Mcgregor. *Automated Theorem Proving Using Sat.* PhD thesis, Potsdam, NY, USA, 2011. AAI3471671.

[13] Y. R. Monneau, P. Rossi, A. Bhaumik, C. Huang, Y. Jiang, T. Saleh, T. Xie, Q. Xing, and C. G. Kalodimos. Automatic methyl assignment in large proteins by the MAGIC algorithm. *J. Biomol. NMR*, 69(4):215–227, Dec 2017.

[14] Iva Pritišanac, Matteo T. Degiacomi, T. Reid Alderson, Marta G. Carneiro, Eiso AB, Gregg Siegal, and Andrew J. Baldwin. Automatic assignment of methyl-nmr spectra of supramolecular machines using graph theory. *Journal of the American Chemical Society*, 139(28):9523–9533, 2017. PMID: 28691806.

[15] P. Rossi, Y. R. Monneau, Y. Xia, Y. Ishida, and C. G. Kalodimos. Toolkit for NMR Studies of Methyl-Labeled Proteins. *Meth. Enzymol.*, 614:107–142, 2019.

[16] P. Schanda, E. Kupce, and B. Brutscher. SOFAST-HMQC experiments for recording two-dimensional heteronuclear correlation spectra of proteins within a few seconds. *J. Biomol. NMR*, 33(4):199–211, Dec 2005.

[17] João P. Marques Silva and Karem A. Sakallah. GRASP - a new search algorithm for satisfiability. In *ICCAD*, pages 220–227, 1996.

[18] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In *Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings*, pages 244–257, 2009.

[19] R. Sprangers, A. Gribun, P. M. Hwang, W. A. Houry, and L. E. Kay. Quantitative NMR spectroscopy of supramolecular complexes: dynamic side pores in ClpP are important for product release. *Proc. Natl. Acad. Sci. U.S.A.*, 102(46):16678–16683, Nov 2005.

[20] R. Sprangers and L. E. Kay. Quantitative dynamics and binding studies of the 20S proteasome by NMR. *Nature*, 445(7128):618–622, Feb 2007.

[21] Marc Thurley. sharpsat - counting models with advanced component caching and implicit BCP. In *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, pages 424–429, 2006.

[22] V. Tugarinov, W. Y. Choy, V. Y. Orekhov, and L. E. Kay. Solution NMR-derived global fold of a monomeric 82-kDa enzyme. *Proc. Natl. Acad. Sci. U.S.A.*, 102(3):622–627, Jan 2005.

[23] Miroslav N. Velev and Ping Gao. Efficient SAT techniques for absolute encoding of permutation problems: Application to hamiltonian cycles. In *Eighth Symposium*

on Abstraction, Reformulation, and Approximation, SARA 2009, Lake Arrowhead, California, USA, 8-10 August 2009, 2009.

[24] A. Velyvis, H. K. Schachman, and L. E. Kay. Assignment of Ile, Leu, and Val methyl correlations in supra-molecular systems: an application to aspartate transcarbamoylase. *J. Am. Chem. Soc.*, 131(45):16534–16543, Nov 2009.

[25] V. Venditti, N. L. Fawzi, and G. M. Clore. Automated sequence- and stereo-specific assignment of methyl-labeled proteins by paramagnetic relaxation and methyl-methyl nuclear Overhauser enhancement spectroscopy. *J. Biomol. NMR*, 51(3):319–328, Nov 2011.

[26] Y. Xiao, L. R. Warner, M. P. Latham, N. G. Ahn, and A. Pardi. Structure-Based Assignment of Ile, Leu, and Val Methyl Groups in the Active and Inactive Forms of the Mitogen-Activated Protein Kinase Extracellular Signal-Regulated Kinase 2. *Biochemistry*, 54(28):4307–4319, Jul 2015.

[27] Y. Xu, M. Liu, P. J. Simpson, R. Isaacson, E. Cota, J. Marchant, D. Yang, X. Zhang, P. Freemont, and S. Matthews. Automated assignment in selectively methyl-labeled proteins. *J. Am. Chem. Soc.*, 131(27):9480–9481, Jul 2009.

[28] Y. Xu and S. Matthews. MAP-XSII: an improved program for the automatic assignment of methyl resonances in large proteins. *J. Biomol. NMR*, 55(2):179–187, Feb 2013.

[29] Catherine Zwahlen, Kevin H. Gardner, Siddhartha P. Sarma, David A. Horita, R. Andrew Byrd, and Lewis E. Kay. An nmr experiment for measuring methyl-methyl noes in 13c-labeled proteins with high resolution. *Journal of the American Chemical Society*, 120(30):7617–7625, 1998.