Convex Optimization Approaches for
NMR Assignment

José Frederico Simões Bravo Ferreira

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Program in
Applied and Computational Mathematics
Adviser: Professor Amit Singer

November 2018
Abstract

Nuclear Magnetic Resonance Spectroscopy (NMR) is the primary tool for structural determination of proteins in solution. It distinguishes itself from other such tools by uncovering geometric constraints at the atomic level, which are utilized in a maximal constraint satisfaction problem to produce accurate structures without requiring crystallization. Despite its widespread use, full automation of the NMR pipeline has not yet been achieved. Chief among the problematic steps in NMR spectroscopy is the problem of backbone assignment, which can be understood as a labeling step in which atoms are tagged with their resonance frequencies. This labeling is crucial for the construction of structural constraints, and consequently, for the determination of accurate structures.

In this thesis, we describe convex optimization approaches to tackle the combinatorial problem of NMR backbone assignment. These approaches differ from mainstream solutions by seeking to find a single, maximum-likelihood solution via global optimization, rather than attempting to solve non-convex problems through heuristics. Chapter 2 introduces the first such approach, C-SDP, which is, at its core, a semidefinite programming relaxation to the Quadratic Assignment Problem (QAP) with PSD variables of dimension $O(n) \times O(n)$. An efficient distributed ADMM algorithm is described for solving the relaxation to optimality, producing strong lower bounds on several large QAP and TSP problems, and recovering optimal (near rank 1) solutions on NMR datasets. Chapter 3 describes a better scaling, linear programming approach to tackle NMR assignment on both spin system and peak list datasets. The resulting assignment pipeline is successfully applied to both simulated and experimental datasets, with state-of-the-art results on both.
Acknowledgements

I find it impossible to properly acknowledge the people who contributed to my graduate studies without offering a little context on my views of graduate school as a whole, but please don’t hesitate to jump right into the thank yous if that doesn’t interest you!

I have often debated the merits and flaws of the academic system with close friends, in private. I have questioned the driving factors that keep it afloat, and have met too many disgruntled, disillusioned, and burnt out graduate students to ignore these questions. The power imbalances inherent to the system force many to stay beyond their liking, and many others to feel like they have lost control over their own research, and their own personal and professional development, especially since abandoning a PhD program is still taboo, and oftentimes extremely prejudicial. Perhaps this is too dark a picture to paint in a document that is meant to highlight positive contributions, but it is this perspective that makes me deeply appreciative of the people who, through their support, made my own graduate experience a positive one, and mostly stress-free. So, onto the thank yous.

First and foremost, I owe a great debt of gratitude to Amit Singer, who welcomed me into his group and who first steered me towards ongoing work on NMR spectroscopy. Amit oversaw all my work, contributed fundamental insights and crucial suggestions at fulcrum junctures of our projects, and never hesitated to reposition my focus where it belonged. Most importantly, however, Amit understood and accepted my own goals and ambitions, and awarded me the freedom to set my own objectives and to take charge of my own professional development, even when it meant going away to different continents to pursue internship opportunities in mostly unrelated topics. I am forever grateful to Amit for his fairness, good judgement, and support, both technical and personal, throughout my time here.
I am also greatly indebted to my collaborators, committee members, and readers. Yuehaw Khoo, whose ideas and input lay much of the foundation for the work described in this thesis, is owed as much credit for our results as I am. David Cowburn kindly accommodated my naive interpretations of the assignment problem, and I thank him for patiently guiding me towards an understanding of the problem that is better grounded on the challenges faced by NMR practitioners (and hope that we are finally in a position to help!). Amirali Ahmadi, who served in both my generals and thesis committees, taught the clearest convex optimization course that I have witnessed, and was an overall amazing academic role model, for which I am very grateful. Finally, Henry Wolkowicz, whose work underlies much of this thesis, kindly agreed to serve as a Reader, reviewing my thesis, and reassuring me that no major blunders were present.

The extended Amit Singer group, collectively, also helped shape my perspectives on our work, and I truly appreciate all of their input. A special thanks goes out to Nicolas Boumal, whose course on numerical analysis I was lucky to TA (twice!), and to the graduate students and company who shared the journey with me, particularly Afonso Bandeira and Yutong Chen (who first showed me the PACM ropes), Tejal Bhamre and Yuan Liu (with whom I bonded over many coursework problems), and João Carreira Pereira and Ane Ferri Victoria (who ensured my Portuguese stayed sharp throughout). I am also thankful to the staff of PACM, and in particular Audrey Mainzer, Tina Dwyer, and Valerie Marino, whose help throughout my time here made everything much easier.

My graduate school experience would have been ultimately meaningless if not for the friends I made along the way. I have agonized over how best to thank everyone in a way that does not collapse a rich relationship into a trivial quip, but have come out awfully short. So I’ll keep things simple. At the risk of leaving many wonderful people out, I’m immensely thankful to Alex Tarr, Allen Xia, Bernat Guillen Pegueroles,
Jonathan Balkind, Koushiki Bose, Kyle Felker, Maciej Halber, Raghav Sethi, Sara Rodriguez Martinez, Siddharth Mishra Sharma, and Zander Berg for their friendship. I reserve a special thank you to Lian Zhu, for her loving companionship, and for knowing me better than I know myself.

I have been extraordinarily lucky to have the support of my family in all decisions, good and bad. Their encouragement, from the moment I left Portugal to pursue my undergraduate studies, has always been unwavering. I am sure my mom and grandmother still wish I was a real doctor, but I also known they forgive me. I miss home more than I can put into words, and have regretted the lost birthdays, holidays, and day-to-day life at home on so many occasions. This thesis is dedicated to my parents and grandparents, for making me who I am, and to my brother, whom I’ll never cease to admire.
To my family.
Contents

Abstract ........................................ iii
Acknowledgements ................................. iv
List of Tables .................................... xi
List of Figures .................................... xii

1 Introduction 1
1.1 The assignment problem ...................... 3
   1.1.1 Homonuclear resonance assignment ............ 5
   1.1.2 Heteronuclear resonance assignment ............ 7
   1.1.3 Challenges in sequential assignment .......... 12
1.2 Convex optimization .......................... 14
   1.2.1 Convexity and optimization .................... 14
   1.2.2 Linear programming ........................... 15
   1.2.3 Semidefinite programming ....................... 17
   1.2.4 Convex relaxations ............................ 19
1.3 Existing work .................................. 22
   1.3.1 RANDOM ........................................ 24
   1.3.2 IPASS .......................................... 26
   1.3.3 FLYA ........................................... 28
1.4 A note on generative assumptions .............. 30
1.5 Our contributions .............................. 30
## Assignment through Semidefinite Programming

### 2.1 Related Work

### 2.2 CSDP - a convex relaxation

- **2.2.1 SDP relaxation of Zhao et al.**
- **2.2.2 E-SDP relaxation**
- **2.2.3 C-SDP relaxation**
- **2.2.4 Illustrative example using the path graph**

### 2.3 Solving C-SDP with ADMM

- **2.3.1 Rewriting constraints in E-SDP relaxation**
- **2.3.2 Dual problem and the ADMM updates**
- **2.3.3 An empirical note on ADMM convergence**

### 2.4 Evaluation of C-SDP

- **2.4.1 Lower bounds**
- **2.4.2 Upper bounds**

### 2.5 C-SDP for NMR assignment

- **2.5.1 Accounting for a prior**
- **2.5.2 Benchmark datasets**
- **2.5.3 Performance on experimental datasets**

### 2.6 Conclusion

## Assignment through Linear Programming

### 3.1 Modeling NMR assignment

### 3.2 Building an assignment graph

- **3.2.1 Graph construction for spin systems**
- **3.2.2 Graph construction for peak lists**
- **3.2.3 Choosing δ**
- **3.2.4 Dummy, start, and end nodes**

### 3.3 Path finding
# List of Tables

2.1 Lower bounds for QAPLIB problems ........................................... 58
2.2 Upper bounds for QAPLIB problems ........................................... 62
2.3 Performance of C-SDP on simulated NMR datasets (low noise) .... 68
2.4 Performance of C-SDP on simulated NMR datasets (high noise) ... 68
2.5 Performance of C-SDP on real datasets ...................................... 69
2.6 Lower bounds for TSPLIB problems .......................................... 71
2.7 Upper bounds for TSPLIB problems .......................................... 72
3.1 Performance of LP on simulated spin datasets (low noise) .......... 90
3.2 Performance of LP on simulated spin datasets (high noise) ....... 91
3.3 Performance of LP on SH2 peak list datasets .......................... 92
3.4 Performance of LP on real datasets .......................................... 92
# List of Figures

1.1 Sketch of typical NMR pipeline. ........................................ 2
1.2 Illustration of $^1$H-$^1$H NOESY .................................................. 4
1.3 1D $^1$H-spectrum for ubiquitin ............................................. 6
1.4 Fingerprint regions of the COSY spectrum ............................. 6
1.5 HSQC, HNCACB, HN(CO)CACB spectra for backbone assignment.. 9
1.6 Sketch of backbone assignment procedure through heteronuclear NMR. 11
1.7 HNCA and HN(CO)CA spectra ................................................. 12
1.8 HNCO and HN(CA)CO spectra ................................................ 13
2.1 E-SDP and C-SDP variables in the path graph problem with 5 nodes. 42
2.2 Comparison of Conic-ADMM3c and simple 3-block ADMM ............... 55
2.3 Runtimes for QAPLIB problems ............................................. 57
2.4 Lower bounds for TSPLIB problems ........................................ 59
2.5 Runtimes for TSPLIB problems ............................................. 60
2.6 Upper bounds for TSPLIB problems ........................................ 63
3.1 Illustration of graph assignment model .................................. 76
Chapter 1

Introduction

Nuclear Magnetic Resonance Spectroscopy (NMR) has established itself as the tool of choice for structural determination of proteins in solution. The analysis of NMR data allows one to establish geometric constraints at the atomic level which restrict the conformations allowed for the protein. Armed with a large set of accurate geometric constraints, the conformation that maximizes constraint satisfaction is likely a good candidate for the true structure of the protein.

A pictorial depiction of a typical NMR pipeline is presented in Figure 1.1 below. As illustrated, the structural determination pipeline involves a number of steps, which can be broken down into: (1) experiments, (2) data processing, (3) peak picking, (4) chemical shift assignment, (5) geometric constraint determination, and finally (6) structural determination [20].

One of the major challenges faced by NMR practitioners lies in obtaining a set of accurate constraints. Both experimental and computational difficulties undermine the quality of the geometric constraints that can be extracted from the data. Further, unlike other methods used for structural determination of macro-molecules such as Cryo-EM and X-ray crystallography, the computational aspects of the NMR pipeline have so far escaped full automation. Expert supervision is typically required, both
Figure 1.1: Sketch of typical NMR pipeline. A protein of unknown structure is placed in solution, NMR experiments produce relaxation signals, Fourier analysis produces spectra from raw signal, peak picking selects local extrema of spectra, and excludes artifacts, chemical shift assignment produces labeled atoms, and labels are used to derive geometric constraints, allowing for structural determination through constraint satisfaction.

in the completion of intermediate steps, such as chemical shift assignment, and in revising previously derived assignments and constraints from the end structure, in an iterative fashion.

This iterative nature of the structural determination pipeline can often lead to problems such as model bias. Further, the skill of experts during assignment, in particular, and a subjective bias during structural determination, in general, can have undue impact on the end structure. Assignment of larger proteins (>25 kDa) is also hampered by lower signal-to-noise ratios, which greatly impact the quality of the signal, and by the ambiguities introduced into the spectral data due to the superposition of atom resonance frequencies.
Attempts to lessen the impact of these factors on the quality of the protein structures have emphasized two major steps in the pipeline: assignment and structural determination. Improved assignment quality is crucial for the production of accurate geometric constraints which, in turn, greatly facilitate accurate structural determination. A complementary approach is to increase the robustness of the structural determination step to outliers, thereby allowing accurate protein structures to be extracted from noisier constraint sets.

This thesis focuses on the former problem of chemical shift, or spectral, assignment. Unlike structural determination, assignment has not been successfully automated, although partial assignments can nowadays be determined in a fully automated fashion. Ultimately, even reasonable partial assignments must still be reviewed by experts, and the process remains time-consuming and error-prone. Full automation, or at the very least simplification of spectral assignment, has the potential to vastly accelerate the NMR pipeline. More importantly, high quality spectral assignment would improve structural determination, eliminating many of the ambiguities and inconsistencies typically present in the geometric constraint set.

1.1 The assignment problem

The spectral assignment problem is the problem of determining the resonance frequencies of individual atoms in the protein. These frequencies are typically defined by their chemical shifts, measured in ppm relative to a reference compound since they tend to depend on the local environment of individual nuclei [20]. To better understand where the term *assignment* is derived, we return to the NMR pipeline as depicted in Figure 1.1.

An NMR practitioner starts with proteins in solution, whose covalent structure (protein sequence) is known. As a result, the bond lengths and bond angles within
the covalent structure are well defined. However useful, there remain a multitude of degrees of freedom within which the protein is allowed to fold and twist, such that the global conformation is unknown.

To extract constraints from which one can deduce global conformation, NMR spectroscopists make use of interactions between atom nuclei, such as the nuclear Overhauser effect (NOE), among others. This effect arises from the dipolar relaxation of a two-spin system, and manifests itself as an off-diagonal peak in NOE spectroscopy experiments (NOESY) [20], as illustrated in Figure 1.2.

![Figure 1.2: Illustration of a protein with two hydrogen atoms in close spatial proximity (left), which induce off-diagonal peaks in $^1$H-$^1$H NOESY spectrum (right).](image)

The efficiency of mixing in dipolar cross-relaxation depends on distance [20], such that cross-peaks in H-H NOESY spectra are indicative of the existence of two hydrogen atoms within close proximity. However, this information is not immediately useful geometrically without the knowledge of which hydrogen induce the cross-peak. It is the solution to the assignment problem which provides this information, by mapping the chemical shifts observed in this and other NMR spectroscopy experiments to the corresponding atoms in the protein. The assignment of all protons is a crucial first-step for high-resolution structure determination in NMR [49].
1.1.1 Homonuclear resonance assignment

Small proteins (<10-12kDa) can often be assigned using only proton experiments, such as NOESY ([49], [20]). For the interested reader, a detailed example of this approach is the assignment of the second IgG-binding domain of *Streptococcus* Protein G presented in [48]. We provide a brief summary of the experiments involved before describing the heteronuclear experiments that form the basis of the modern resonance assignment toolbox.

The simplest \(^1\)H experiment one can analyze is the 1D \(^1\)H spectrum, illustrated in Figure 1.3 below. Distinct peaks can be observed, corresponding to the resonance chemical shifts of individual hydrogen atoms present in the protein. The signal to noise ratio (SNR) decreases with increasing protein size, while linewidths increase, rendering this type of spectra less useful for assignment due to degenerate resonances.

Correlated spectroscopy experiments, such as COSY and TOCSY, rely on coherence or magnetization transfer between coupled spins within the same residue [20]. Unlike the \(^1\)H experiment described above, these experiments are inherently two-dimensional. Certain regions within each of these spectra, such as the \(^1\)H\(^N\)–\(^1\)H\(^\alpha\) region in the COSY spectrum highlighted in Figure 1.4, tend to depict well-separated cross-peaks, and are thus termed fingerprint regions. These regions are utilized to disambiguate chemical shifts belonging to different hydrogen atoms within the same residue.

Finally, as discussed above, cross-relaxation experiments such as NOESY provide information on proton correlations due to through-space dipolar interactions (rather than through covalent bonds). But in addition to being crucial in deriving geometric constraints, NOE connectivities between side-chain hydrogens can help further disambiguate the frequency assignments derived from COSY and TOCSY spectra.

Certain NOE connectivities, such as between \(^1\)H\(^\alpha\) or \(^1\)H\(^\beta\) of residue \(i\) and \(^1\)H\(^N\) of residue \(i + 1\) can be used to link sequential residues. Starting from amino acids
Figure 1.3: 1D $^1$H-spectrum for ubiquitin. Reproduced with permission from [20], chapter 6, Figure 6.1. (a) 2-mM ubiquitin in H$_2$O, (b) inset of methyl group of Leu50 (indicated by the arrow), (c) 2-mM ubiquitin in 8-M urea/H$_2$O solution (in denatured stated, leading to reduced chemical-shift dispersion).

Figure 1.4: The five fingerprint regions of the COSY spectrum. Reproduced with permission from [20], chapter 6, Figure 6.6.
which present distinct chemical shift signatures one can thus use these connectivities to walk along the protein, in a process often termed sequential walking. This linking between adjacent residues also forms the basis for assignment through heteronuclear resonance experiments, and will be made clearer in the following subsection and the remainder of this thesis.

1.1.2 Heteronuclear resonance assignment

In larger proteins, increased linewidths, lower SNR, and a sheer increase in resonances can lead to ambiguous $^1$H spectra. Heteronuclear NMR experiments utilize NMR-active isotopes in $^{13}$C- and $^{15}$N-labeled proteins to increase the dimensionality of NMR spectra in order to resolve ambiguities that would otherwise occur in the proton dimension alone.

This thesis focuses primarily on a collection of 7 heteronuclear experiments regularly used for the assignment of the protein’s backbone atoms. For medium-sized proteins (<150 residues) the HSQC, HNCACB, and HN(CO)CACB experiments are often sufficient for full backbone assignment, with HSQC – a two-dimensional experiment described below – serving as the fingerprint spectra. As protein size increases further, additional experiments are necessary to disambiguate the spectra. The full set of experiments considered is thus HSQC, HNCACB, HN(CO)CACB, HNCA, HN(CO)CA, HNCO, and HN(CA)CO. We offer a simplified description of the output of each of these experiments in terms of the observable peaks (full details on the theory of these experiments can be found in reference textbooks such as [20] and [55]).

**Assignment of medium-sized proteins (<150 residues)**

**HSQC** The heteronuclear single quantum coherence experiment [15] involves a transfer of magnetization between the base amide proton, $^1$H$^N$, and the nitrogen
$^{15}$N and back, as illustrated in Figure 1.5. With the exception of proline, all basic amino acids feature this amide pair, such that a distinct peak can be expected for most residues, leading to the use of HSQC as a fingerprinting experiment.

**HNCACB** This experiment involves magnetization transfer from $^1$H$^a$ and $^1$H$^b$ to $^{13}$C$^a$ and $^{13}$C$^b$, respectively, and then from $^{13}$C$^b$ to $^{13}$C$^a$ and finally to $^{15}$N and to $^1$H$^N$ of the same or subsequent residue, as illustrated in Figure 1.5 and described in [34]. The polarities of the $^{13}$C$^a$ and $^{13}$C$^b$ peaks are opposite, which allows these to be distinguished. Importantly, note that $^{13}$C$^a$ and $^{13}$C$^b$ peaks are observed with the same root $^1$H$^N-^{15}$N pair. Like with the NOESY spectra described in subsection 1.1.1 this will facilitate the adoption of a sequential walking approach to assignment.

**HN(CO)CACB** The last of the experimental toolset for backbone assignment of medium-size proteins also gives rise to $^{13}$C$^a$ and $^{13}$C$^b$ peaks [33], as illustrated in Figure 1.5. Magnetization transfer happens from $^1$H$^a$ and $^1$H$^b$ to $^{13}$C$^a$ and $^{13}$C$^b$, onto $^{13}$CO and finally the base amide pair. Chemical shifts are evolved only on $^{13}$C$^a$ and $^{13}$C$^b$ before detection, so no $^{13}$CO peaks are observed.

**Assignment** We take pause here to describe how the three heteronuclear experiments described above can be used to deduce an accurate backbone assignment. The assignment process is summarized below, with reference to Figure 1.6:

1. HSQC is used as a fingerprint experiment due to high sensitivity and resolution, allowing for accurate determination of base $^1$H-^{15}$N pairs. Peaks in HSQC are matched with peaks in HNCA and HN(CO)CACB spectra which satisfy tolerance bounds (typically 0.02-0.03ppm for hydrogen and 0.2-0.3ppm for nitrogen).
Figure 1.5: Illustration of three heteronuclear spectra for backbone assignment. HSQC is used as a fingerprinting experiment. Peaks in HNCACB and HN(CO)CACB develop off the $^1$H-$^{15}$N plane, along the carbon dimension. Polarity differences of antiphase peaks in HNCACB disambiguate $^{13}$C$^{\alpha}$ from $^{13}$C$^{\beta}$, and HN(CO)CACB further disambiguates intra- from inter-residue atoms.
(2) Peaks in HNCACB and HN(CO)CACB within the same $^1$H—$^{15}$N grouping are correlated and disambiguated using phase information, allowing for the assembly of spin systems.

(3) Fragments are created through sequential walking along the $^{13}$C$\alpha$ and $^{13}$C$\beta$ chains.

(4) Fragments are placed along the protein chain by statistical typing – chemical shifts of atoms in fragment are compared with global chemical shift statistics collected in a public database such as BMRB [61], and placed optimally according to that prior.

Note that the widespread availability of NMR data collected in databases such as BMRB is of fundamental in assignment. The distributions of chemical shifts in different amino acid types is not the same, due to the unique environment induced by the different chemical structures. Certain amino acids (alanine, glycine, isoleucine, leucine, proline, serine, threonine, and valine) present particularly distinct signatures.

**Assignment of large proteins ($\geq$150 residues)**

In larger, slower tumbling proteins, even the heteronuclear spectra described above are not sufficiently discriminative for backbone assignment. For the purposes of this thesis, we consider four additional experiments, which are summarized below. These include the carbonyl $^{13}$CO in addition to additional $^{13}$C$\alpha$.

**HNCA** This experiment, described in [42] and [30], is a useful addition for proteins with ambiguous HNCACB and HN(CO)CACB spectra. Magnetization transfer starts at the $^1$H$^N$ and is passed to $^{15}$N and then between $^{15}$N and $^{13}$C$\alpha$ of both the same and preceding residues, as seen in Figure 1.7. The two peaks formed are in-phase, but the intra-residue peak is typically more intense and is not observed in the HN(CO)CA experiment described below.
Figure 1.6: Sketch of backbone assignment procedure through heteronuclear NMR.

**HN(CO)CA**  Unlike HNCA, this experiment correlates the base $^1H^N-^{15}N$ pair to the $^{13}C\alpha$ atom of the preceding residue only, via magnetization transfer through the carbonyl (whose chemical shift is not evolved, see Figure 1.7) [12]. An early historical example of the use of HNCA and HN(CO)CA spectra together is presented in [35].

**HNCO**  This experiment, illustrated in Figure 1.8, utilizes the J coupling between $^{15}N$ and the carbonyl $^{13}CO$ to correlate the base amide pair with the carbonyl of the preceding residue [42], [35]. It is an excellent fingerprinting alternative to the HSQC, with the carbonyl dimension allowing for disambiguation of overlapping peaks in larger proteins.
HN(CA)CO Complementing HNCO, the HN(CA)CO experiment correlates the amide pair and the intra-residue $^{13}C$ via the $^{13}C$—$^{13}CO J$ coupling [24], as seen in Figure 1.8. The carbonyl for the preceding residue can also be observed. When combined, these two experiments allow for sequential assignment along the carbonyls.

1.1.3 Challenges in sequential assignment

As described above, accurate assignment relies on the correct identification of peaks in NMR experiments, and their assembly into consistent spin systems that can be sequentially assigned.
Figure 1.8: Illustration of HNCO and HN(CA)CO spectra. Combined, these spectra can yield unambiguous $^{13}$CO chemical shifts and facilitate sequential walking along the carbonyl chain.

In practice, as the quality of NMR spectra deteriorates, some peaks will overlap, and others cannot be detected at all, due to linewidth increase and lower SNR. Artifacts included in automatically selected peak lists further hamper sequential assignment. Even with a decent set of spin systems, sequential assignment itself is not as simple as solving a one-dimensional puzzle, as experimental noise, erroneous spin systems, overlapping chemical shifts, and missing spin systems introduce ambiguity to the process.
The next section provides a high level overview of our approach in tackling some of these challenges, and a brief summary of the convex optimization tools we use throughout the remainder of the thesis.

1.2 Convex optimization

This thesis focuses on the formulation of sequential assignment as a global, non-convex maximum-likelihood problem. We then present convex optimization approaches to arrive at a satisfactory solution to the non-convex problem, making use of tools such as linear programming, semidefinite programming, as well as first-order methods such as alternating direction method of multipliers (ADMM) for efficiently solving some of these problems.

We therefore begin this section with a review of convexity and an introduction to the concepts enumerated above. Much of the material presented in this section can be readily found in standard convex optimization textbooks, such as [17].

1.2.1 Convexity and optimization

A convex optimization problem is a problem of the form

**Problem 1.1 (Convex Problem).**

\[
\min_x \quad f(x) \\
\text{s.t.} \quad x \in S
\]

where \( f \) is a convex function and \( S \) is a convex set, as per the following definitions

**Definition 1.1 (Convex set).** A set \( S \in \mathbb{R}^d \) is convex if and only if for all \( x, y \in S \) and \( \lambda \in [0, 1] \)

\[
\lambda x + (1-\lambda)y \in S.
\]
Definition 1.2 (Convex function). A function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) is convex if and only if for all \( x, y \in \mathbb{R}^d \) and \( \lambda \in [0, 1] \)

\[
f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).
\]

The importance of convexity in the context of optimization is due to the fact that for a convex problem, \textbf{any local minimizer of} \( f \) \textbf{in the set} \( S \) \textbf{is also globally optimal}.

Two types of convex problems are particularly relevant to the work presented in this thesis: linear programming, and semidefinite programming.

1.2.2 Linear programming

A linear programming problem (LP) is a problem which can be written in the following canonical form

\textbf{Problem 1.2 (Linear Program).}

\[
\begin{align*}
\min_{x \in \mathbb{R}^d} & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

An example of a problem that can be written in this form is that of finding a shortest path between two nodes in a weighted directed graph (we assume positive weights to exclude the possibility of negative cycles and also to facilitate intuition):

\textbf{Problem 1.3 (Shortest Path).} Given a graph, \( G \), with node set \( \mathcal{V} \) and edge set \( \mathcal{E} \), find the shortest path between nodes \( s, t \in \mathcal{V} \).
Let \( w_{ij} \geq 0 \) be the weight of edge \((i, j)\) between nodes \(i\) and \(j\). Then the problem is equivalent to

\[
\min_{\{x_{ij}\}} \sum_{(i,j) \in \mathcal{E}} w_{ij} x_{ij} \\
\text{s.t.} \quad \sum_{j: (i,j) \in \mathcal{E}} x_{ij} - \sum_{j: (j,i) \in \mathcal{E}} x_{ji} = \begin{cases} 
1 & \text{if } i = s \\
-1 & \text{if } i = t \\
0 & \text{otherwise}
\end{cases} \\
x_{ij} \geq 0 \quad \forall i,j \in \mathcal{E}
\]

To better understand the first constraint, one can first consider the case where all \( x_{ij} \in \{0, 1\} \) such that \( x_{ij} \) is an indicator variable for whether edge \( ij \) is included in the path. Then, the quantity

\[
\sum_{j: (i,j) \in \mathcal{E}} x_{ij} - \sum_{j: (j,i) \in \mathcal{E}} x_{ji}
\]

is the number of outgoing edges from node \(i\) minus the number of incoming edges to node \(i\). This quantity must be 1 for the start node \(s\), -1 for the target node \(t\), and 0 for every other node in the path. As a result, the problem amounts to selecting directed edges in the graph that satisfy the path constraint and which sum to the smallest possible weight.

Equivalency between the solution with integral \( x_{ij} \) and the given linear program follows from the fact that \textbf{every bounded linear program has a vertex solution} [2], that is, a solution on a vertex of the feasible set, and the feasible set in this case is a polytope whose vertices are valid paths.
1.2.3 Semidefinite programming

A semidefinite program (SDP) is a optimization problem that can be written in the following standard form

Problem 1.4 (Semidefinite program).

\[
\begin{align*}
\min_{X \in \mathbb{R}^{n \times n}} & \quad \text{Tr} (CX) \\
\text{s.t.} & \quad \text{Tr} (A_i X) = b_i, \quad i = 1, \ldots, p \\
& \quad X \succeq 0
\end{align*}
\]

where the inequality \( X \succeq 0 \) indicates positive semidefiniteness of \( X \), such that \( v^T X v \geq 0, \forall v \in \mathbb{R}^n \). Understood more broadly, this is a conic optimization problem over the cone of positive semidefinite matrices. Note that a linear program also falls within the conic optimization paradigm, this time over the positive orthant.

This type of problem arises naturally in a variety of fields, such as control theory (in the context of linear matrix inequalities, see e.g. [16]), but it has recently gained traction as a powerful tool for approximation in NP-hard problems. A landmark example of this is Goemans and Williamson’s approximation algorithm for the max-cut problem [32], defined as follows

Problem 1.5 (Max-cut). Given an undirected, weighted graph \( \mathcal{G} \), with node set \( \mathcal{V} \) with \( |\mathcal{V}| = n \) and edge set \( \mathcal{E} \), find a partition of the node set into \( \mathcal{V}_1 \) and \( \mathcal{V}_2 \) that maximizes the weight of the edges between the two parts of the partition. Letting \( x \in \{\pm 1\}^n \) such that \( x_i = 1 \) if node \( i \in \mathcal{V}_1 \) and \( x_i = -1 \) if \( i \in \mathcal{V}_2 \), and denoting the weight of edge \( (i,j) \in \mathcal{E} \) by \( w_{ij} \) (or 0 if the edge is not defined) the problem is
formulated as

\[
\max_x \sum_{1 \leq i < j \leq n} w_{ij} \left( \frac{1 - x_i x_j}{2} \right)
\]

s.t. \( x \in \{\pm 1\}^n \)

Note that each term \( \frac{1 - x_i x_j}{2} \) is 1 if \( x_i \neq x_j \), and 0 otherwise, such that the only weights considered in the objective are those corresponding to edges between the two partitions.

The problem is not convex due to the non-convexity of its domain, but Goemans and Williamson’s algorithm instead solves a problem over a set of vectors \( x_1, \ldots, x_j \in \mathbb{R}^n \) on the unit sphere:

**Problem 1.6** (Lifted max-cut).

\[
\max_x \sum_{1 \leq i < j \leq n} w_{ij} \left( \frac{1 - x_i^T x_j}{2} \right)
\]

s.t. \( x_1, \ldots, x_n \in \mathbb{R}^n \)

\( x_i^T x_i = 1. \)

To establish this problem as a semidefinite program, first define \( X \triangleq [x_1, \ldots, x_n] \), and then \( Y \triangleq X^T X \), such that \( Y(i, j) = x_i^T x_j \). Note also that \( Y \succeq 0 \) (and, in fact, any \( Y \succeq 0 \) admits a decomposition \( Y = B^T B \) for some \( B \) of rank \( \leq n \)). The problem can therefore be rewritten as
Problem 1.7 (Lifted max-cut, alternative).

\[
\max_Y \sum_{1 \leq i < j \leq n} w_{ij} \left( \frac{1 - Y(i,j)}{2} \right)
\]

s.t. \( Y \succeq 0 \)

\( Y(i,i) = 1, \quad \forall i \)

which can readily be converted to the standard SDP form. To recover node labels, Goemans and Williamson propose the following randomized procedure: (1) sample \( r \in \mathbb{R}^n \) uniformly at random on the unit sphere, (2) set label of node \( i \) equal to \( v_i = \text{sign}(r^T x_i) \). They then proceed to prove that, in expectation:

\[
\mathbb{E}[\mu_r^*] > 0.87856 \mu^*
\]

where \( \mu_r^* \) is the cut value produced by the algorithm, whereas \( \mu^* \) is the optimal cut value, thus establishing the approximation power of the algorithm.

1.2.4 Convex relaxations

In its more intuitive formulation on the set \( x_{ij} \in \{0, 1\} \), the shortest path formulation in 1.3 is markedly non-convex. The max-cut problem 1.5, too, is non-convex (and, indeed, NP-hard). In practice, most problems of interest to us are non-convex, and many do not admit polynomial-time solutions.

However, those two examples demonstrated how by relaxing a non-convex domain to an enclosing convex domain can yield a problem that either solves (in the case of shortest path) or approximates (in the case of max-cut) the original problem. This procedure is often denoted convex relaxation.

Let \( \mathcal{P} \) be a non-convex set, and let \( \mathcal{S} \) be a convex set such that \( \mathcal{P} \subset \mathcal{S} \). For convex \( f : \mathcal{S} \to \mathbb{R} \), consider the following two problems
Problem 1.8. (Non-convex problem)

\[
\begin{align*}
\min_x & \quad f(x) \\
\text{s.t.} & \quad x \in P
\end{align*}
\]

and

Problem 1.9. (Convexified problem)

\[
\begin{align*}
\min_x & \quad f(x) \\
\text{s.t.} & \quad x \in S
\end{align*}
\]

It is clear that \( f(x^*_S) \leq f(x^*_P) \), as the constraint in the convexified problem is strictly weaker than that of the non-convex problem. However, if the solution to the convexified problem \( x^*_S \) lies in the non-convex set \( P \), then the inequality above implies that the solutions of the two problems coincide, a feature we call exact recovery.

Broadly speaking, convex relaxation techniques are used to approximate non-convex problems with alternative convex problems whose global solution can be found efficiently. The hope is that the solution to the convex problem closely approximates (or exactly matches) the solution to the original, non-convex problem.

To consolidate this discussion, we discuss another non-convex problem which will play an important role in this thesis: the bipartite weighted matching problem.

Problem 1.10 (Bipartite weighted matching). Let \( \mathcal{G} \) be a bipartite graph with node set \( V = V_1 \cup V_2 \) where \( |V_1| = |V_2| = n \). Let \( w_{ij} \) be the score of matching node \( i \in V_1 \) to node \( j \in V_2 \). We aim to find a perfect matching between nodes in \( V_1 \) and those in \( V_2 \) (that is, a one-to-one matching that maximizes matching score). Expressed as an
optimization problem:

\[
\min_{\{x_{ij}\}} \sum_{ij} w_{ij} x_{ij} \\
\text{s.t.} \quad x_{ij} \in \{0, 1\} \\
\sum_j x_{ij} = 1, \forall i \\
\sum_i x_{ij} = 1, \forall j
\]

Note that the problem is non-convex due to the non-convexity of the first constraint. In particular, we note that this set of constraints defines the set of permutations, \(\text{Perm}(n) \triangleq \{ P \in \{0, 1\}^{n \times n} \mid P1 = 1, P^T1 = 1\}\).

The most straightforward convex relaxation of Problem 1.10 is to relax the non-convex domain to a close convex counterpart. By definition, the smallest convex set that contains a non-convex set \(\mathcal{P}\) is called its **convex hull**. By the Birkhoff-von Neumann theorem \cite{14,63}, the convex hull of the set of permutation matrices is the set of doubly stochastic matrices

\[
\text{DS} = \{ D \in \mathbb{R}^{n \times n} \mid D1 = 1, D^T1 = 1, D \geq 0\}
\]

which is the set of matrices with nonnegative entries whose rows and columns all sum to 1. Making use of this relaxation, we obtain
Problem 1.11 (Relaxed bipartite weighted matching).

\[
\begin{align*}
\min_{\{x_{ij}\}} \quad & \sum_{ij} w_{ij} x_{ij} \\
\text{s.t.} \quad & x_{ij} \geq 0 \\
& \sum_j x_{ij} = 1, \forall i \\
& \sum_i x_{ij} = 1, \forall j
\end{align*}
\]

which can be recognized as a linear program. In fact, the Birkhoff-von Neumann theorem states that the vertices of \( \textbf{DS} \) are precisely the permutation matrices. Therefore, by the same reasoning as in shortest paths, there must exist a vertex solution, which will coincide with the solution to the non-convex problem.

Note that the case where the solution to a convex relaxation exactly matches the solution to the original non-convex problem is not a general occurrence, but, for many problems it appears to happen with high probability in certain regimes (such as with several semidefinite relaxations of maximum-likelihood problems [9]).

Tied with the ease of solving certain convex problems, including LPs and SDPs, convex relaxations have become ubiquitous in optimization literature and have seen use in a wide variety of problems, including max-cut [32], community detection [1], quadratic assignment problem [68], game theory [3], cryo electron microscopy [?], camera location estimation [70], angular synchronization [58], point cloud registration [21], and, indeed, NMR structural determination [44].

1.3 Existing work

The work described in this thesis is certainly not the first to tackle full-automation of the NMR assignment problem. This section offers only a brief overview of the
state-of-the-art in automated assignment, and the reader is invited to explore some of the references contain herein which we will not detail in full.

As early as 2004, a detailed review identified twelve important works on automated NMR assignment [10]. A more recent protocol overview [36] cited 44 works on automated chemical shift assignment, which is still not a complete list. Nearly all of the works cited leveraged a similar pipeline of: (1) registering peaks across different dimensions, (2) spin system construction, (3) fragment building through sequential walking, and finally (4) mapping of fragments through probabilistic typing, where a variety of different techniques have been explored, including exhaustive search [38], best-fit heuristics [69], simulated annealing/monte carlo [39], [46], and genetic algorithms [11], [66], [62].

Among these, a small subset has seen extensive use reported on the protein data bank (PDB, [13]) including AutoAssign [69], CYANA [37], GARANT [11], and PINE [7]. Yet automated assignment software features in only a small fraction of NMR structures deposited in the database, an indication that semi-automated, or interactive assignment tools are still a go-to choice among NMR experts.

One of the challenges faced in the adoption of new automated assignment tools lies in the disparity between the inputs required by each approach. Some older approaches, like MAPPER [38], relied on spin system fragments, while most recent approaches accept either peak lists or grouped peaks in the form of spin systems. However, it is telling that methods such as AutoAssign, which feature an interactive mode and which attempt to replicate the process by which experts assign, even if in a heuristic fashion, are still preferred over alternative black-box methods.

A second major challenge is that of reproducibility and benchmarking. Some important tools, such as CYANA [37], lie behind a pay wall, and only a few of the works on automated assignment have been fully open sourced. Furthermore, individual research teams have adopted tests suites which are unique to them, and for
which data is not publicly available. The work in [64], which described an algorithm
called CISA, attempted to rectify this by introducing a standardized simulated test
suite of spin systems according to empirically accepted experimental error margins.
It tested against three other relevant assignment tools: an iterative, connectivity-
based approach called PACES [25], the random-graph theoretic approach RANDOM
[8], and MARS [40], yet another iterative, connectivity-based method using random
perturbations to nudge current assignments into better ones. An integer programming
approach called IPASS [4] later took up the challenge to test on the same experimental
suite.

Both IPASS and RANDOM share the philosophy of optimizing globally rather
than locally through fragment building, albeit in two very different ways. Another
important algorithm adopting a global approach, FLYA, was proposed in [57], present-
ing state-of-the-art performance through genetic algorithms. The software is available
as part of the CYANA package. Each of these three algorithms shares similarities
with the approaches described in this thesis, so we describe them briefly below.

1.3.1 RANDOM

RANDOM [8] differs from most works on automated NMR assignment by attempting
to develop a provably correct algorithm under strict assumptions. It assumes the
existence of an interaction graph, \( \mathcal{G} \) where the vertices (\( V \)) are compiled spin systems
and the labeled edges (\( E \)) represent inter-residue or intra-residue connectivities derived
through traditional matching methods such as those in [69]. The authors introduce
the concept of sequential fragment as follows

**Definition 1.3** (Sequential fragment ([8, Definition 2])). For an NMR graph \( \mathcal{G} =
(\mathcal{V}, \mathcal{E}) \), a sequential fragment \( F = (\mathcal{V}', \mathcal{E}') \), where \( \mathcal{V}' \subset \mathcal{V} \) and \( \mathcal{E}' \subset \{(u, v) \in \mathcal{E} \mid u, v \in
\mathcal{V}'\} \) is a set of edges supporting that sequential order.

Further
Definition 1.4 (Sequential cover, [8, Definition 3]). For an NMR graph \( G = (V, E) \), a sequential cover \( C = \{ F_1, \ldots, F_{|C|} \} \) is a set of fragments such that each vertex \( v \in V \) appears in exactly one sequential fragment.

Their approach is one of finding a Hamiltonian path in the graph \( G \) by a randomized process, under assumptions of randomness on the graph and with good expected-case performance. Starting with \( C = V \) (i.e. each fragment is an independent vertex) and an empty set of vertices with successors \( W = \emptyset \). After connecting all unambiguous fragments (those with a single outgoing or incoming edge), the Hamiltonian path is extended iteratively by selecting a vertex, \( u \), at random from \( V - W \) and choosing an edge amongst its outgoing edges with probability proportional to a scoring function. The vertices are merged in the cover if the chosen outgoing edge points a fragment with no predecessor in \( C \), otherwise two fragments are created, one connecting \( u \) to the fragment and the other connecting the fragment to its predecessor.

The scoring, like in most other algorithms, relies on distance between observed chemical shifts. However, distance is defined in terms of a sorting order of the observed chemical shifts for each dimension (\(^1\)H, \(^{15}\)N, \(^{13}\)C\(^\alpha\), etc). A degeneracy window \( w \) is defined in terms of this sorting order (distance within the sorted values along an atomic dimension) to state their main result

Theorem 1.1 (8, Theorem 1). Let \( G = (V, E) \) be an NMR graph of degeneracy window \( w \) on \( n \) nodes, with \( d > 1 \) dimensions matched for each edge. Then

1. If \( w = o(n^{1-(3/2d)}) \), then a Hamiltonian path can be found asymptotically almost surely in expected \( O(n^4) \) steps.

2. Let \( w \) be an arbitrary constant (independent of \( n \)). Then, for a suitably large \( n \), with high probability, a Hamiltonian path can be found in expected \( O(n^{4+\log(w-1)}) \) number of steps.
We note that RANDOM does not attempt to produce a full assignment, but rather to determine fragment connectivity, but it offers a good proof of concept of the feasibility of finding connectivities through a global approach (MAPPER [38] is used by the authors to finalize an assignment).

However, missing spin systems or the inclusion of artifact spin systems (both a common occurrence in NMR datasets) mean that the existence of a Hamiltonian path is not guaranteed. If it exists, a Hamiltonian path is not necessarily correct, and probabilistic typing information might be necessary during path-finding in order to bias the selection of the path towards a probabilistically sensible one.

1.3.2 IPASS

Like RANDOM, IPASS builds a connectivity graph from spin systems, but approaches the assignment step by constructing an integer linear program to maximize a probabilistically-inspired score. It also introduces a powerful graph-based method to building spin systems from peak lists of HSQC, HNCACB, and HN(CO)CACB spectra [4], which we also describe as it shares some similarities to a spin enumeration approach we adopt later in this thesis.

Building spin systems IPASS defines two distance functions: one for the root pair ($^{15}$N, $^1$H) and the other including the $^{15}$N dimension. Let $p_1, p_2 \in \mathbb{R}^3$ be peaks from 3D heteronuclear experiments such that $p_i = (N_i, H_i, C_i)$ are the measured chemical shifts of the peak according to the specified dimension ordering.

Definition 1.5 (Peak distances, [4]).

\[
d_{rp}(p_1, p_2) \triangleq \sqrt{(N_1 - N_2)^2 + 10^2(H_1 - H_2)^2}
\]

\[
d(p_1, p_2) = \sqrt{(N_1 - N_2)^2 + 10^2(H_1 - H_2)^2 + (C_1 - C_2)^2}
\]
where the $10^2$ is used to account for the differing experimental sensitivity in the $^1$H dimension. The distance between each two peaks is first computed according to $d_{rp}$, and a nearest neighbor distance is recorded for each peak, $d_{i,NN,rp}$. Directional edges are introduced for between $p_i$ and any peak where $d_{rp}(p_i, p_j) \leq 2d_{NN,rp}$. The resulting graph represents connectivity through the base pair.

A similar procedure is applied through $d(\cdot)$. Any pair of edges of opposing directions between two peaks are replaced by a nondirectional edge and the remaining directional edges discarded. Spin systems, $\{s_1, \ldots, s_m\}$ are finally recovered by brute-force search of consistent $C^\alpha$ and $C^\beta$ values within each connected components of the computed graph.

**Connectivity graph and probabilistic typing**  
As explained in [1.1.2], the choice of HSQC, HNCACB, HN(CO)CACB means that sequential walking is possible along the $C^\alpha$ and $C^\beta$ chains. In IPASS, a connectivity graph, $F$, is established by creating edges between any two spin systems where the $C^\alpha$ and $C^\beta$ connections satisfy a loose threshold $\delta = 0.5$ ppm.

For each edge $(s_i, s_j) \in F$, a heuristic connectivity score $g_{ij}$ is computed based on a weighted sum of three indicator variables evaluating: (1) connectivity in the $C^\alpha$ chain ($\delta' = 0.05$ ppm), (2) connectivity in the $C^\beta$ chain ($\delta' = 0.05$ ppm), and (3) the existence of a cross-peak in the 3D NOESY spectrum between the $^1$H atoms of each residue. Only edges with connectivity scores above a threshold are kept, and the edge set is further trimmed by retaining only the incoming edge with highest connectivity score.

Fragments built from this connectivity graph are statistically typed against every position of the target sequence. That is, a likelihood is computed based on a multivariate Gaussian assumption on the generative distribution of the chemical shifts
within each residue, given statistics collected from [13]. Only mappings satisfying a threshold condition are retained as potential matches.

**Assignment** All combinations of fragments where no spin system appears in more than one fragment, and where no fragment’s valid mappings overlap are enumerated. An integer linear program is then constructed on each such combination in order to compute a best assignment for that combination.

Let \( s_i \) and \( s_j \) be two connected spin systems. The utility of assigning \( s_i \) and \( s_j \) to residues \( r_k \) and \( r_{k+1} \) respectively is denoted by

\[
w_{i,j,k} = \log \Pr [r_k | s_i] + \log \Pr [r_{k+1} | s_j].
\]

The ILP is designed to maximize the sum of the above quantities over all possible matchings within one particular fragment combination.

### 1.3.3 FLYA

Unlike IPASS, FLYA attempts to optimize a global score directly from peak lists without the intervening steps of spin system construction or fragment building [57]. Instead, given a set of measured peak lists, FLYA compares it directly to a hypothetical set of peak lists which one would expect to observe given the NMR experiments that were carried out. Expected peaks are matched to measured peaks with the goal of maximizing the global score, and chemical shift values are inferred from this matching.

The global score is constructed based on probabilistic assumptions about the generative distributions of individual chemical shifts and of the experimental noise. It makes use of two quality measures, \( Q_1 \) and \( Q_2 \) which we paraphrase here:
**Definition 1.6** \((Q_1 [57])\). \(Q_1(a)\) measures agreement between the average frequency assigned to atom \(a\) by the matching between expected and measured peaks. It is defined in terms of a Gaussian prior and is given by

\[
Q_1(a) = 1 + \frac{\log \left( 1 - \text{erf} \left( \frac{x_1}{\sqrt{2}} \right) \right)}{\log \left( 1 - \text{erf} \left( \frac{1.5}{\sqrt{2}} \right) \right)}
\]

where

\[x_1 \triangleq \frac{f(a) - \mu_a}{\sigma_a}\]

where \(f(a)\) is the average frequency assigned to \(a\) and \(\mu_a\) and \(\sigma_a\) are the prior distribution parameters for atom \(a\). Note that the quantity inside the log term evaluates the area under a standard normal distribution lying outside a tail of \(x_1\) standard deviations. Therefore, the score will be positive for atoms whose assigned frequencies lies inside 1.5 standard deviations of the prior mean, and negative otherwise.

**Definition 1.7** \((Q_2 [57])\). \(Q_2(a, e)\) measures agreement between the frequency assigned to \(a\) by expected peak \(e\) with regards to the average assigned frequency. It is defined analogously to \(Q_1\)

\[
Q_2(a) = 1 + \frac{\log \left( 1 - \text{erf} \left( \frac{x_2}{\sqrt{2}} \right) \right)}{\log \left( 1 - \text{erf} \left( \frac{2}{\sqrt{2}} \right) \right)}
\]

where

\[x_2 \triangleq \frac{f(a, e) - f(a)}{\varepsilon_a / 4}\]

Where \(\varepsilon_a\) is the accepted experimental tolerance for atom \(a\).

Let \(A^o\) be the set of atoms for which peaks are expected to exist. Let \(A\) be the set of atoms which have been assigned at least one observed value by the algorithm. Let \(E^o_a\) be the set of expected peaks in which atom \(a\) is featured and \(E_a\). The global
score is defined as

\[ G \triangleq \frac{\sum_{a \in A} \left( 4Q_1(a) + \sum_{e \in E_a} \frac{Q(a,e)}{b(e)} \right)}{\sum_{a \in A^o} (4 + |E_a|)} \]

where \( b(e) \) equals the number of expected peaks mapped to the same measured peak as peak \( e \).

**Optimization** The optimization of global score \( G \) is extremely difficult due to the combinatorial nature of the mapping between expected and measured peaks. The authors of FLYA thereby combine heuristic local optimization procedures – which remap local regions of the assignment – with a genetic algorithm which probabilistically recombines and mixes existing assignments from generation to generation, with state-of-the-art results.

### 1.4 A note on generative assumptions

While probabilistic assumptions are commonly made across automated assignment tools, they do not always coincide. In particular, CISA \[64] and, by extension, IPASS \[4] generate simulated data by adding white noise chemical shifts at the spin system level. FLYA, on the other hand, assumes truncated Gaussian noise on measured peaks to ensure valid assignments are possible under their evaluation framework \[57]. As a result, great care is required when comparing different algorithms. We will describe our data generation procedure in full for each test described in this thesis.

### 1.5 Our contributions

This thesis attempts to address some of the challenges and limitations faced by automated assignment algorithms. In particular,
• We propose an alternative formulation of maximum-likelihood spin system assignment as a quadratic assignment problem (QAP) and develop a sparsity-exploiting semidefinite relaxation of the problem (Chapter 2);

• We develop a carefully-crafted ADMM scheme for efficiently solving our proposed relaxation, and evaluate solution quality both in the context of NMR assignment and more broadly in the context of QAP (Chapter 2);

• We propose a consolidated probabilistic model for assignment and adopt a linear programming relaxation approach to develop a maximum-likelihood solver for assignment based on either spin systems or peak lists directly (Chapter 3).

Chapter 2 will focus on our attempts to leverage semidefinite programming relaxations for NMR assignment. Building upon the work of Zhao et al [68] we develop a convex relaxation for the particular case of quadratic assignment with at least one sparse graph to achieve a problem size of $O(n^3)$ in the number of nodes, which we term C-SDP.

We further demonstrate how the provably convergent ADMM algorithm in [60] can yield an efficient distributed algorithm for the C-SDP relaxation, through a wise grouping of the dual variables. After careful implementation, we compare the recovery results against similar-scaling convex relaxations for the quadratic assignment problem, illustrating that C-SDP achieves strong bounds on many of these problems, and can often recover good integral solutions after projection.

In keeping with our original goal, we evaluate CSDP on a subset of the benchmark test suite developed in [64], where state-of-the-art performance is achieved. However, that performance fails to generalize to lower quality datasets, and does not scale sufficiently well for the efficiency assignment of large proteins, prompting the need for a different approach.
Chapter 3 focuses instead on a more straightforward linear programming relaxation to solve maximum-likelihood assignment. We describe a flexible framework for the probabilistic model of the assignment graph, and a generalized linear programming framework for achieving fast solutions for both spin systems and peak lists in medium and large proteins, using off-the-shelf solvers and a single desktop computer.

The performance of our proposed algorithm is demonstrated on both the CISA dataset, and on spin system data provided by the authors of IPASS, with state-of-the-art results on both. We also evaluate our algorithm under the simulation framework of FLYA, where preliminary results demonstrate comparable performance with little to no tuning, in what constitutes promising progress.

Ultimately, one of the main goals in writing this thesis has been to describe the problem of NMR assignment in a way that’s easily understandable to applied mathematicians and computer scientists. All code produced for this work is intended to be shared. Much of it is already online (and is linked to in the relevant section of the thesis) or will be publicly available very soon, and can be shared with interested readers ahead of the release date.
Chapter 2

Assignment through Semidefinite Programming

Viewed from the perspective of reordering of spin systems, backbone assignment can be understood as an attempt to minimize the total discrepancy along the $^{13}\text{C}_\alpha$, $^{13}\text{C}_\beta$, and $^{13}\text{CO}$ chains for the reordered spins.

The source of discrepancy in an optimal ordering is due to experimental noise across different NMR spectra, which implies that chemical shifts for atom $i$ will not exactly match across different experiments. Let $f_i$ be the true chemical shift of atom $i$. A common assumption adopted in the field (see e.g. [57], [4], [37], [64]) is that the chemical shift of atom $i$ measured on spectrum $S$, $f_i^S$, is distributed according to

$$f_i^S \sim \mathcal{N}(f_i, \sigma_i^2)$$ (2.1)

where $\sigma_i$ is on the same order of magnitude as the experimental tolerance of atom $i$ ($\sim 0.03$ppm for hydrogen, $\sim 0.2 - 0.4$ppm for carbon and nitrogen).

Let $s_1, \ldots, s_n$ denote spin systems constructed from heteronuclear experiments. Under the given noise model, a straightforward measure of discrepancy between two spin systems, $s_i \rightarrow s_j$ might be
**Definition 2.1** (Discrepancy between spin systems $s_i \rightarrow s_j$).

\[ d(s_i, s_j) \triangleq \frac{(s_j(C_{\alpha}^0) - s_i(C_{\alpha}^0))^2}{2\sigma_{\alpha}^2} + \frac{(s_j(C_{\beta}^0) - s_i(C_{\beta}^0))^2}{2\sigma_{\beta}^2} + \frac{(s_j(C_O^0) - s_i(C_O^0))^2}{2\sigma_{O}^2} \quad (2.2) \]

which is directed, and proportional to $-\log \Pr [s_i \rightarrow s_j]$ (the probability that spin system $s_i$ precedes $s_j$). Assume for now that the target protein has exactly $n$ residues, and that we believe the $n$ spin systems available are accurate (i.e. nothing could be more ideal). Then we wish to reorder the spin systems according to some permutation $\pi$, such that

\[ \text{discrepancy} \triangleq \sum_{i=1}^{n-1} d(s_{\pi(i)}, s_{\pi(i+1)}) \quad (2.3) \]

is minimized. This problem is a particular instance of the quadratic assignment problem (QAP) \[45\], which was originally defined as

**Problem 2.1** (Quadratic Assignment Problem, QAP \[45\]).

\[
\min_{\pi} \sum_{i=1}^{n} \sum_{j=1}^{n} B(i, j) A(\pi(i), \pi(j)) \\
\text{s.t.} \quad \pi \text{ is a permutation.}
\]

An intuitive interpretation of the cost is obtained in terms of optimal transportation of supplies between factories. Let $B(i, j)$ be the weight of supplies that must be transported between factories $i$ and $j$. Factories can be assigned to any of $n$ locations, and the cost of transporting supplies between two locations is proportional to the weight of the supplies and to the distance to be traveled. Let $A(k, l)$ be the distance between locations $k$ and $l$. The QAP as formulated above solves the problem of assigning factories to locations so as to minimize the total transportation cost.
The assignment problem, under the perspective adopted in this chapter, is simply a variant of the QAP where a single unit of supplies has to be transported in a simple path that visits every factory (i.e. every residue) exactly once.

This chapter discusses the applicability of a novel semidefinite programming (SDP) relaxation to the problem of NMR assignment. Section 2.1 discusses related work in the context of the QAP and SDP relaxations, section 2.2 introduces the CSDP relaxation, and section 2.3 derives a careful alternating direction method of multipliers (ADMM) algorithm for CSDP. The remainder of the chapter is dedicated to evaluation, on both NMR 2.5 and other benchmarks 2.4.

Notation

The notation in the chapter is more intricate than one would like, so we summarize the main points before kicking off our discussion of the QAP.

Capitalized Roman letters, such as $A$, represent matrices, while their lower case equivalents stand for the corresponding column-wise vectorization, i.e. $a := \text{vec}(A)$. The symbol $\otimes$ is used to denote the Kronecker product. $\Pi_K$ represents a projection into the convex space $K$. $\text{Perm}(n)$ denotes the set of permutation matrices of dimension $n$ while $\text{DS}(n)$ denotes the set of doubly stochastic matrices. $I_n$ is the identity matrix of dimension $n$, $J_n$ is the all ones matrix of dimension $n$ and $1_n$ is the all ones column vector of length $n$. For a matrix $Q$, the notation $Q_{ij}$ denotes the $(i, j)$-th block of the matrix (whose size should be clear from context), while $Q(i, j)$ denotes the $(i, j)$-th entry. Column $i$ of matrix $P$ is denoted by $p_i$, and $v(i)$ is used to indicate the $i$-th entry of vector $v$. Finally, we use $\delta_{ij}$ to denote the Kronecker-delta.
2.1 Related Work

Introduced by Koopmans and Beckmann in their 1957 paper [45], the QAP is an NP-hard problem (the \( \epsilon \)-approximation problem is also NP-hard [56]) which has been the subject of extensive work, as it encapsulates a variety of interesting combinatorial problems, such as the traveling salesman problem (TSP), the max-clique problem, among many others (we refer the interested reader to [50] for many interesting applications of the QAP).

Letting \( P \in \text{Perm}(n) \) be defined as

\[
P \in \{0, 1\}^{n \times n} \quad \text{s.t.} \quad P(i, j) = \begin{cases} 
1 & \text{if } \pi(i) = j \\
0 & \text{otherwise}
\end{cases}
\]

the problem admits a simple matrix formulation

**Problem 2.2** (QAP, matrix form).

\[
\min_P \quad \text{Tr} \left( B^T P A P^T \right) \\
\text{s.t.} \quad P \in \text{Perm}(n).
\]

Regardless of the formulation, the QAP is an extremely challenging problem where \( n = 25 \) represents a computational challenge to this day [19]. However, its quadratic nature invites semidefinite programming relaxations, and the seminal work of Zhao et al [68] introduced a remarkably tight relaxation which can exactly recover solutions to many problem instances in the QAPLIB (a library of hard QAP instances, [19]). Close variations of this relaxation have already seen use in shape matching [43].

The strength of the relaxation comes at the cost of introducing a positive semidefinite (PSD) variable of size \( O(n^2) \times O(n^2) \), such that it cannot readily be tackled by interior point methods. An ADMM-based solver for the proposed relaxation,
presented in [29], greatly extends the usability of the relaxation to larger problems (while achieving new improved bounds on a variety of QAPLIB problems previously inaccessible). However, problems with \( n = 100 \) remain far out of reach.

A variety of SDP relaxations with PSD variable size of order \( \mathcal{O}(n) \times \mathcal{O}(n) \) have already been proposed, hoping to leverage semidefinite programming in larger problems. The matrix-lifting approach of Ding and Wolkowicz [28] is one such relaxation, with order \( \mathcal{O}(n^2) \) variables and constraints, leveraging a matrix lifting \( Y \succeq XX^T \) as opposed to the vector lifting \( Q \succeq \text{vec}(x)\text{vec}(x)^T \) of the canonical relaxation, and achieving very strong bounds, although interior point methods are still not scalable to large problem instances.

The QPB algorithm of Anstreicher and Brixius [6], on the other hand, utilizes a quadratic programming approach to obtain strong bounds at low computational cost, and has been applied to problems of size \( n \geq 100 \). Other SDP relaxations have since been utilized at this scale. The work in [52] and [53] uses a PSD-splitting of the data matrix \( B \) to obtain a tractable relaxation, while the eigenspace relaxation in [27] and [26] leverages the spectral decomposition of \( PBP^T \) for the same purpose.

Our proposed relaxation also leverages the structure of \( B \) – in our case its sparsity. The rationale for this approach is due to the fact that for our description of NMR assignment, \( B \) would simply be the adjacency matrix of the path graph

\[
B = \begin{pmatrix}
0 & 1 & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & 1 & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & 0
\end{pmatrix}
\]

with \( n - 1 \) nonzero entries.
2.2 CSDP - a convex relaxation

We begin this section by introducing the canonical relaxation of Zhao et al, which lays the foundation for the work presented in this chapter.

2.2.1 SDP relaxation of Zhao et al

Making use of the cyclic properties of the trace and of the vectorization identity vec\((AYB) = (B^T \otimes A)\text{vec}(Y)\), one can rewrite the QAP objective as follows:

\[
\text{Tr} \left( PP^T A^T \right) = \text{Tr} \left( P^T APB^T \right) = \text{vec}(P)^T \text{vec}(APB^T) = \text{vec}(P)^T (B \otimes A) \text{vec}(P) = \text{Tr} \left( (B \otimes A) \text{vec}(P) \text{vec}(P)^T \right).
\]

The QAP can therefore be reformulated as

**Problem 2.3** (QAP, lifted).

\[
\max_{Q,P} \quad \text{Tr} \left( (B \otimes A)Q \right) \\
\text{s.t.} \quad P \in \text{Perm}(n) \\
\quad Q = \text{vec}(P) \text{vec}(P)^T.
\]

Note that the constraints on \(P\) and \(Q\) are both nonconvex. Recall from the discussion in [1.2.4] that \(\text{Perm}(n)\) has the set of doubly stochastic matrices as its convex hull. Relaxing \(P\) accordingly, one can also replace the nonconvex constraint on \(Q\) with \(Q - \text{vec}(P) \text{vec}(P)^T \succeq 0\), which, by the Schur complement, is equivalent
to:

\[
\begin{pmatrix}
Q & \text{vec}(P) \\
\text{vec}(P)^T & 1
\end{pmatrix} \succeq 0.
\]

Note that, in its unrelaxed state, each \( n \times n \) block \( Q_{ij} \) of \( Q \) is the outer product of two columns of a permutation matrix. This induces a number of linear constraints allowing one to arrive at the following relaxation

**Problem 2.4 (SDP relaxation [68]).**

\[
\max_{Q,P} \quad \text{Tr} ((B \otimes A)Q)
\]

\[
\text{s.t.} \quad \begin{pmatrix}
Q & \text{vec}(P) \\
\text{vec}(P)^T & 1
\end{pmatrix} \succeq 0, \quad (2.5)
\]

\[
\sum_i Q_{ii} = I_n, \quad (2.6)
\]

\[
\text{Tr} (Q_{ij}) = 0, \ i \neq j, \ i,j = 1, \ldots, n, \quad (2.7)
\]

\[
\text{Tr} (Q_{ij} J_n) = 1, \ i,j = 1, \ldots, n, \quad (2.8)
\]

\[
Q_{ii}(j,j) = P(j,i), \ i = 1, \ldots, n, \quad (2.9)
\]

\[
P \in \text{DS}(n), \ i = 1, \ldots, n, \quad (2.10)
\]

\[
Q_{ij} \geq 0, \ i,j = 1, \ldots, n. \quad (2.11)
\]

Constraint 2.6 arises from the fact that each diagonal block of the unrelaxed \( Q \) is the outer product of one of the columns of the permutation matrix with itself, such that it must have a single 1 on its diagonal. Constraints 2.7 and 2.8 arise from the orthogonality of the columns of a permutation matrix. Constraint 2.9 follows from the fact that the diagonal of \( Q \) corresponds to the squared terms of the permutation, which are either 0 or 1, such that \( Q_{ii}(j,j) = P(j,i)^2 = P(j,i) \).

Our main observation is that, in many instances of interest, such as the traveling salesman problem, and shortest simple path (or NMR assignment), the data matrix
Let $B$ represent the adjacency matrix of a connected graph, $G_B$, with $O(n)$ edges. Let $E_B$ be the edge set of $G_B$. The QAP cost can be decomposed as

$$\text{Tr}((B \otimes A)Q) = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{Tr}(B(i,j)A^T Q_{ij}) = \sum_{(i,j) \in E(B)} \text{Tr}(B(i,j)A^T Q_{ij}). \quad (2.12)$$

In light of this, we propose the following relaxation

**Problem 2.5 (E-SDP relaxation [31]).**

$$\max_{\{Q_{ij}\}, P} \sum_{(i,j) \in E(B)} \text{Tr}(B(i,j)A^T Q_{ij})$$

s.t.

$$\begin{pmatrix} Q_{ii} & Q_{ij} & p_i \\ Q_{ij}^T & Q_{jj} & p_j \\ p_i^T & p_j^T & 1 \end{pmatrix} \succeq 0, \forall i, j,$$

$$\text{Tr}(Q_{ij}) = 0, \ i \neq j,$$

$$\text{Tr}(Q_{ij}(J_n - I_n)) = 1, \ i \neq j,$$

$$\text{Tr}(Q_{ii}) = 1, \forall i,$$

$$\text{Tr}(Q_{ii}J_n) = 0, \forall i,$$

$$Q_{ii}(j,j) = P(j,i), \forall i, j,$$

$$P \in \text{DS}(n),$$

$$Q_{ij} \geq 0, \forall i, j,$$

where we remind the reader that $p_i$ denotes the $i$-th column of $P$. 

40
For a large graph $B$ with $O(n)$ edges, we see that this relaxation has on the order of $O(n^3)$ variables. To achieve this reduction in problem size we sacrifice positive semidefiniteness of $Q$, and instead enforce only positive semidefiniteness of submatrices of $Q$, so this relaxation is dominated by that in [68].

### 2.2.3 C-SDP relaxation

The relaxation can be strengthened in a straightforward way by considering cliques, or any other decomposition of the nodes of the graph into connected components of arbitrary size. Let $r$ be such a component, with nodes $V_r = \{V_r(1), \ldots, V_r(|V_r|)\}$. To simplify notation down the line, and because it is important in writing the ADMM formulation in section 2.3, we introduce the variable $X_r$ defined as follows:

$$X_r = \begin{pmatrix}
Q_{V_r(1)V_r(1)} & Q_{V_r(1)V_r(2)} & \cdots & Q_{V_r(1)V_r(|V_r|)} & p_{V_r(1)} \\
Q_{V_r(2)V_r(1)} & Q_{V_r(2)V_r(2)} & \cdots & Q_{V_r(2)V_r(|V_r|)} & p_{V_r(2)} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
Q_{V_r(|V_r|)V_r(1)} & Q_{V_r(|V_r|)V_r(2)} & \cdots & Q_{V_r(|V_r|)V_r(|V_r|)} & p_{V_r(|V_r|)} \\
p_{V_r(1)}^T & p_{V_r(2)}^T & \cdots & p_{V_r(|V_r|)}^T & 1
\end{pmatrix} \succeq 0. \quad (2.13)$$

An appropriate choice of cost matrices $C_r$, and a variable $X_r$ for each component in a connected decomposition of the nodes in $B$, the QAP objective can be rewritten as:

$$\text{Tr} \ ((B \otimes A)Q) = \sum_r \text{Tr} \ ((C_rX_r). \quad (2.14)$$

Most constraints in the E-SDP extend straightforwardly to the case of arbitrary components. An additional constraint arises from the fact that components of size $> 2$ may share more than a single node, such that equality between different submatrices of the variables $\{X_r\}$ must be enforced also on offdiagonal (and thus dense) blocks. This challenge is illustrated with an example in section 2.2.4 below.
2.2.4 Illustrative example using the path graph

Let $B$ be the adjacency matrix for the path graph, given by

$$
B = \begin{pmatrix}
0 & 1 & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & 1 \\
0 & \cdots & \cdots & 0
\end{pmatrix}
$$

(2.15)

In this case, the cliques of the graph correspond to the $n - 1$ edges of $B$. As a result, we will have $n - 1$ variables of the form

$$
X_{i(i+1)} = \begin{pmatrix}
Q_{ii} & Q_{i(i+1)} & p_i \\
Q_{(i+1)i} & Q_{(i+1)(i+1)} & p_{i+1} \\
p_i^T & p_{i+1}^T & 1
\end{pmatrix}.
$$

If we consider an example with 5 nodes and look at the matrix $Q = \text{vec}(P)\text{vec}(P)^T$, we see that the E-SDP variables include the $2n \times 2n$ blocks along the diagonal, as illustrated in Figure 2.1a.

![Figure 2.1: E-SDP and C-SDP variables in the path graph problem with 5 nodes.](image)

(a) E-SDP with cliques of size 2. (b) C-SDP with cliques of size 3.

Figure 2.1: E-SDP and C-SDP variables in the path graph problem with 5 nodes.
Several of the diagonal blocks of $Q$, highlighted in blue, overlap between adjacent variables, and thus it is necessary to enforce these equality constraints. Fortunately, the diagonal blocks of $Q$ are diagonal themselves, such that $n$ equality constraints are sufficient to enforce equality between two blocks.

We draw attention to the fact that when using components of size greater than 2 the C-SDP variables $X_r$ will overlap in off-diagonal blocks (as illustrated in Figure 2.1b). These are less convenient to handle computationally (as they are generally dense), but, in practice, we have observed that enforcing equalities only between diagonal blocks sacrifices little in terms of performance, while greatly reducing the computational burden, as we shall see from the ADMM scheme in section 2.3 below.

### 2.3 Solving C-SDP with ADMM

This section describes an ADMM scheme devised for the C-SDP relaxation that allows for a distributed solution. For ease of exposition, we focus on describing the updates in each iteration on E-SDP, but they extend to C-SDP in a straightforward manner.

**Note on convention:** Since the variables $X_{ij}$ are symmetric, it is equivalent to consider $X_{ij}$ or $X_{ji}$. Therefore, we define graph $G_B$ with adjacency matrix, $\tilde{B}$ defined as

$$\tilde{B} = \text{triu}(B + B^T)$$

(2.16)

where triu$(M)$ extracts the upper triangular portion of matrix $M$. Thus, in all that follows, the variables $X_{ij}$ will be defined according to the edges specified by $\tilde{B}$, such that $i < j$. 


2.3.1 Rewriting constraints in E-SDP relaxation

As discussed in 2.2.2, the E-SDP objective can be written in terms of the variables $X_{ij}$ and cost matrices $C_{ij}$.

It remains to rewrite the constraints in terms of these variables. We remind the reader that the variables under consideration are

$$X_{ij} = \begin{pmatrix} Q_{ii} & Q_{ij} & p_i \\ Q_{ij}^T & Q_{jj} & p_j \\ p_i^T & p_j^T & 1 \end{pmatrix}, \quad (i, j) \in E(G_{\tilde{B}}) \quad (2.17)$$

where all $X_{ij}$’s are non-negative and PSD. Going forward, all the constraints will be rewritten in terms of the variables $X_{ij}$ and the variables $Q_{ij}$ and $p_i$ will no longer be used.

The first set of constraints on $X_{ij}$ follows directly from the constraints on $Q$ and $P$, which are the following:

- $\operatorname{Tr}(Q_{ii}) = \operatorname{Tr}(Q_{jj}) = 1,\quad (2.18)$
- $\operatorname{Tr}(Q_{ii}J_n) = \operatorname{Tr}(Q_{jj}J_n) = 0,\quad (2.19)$
- $\operatorname{Tr}(Q_{ij}) = 0,\quad (2.20)$
- $\operatorname{Tr}(Q_{ij}J_n) = 1,\quad (2.21)$
- $\operatorname{diag}(Q_{ii}) = p_i,\quad (2.22)$
- $\operatorname{diag}(Q_{jj}) = p_j.\quad (2.23)$

Letting $x_{ij} := \text{vec}(X_{ij})$, all the constraints above can be written in the form:

$$\mathcal{A}x_{ij} = b_E \quad (2.24)$$

where $\mathcal{A} \in \mathbb{R}^{m_E \times (2n+1)^2}$, $b_E \in \mathbb{R}^{m_E}$, and $m_E$ is the number of equality constraints.
Note that the $X_{ij}$'s are not independent of each other. Firstly, for the edges that are incident on the same node, the associated variables $X_{ij}$'s share a common $n \times n$ block on the diagonal. This is illustrated in the example of a path graph in section 2.2.4. Therefore, equality constraints between the overlapping diagonal blocks of $X_{ij}$'s have to be enforced. Since $\text{Tr} (Q_{ii}(J_n - I_n)) = \text{Tr} (Q_{jj}(J_n - I_n)) = 0$ and $Q_{ii}, Q_{jj} \geq 0$, the off-diagonal terms of $Q_{ii}$ and $Q_{jj}$ are zeros and it suffices to enforce equality of the diagonals. Further, since $p_i$ and $p_j$ equal the diagonals of $Q_{ii}$ and $Q_{jj}$, one can enforce consistency of the overlapping blocks by looking at the last row and column of each $X_{ij}$ instead. Consider the sampling matrices $B_1$ and $B_2$, which sample $p_i$ and $p_j$ from the vector $x_{ij}$ above. If $(i, j)(k, i) \in E(G_B)$, then a consistency relationship of the form
\[ B_1 x_{ij} = B_2 x_{ki} \] (2.25)
must hold.

Adding the conic constraints for positivity and positive semi-definiteness, the E-SDP relaxation can be reformulated as:

**Problem 2.6** (E-SDP, version 2).

\[
\begin{aligned}
\max_{\{x_{ij}\}} & \quad \sum_{(i,j) \in E(G_B)} c^T_{ij} x_{ij} \\
n\text{s.t.} & \quad Ax_{ij} = b_e, \\
& \quad B_1 x_{ij} = B_2 x_{ki} \forall (i, j)(k, i) \in E(G_B), \ i = 1, \ldots, n, \\
& \quad B_2 x_{kn} + \sum_{i=1}^{n-1} B_1 x_{ij} = 1, (i, j), (k, n) \in E(G_B) \\
& \quad x_{ij} \geq 0, \\
& \quad D x_{ij} \geq 0.
\end{aligned}
\]

45
Where through a slight abuse of notation we used \( x_{ij} \geq 0 \) to denote \( X_{ij} \succeq 0 \). We have also used \( c_{ij} \equiv \text{vec}(C_{ij}) \). The third constraint states that the sum of the diagonal blocks of \( Q \) equal the identity. The matrix \( D \) is of size \( 4n^2 \times (2n+1)^2 \) and it samples all elements of \( x_{ij} \) except for those corresponding to the last row and column of \( X_{ij} \). The reason for this sampling is two-fold:

1. Sampling the last row and column is unnecessary, since these entries are implicitly defined by the linear constraints covered in equations 2.22 and 2.23.

2. Using a sampling operator of this form ensures mutual orthogonality between \( D, B_1 \) and \( B_2 \),

\[
B_1 B_2^T = 0_{n \times n}, \quad B_1 D^T = 0_{n \times 4n^2}, \quad B_2 D^T = 0_{n \times 4n^2}, \quad (2.26)
\]

which shall prove crucial in obtaining fast ADMM updates involving a least-squares problem with a block-diagonalized Hessian.

In order to derive a fast ADMM routine to solve Problem 2.6, slack variables are introduced. Let \( N_i \) be the one-hop neighborhood of node \( i \) on graph \( \tilde{G}_B \). Then the consistency relation in equation 2.25 can be enforced by introducing slack variables \( p_i \), such that

\[
B_1 x_{ij} = p_i, \; \forall j \in N_i \quad (2.27)
\]

\[
B_2 x_{ij} = p_j, \; \forall j \in N_i. \quad (2.28)
\]

As a result, our problem can finally be written in the form
Problem 2.7 (E-SDP, version 3).

\[
\max_{\{x_{ij}\}, \{p_i\}} \sum_{(i,j) \in E(G)} c_{ij}^T x_{ij}
\]

s.t. \[y_{ij} : Ax_{ij} = b_e,\]
\[w_{ij}^{(1)} : B_1 x_{ij} = p_i \forall j \in N_i, \ i = 1, \ldots, n,\]
\[w_{ij}^{(2)} : B_2 x_{ij} = p_j \forall j \in N_i, \ i = 1, \ldots, n,\]
\[t : \sum_i p_i = 1,\]
\[s_{ij} \geq 0 : x_{ij} \geq 0,\]
\[z_{ij} \geq 0 : Dx_{ij} \geq 0\]

where the variable in front of each colon is the dual variable tied to the corresponding constraint. The constraint \(\sum_{i=1}^{n} p_i = 1\) couples the \(X_{ij}\) from different blocks together.

**Generalization to C-SDP**: One can generalize the presentation above to the C-SDP relaxation in a straightforward way. The one important difference is that for general sets of nodes of size greater than two, the corresponding \(X_{r}\)’s as defined by equation 2.13 might overlap in non-diagonal blocks (if two or more nodes are shared by two cliques). Figure 2.1b highlights the fact that one must enforce equalities between \(X_{ijk}\) and \(X_{ijl}\) not only for \(Q_{ii}\) and \(Q_{jj}\), but also for \(Q_{ij}\). However, we have observed that enforcing only the equalities on the diagonal blocks produces solutions which are nearly as good with a much decreased computational cost, so we adopt this approach for all of our evaluations.
2.3.2 Duality problem and the ADMM updates

We now turn to the dual problem of the E-SDP relaxation presented in the form of problem 2.7. In this section we show that through an appropriate grouping of the dual variables the ADMM updates for solving the dual problem can be computed in a distributed manner.

The dual of problem 2.7 is the following:

**Problem 2.8 (Dual E-SDP).**

\[
\begin{align*}
\min_{y_{ij}, w_{ij}^{(k)}, t, s_{ij}, z_{ij}} & \quad \sum_{(i,j) \in E(G_B)} b^T y_{ij} - 1^T t \\
\text{s.t.} & \quad s_{ij} \geq 0, (i, j) \in E(G_B) \\
& \quad z_{ij} \geq 0, (i, j) \in E(G_B) \\
& \quad x_{ij} : -c_{ij} + s_{ij} + D^T z_{ij} + A^T y_{ij} + B^T_1 w_{ij}^{(1)} + B^T_2 w_{ij}^{(2)} = 0, (i, j) \in E(G_B) \\
& \quad g_i : t - \sum_{j \in N_i} w_{ij}^{(1)} - \sum_{j : i \in N_j} w_{ji}^{(2)} = 0, i = 1, \ldots, n.
\end{align*}
\]

Using \( \delta_K(x) \) to denote a function that takes the value \( +\infty \) for \( x \notin K \) and 0 otherwise, the augmented Lagrangian is

\[
\mathcal{L} = \sum_{(i,j) \in E(G_B)} \left( \delta_{S_B^+}(s_{ij}) + \delta_{K_p}(z_{ij}) - b^T y_{ij} \right) - 1^T t +
\]

\[
\frac{\rho}{2} \sum_{(i,j) \in E(G_B)} \left\| -c_{ij} + s_{ij} + D^T z_{ij} + A^T y_{ij} + B^T_1 w_{ij}^{(1)} + B^T_2 w_{ij}^{(2)} + \frac{x_{ij}}{\rho} \right\|^2 +
\]

\[
\frac{\rho}{2} \sum_{i=1}^{n} \left\| t - \sum_{j \in N_i} w_{ij}^{(1)} - \sum_{j : i \in N_j} w_{ji}^{(2)} + \frac{g_i}{\rho} \right\|^2
\]

where \( x_{ij} \) and \( g_i \) are now the dual variables of the dual problem, and \( \rho \) is some constant greater than 0.
In [60], a convergent ADMM is proposed to solve optimization problems with a 3-block structure where one of the blocks only involves linear operators. In our problem, we let the three blocks be defined by the groups of variables \((s_{ij}, t), (y_{ij})\) and \((z_{ij}, w_{ij}^{(k)})\). The algorithm is described below as Algorithm 1. Note that our problem falls into this setting thanks to the orthogonality between \(D, B_1\) and \(B_2\), but recent developments in ADMM [47] [23] would allow for schemes with additional blocks and offer an alternative to these orthogonality requirements.

**Algorithm 1 Conic-ADMM3c [60]**

**Require:** \(\rho > 0\) and \(\tau = 1.618\)

for \(l = 1, \ldots, \text{MAXIT}\) do

\[
(s_{ij}, t)^{l+1} \leftarrow \arg\min_{s_{ij}, t} \mathcal{L}(s_{ij}, t, y_{ij}, z_{ij}, w_{ij}^{(k)}, x_{ij}^l, g_{ij}^l; \rho)
\]

\[
(y_{ij})^{l+1/2} \leftarrow \arg\min_{y_{ij}} \mathcal{L}(s_{ij}^{l+1/2}, t^{l+1/2}, y_{ij}, z_{ij}, w_{ij}^{(k)}); x_{ij}^l, g_{ij}^l; \rho)
\]

\[
(z_{ij}, w_{ij}^{(k)})^{l+1} \leftarrow \arg\min_{z_{ij}, w_{ij}^{(k)}} \mathcal{L}(s_{ij}^{l+1/2}, t^{l+1/2}, y_{ij}^{l+1/2}, z_{ij}, w_{ij}^{(k)}); x_{ij}^l, g_{ij}^l; \rho)
\]

\[
(y_{ij})^{l+1} \leftarrow \arg\min_{y_{ij}} \mathcal{L}(s_{ij}^{l+1}, t^{l+1}, y_{ij}, z_{ij}^{l+1}, w_{ij}^{(k)}); x_{ij}^{l+1}, g_{ij}^l; \rho)
\]

\[
x_{ij}^{l+1} \leftarrow x_{ij}^l + \tau \arg\min_{x_{ij}} \mathcal{L}(s_{ij}^{l+1}, t^{l+1}, y_{ij}^{l+1}, z_{ij}^{l+1}, w_{ij}^{(k)}); x_{ij}, g_{ij}^l; \rho)
\]

\[
g_{ij}^{l+1} \leftarrow g_{ij}^l + \tau \arg\min_{g_{ij}} \mathcal{L}(s_{ij}^{l+1}, t^{l+1}, y_{ij}^{l+1}, z_{ij}^{l+1}, w_{ij}^{(k)}); x_{ij}^{l+1}, g_{ij}; \rho)
\]

end for

In the remainder of this section, we will derive each of the updates in turn and illustrate how this choice of variable groupings allows for easy parallelization.

**Update for \((s_{ij}, t)\):** The updates for \(s_{ij}\) and for \(t\) are independent. The update for \(t\) is given by the solution to a least-squares problem:

\[
\arg\min_t \mathcal{L} = \frac{1}{n} \left( \sum_{i=1}^{n} \sum_{j \in N_i} w_{ij}^{(1)} + \sum_{i=1}^{n} \sum_{j \in N_j} w_{ji}^{(2)} - \frac{1}{n \rho} \sum_i g_i \right) + \frac{1}{n \rho} \mathbf{1}. \tag{2.30}
\]

The new \(s_{ij}\) is obtained from

\[
\arg\min_{s_{ij}} \mathcal{L} = \Pi_{S^+_n} (c_{ij} - D^T z_{ij} - A^T y_{ij} - B_1^T w_{ij}^{(1)} - B_2^T w_{ij}^{(2)} - \rho^{-1} x_{ij}), \tag{2.31}
\]

where \(\Pi_{S^+_n}\) is a projection to the positive semidefinite cone.
**Update for** \((y_{ij})\): The update for \(y_{ij}\) is the solution to a least-squares problem, given by

\[
\arg\min_{y_{ij}} \mathcal{L} = (\mathcal{A} \mathcal{A}^T)^{-1} (\mathcal{A}(c_{ij} - s_{ij} - D^T z_{ij} - B_1^T w_{ij}^{(1)} - B_2^T w_{ij}^{(2)} - \rho^{-1} x_{ij}) + \rho^{-1} b_e). \quad (2.32)
\]

By construction, \(\mathcal{A}\) is the matrix that encodes the linear constraints. Note that \(\mathcal{A}\) is of size \(m_E \times \mathcal{O}(n^2)\) and has linearly independent rows. Since \(m_E\) is of order \(\mathcal{O}(n)\), \(\mathcal{A} \mathcal{A}^T\) is a full-rank matrix of dimension \(\mathcal{O}(n)\).

**Update for** \((z_{ij}, w_{ij}^{(k)})\): The updates for \(z_{ij}\) and for the \(w_{ij}^{(k)}\)'s decouple due to the fact that the sampling matrices \(D\) and \(B_1\) and \(B_2\) have mutually orthogonal rows, as they sample different entries of \(X_{ij}\). To see this, we write the relevant minimization problem as follows

\[
\min_{w_{ij}^{(k)}, z_{ij}} \sum_{(i,j) \in E(\tilde{G}_B)} \delta_{K_p}(z_{ij}) \\
+ \frac{\rho}{2} \sum_{(i,j) \in E(\tilde{G}_B)} \left\| -c_{ij} + s_{ij} + D^T z_{ij} + \mathcal{A}^T y_{ij} + B_1^T w_{ij}^{(1)} + B_2^T w_{ij}^{(2)} + \frac{x_{ij}}{\rho} \right\|_2^2 \\
+ \frac{\rho}{2} \sum_{i=1}^n \left\| t - \sum_{j \in N_i} w_{ij}^{(1)} - \sum_{j:j \in N_j} w_{ji}^{(2)} + \frac{g_i}{\rho} \right\|_2^2.
\]

Recalling that \(N_i\) is the set of one-hop neighbors of node \(i\), define

\[
K_{N_i} = \begin{pmatrix}
B_1^T & B_2^T & D^T \\
\cdots & \cdots & \cdots \\
B_1^T & B_2^T & D^T \\
I & \cdots & I & I & \cdots & I
\end{pmatrix}
\]
which has \(|N_i|\) copies of \(B_1^T\), \(B_2^T\) and \(D^T\), \(2|N_i|\) copies of the identity, and is zero everywhere else. Then the problem becomes

\[
\min_{w_{ij}^{(k)}} \sum_{(i,j) \in E(G_B)} \delta_{K_p}(z_{ij}) + \sum_{i=1}^{n} \frac{\rho}{2} \left\| \mathcal{K}_{N_i} \text{vec} (V_{wz(i)}) \right\|^2_2 + \left\| \begin{pmatrix} v_{iN_i(1)} \\ \vdots \\ v_{iN_i(|N_i|)} \\ -t - \rho^{-1} y_i \end{pmatrix} \right\|^2_2
\]

(2.33)

where

\[
V_{wz(i)} = \begin{pmatrix} w_{iN_i(1)}^{(1)} & \cdots & w_{iN_i(|N_i|)}^{(1)} \\ \vdots & \ddots & \vdots \\ w_{iN_i(1)}^{(2)} & \cdots & w_{iN_i(|N_i|)}^{(2)} \\ z_{iN_i(1)} & \cdots & z_{iN_i(|N_i|)} \end{pmatrix}
\]

and

\[
v_{ij} = -c_{ij} + s_{ij} + \mathcal{A}^T y_{ij} + \frac{x_{ij}}{\rho}.
\]

\(\mathcal{K}_{N_i}^T\mathcal{K}_{N_i}\) has a block-diagonal structure, owing to the mutual orthogonality of the following blocks

\[
\begin{pmatrix} B_1^T \\ \vdots \\ B_1^T \end{pmatrix}, \begin{pmatrix} B_2^T \\ \vdots \\ B_2^T \end{pmatrix}, \begin{pmatrix} D^T \\ \vdots \\ D^T \end{pmatrix}, \begin{pmatrix} I & \cdots & I \\ I & \cdots & I \\ I & \cdots & I \end{pmatrix}
\]

as specified in 2.26.

The form of 2.33 also shows that the problem can be solved independently for each neighborhood. The optimal value for \(z_{ij}\) is then

\[
z_{ij} = \Pi_{K_p} \left( \mathcal{D} \left( c_{ij} - s_{ij} - \mathcal{A}^T y_{ij} - \frac{x_{ij}}{\rho} \right) \right),
\]

(2.34)

where \(\Pi_{K_p}\) is a projection to the positive cone.
For $w_{ij}^{(k)}$, let

$$\mathcal{B}_{N_i} = \begin{pmatrix} \mathcal{B}_1^T & \mathcal{B}_2^T \\ \vdots & \vdots \\ B_1^T & B_2^T \\ I & I & I & \cdots & I \end{pmatrix}. $$

Then

$$\text{vec}\left( \begin{bmatrix} w_{iN_1(1)}^{(1)} & \cdots & w_{iN_1(|N_i|)}^{(1)} & w_{iN_2(1)}^{(2)} & \cdots & w_{iN_2(|N_i|)}^{(2)} \end{bmatrix} \right) = (\mathcal{B}_{N_i}^T \mathcal{B}_{N_i})^{-1} \mathcal{B}_{N_i}^T \begin{bmatrix} v_{iN_1(1)} \\ \vdots \\ v_{iN_2(|N_i|)} \\ -t - \rho^{-1} g_i \end{bmatrix}. $$

The matrices $\mathcal{B}_{N_i}^T \mathcal{B}_{N_i}$ for $i = 1, \ldots, n$ take the generic form

$$\mathcal{B}_{N_i}^T \mathcal{B}_{N_i} = (\alpha - \beta) I_{|N_i|n} + \beta J_{|N_i|} \otimes I_n $$

where $\alpha$ and $\beta$ are constants. Their inverse can be verified to be the $n|N_i| \times n|N_i|$ matrix

$$ (\mathcal{B}_{N_i}^T \mathcal{B}_{N_i})^{-1} = \frac{1}{\alpha - \beta} I_{|N_i|n} - \frac{\beta}{(\alpha - \beta)(\alpha - \beta + |N_i|\beta)} J_{|N_i|} \otimes I_n $$

by direct computation. Let

$$\mathcal{H}_i = \frac{1}{\alpha - \beta} I_{|N_i|} - \frac{\beta}{(\alpha - \beta)(\alpha - \beta + |N_i|\beta)} J_{|N_i|} \quad (2.35)$$

52
which is a $|N_i| \times |N_i|$ matrix. Then, $(B^T_i B_i)^{-1} v = v_{Hi}$ and the update for $w_{ij}^{(k)}$’s is finally given by

$$\text{vec} \left( \begin{bmatrix} w_{iN_i(1)}^{(1)} & \cdots & w_{iN_i(|N_i|)}^{(1)} & w_{iN_i(1)}^{(2)} & \cdots & w_{iN_i(|N_i|)}^{(2)} \end{bmatrix} \right) = B^T_i \begin{bmatrix} v_{iN_i(1)} \\ \vdots \\ v_{iN_i(|N_i|)} \\ -t - \rho^{-1} g_{Hi} \end{bmatrix}.$$  

(2.36)

The updates for $x_{ij}$ and $g_i$, taken directly from [60], are the following:

$$x_{ij}^{k+1} = x_{ij}^k + \tau \rho \left( -c_{ij} + s_{ij} + D^T z_{ij} + A^T y_{ij} + B^T_i w_{ij}^{(1)} + B^T_2 w_{ij}^{(2)} \right) \quad (2.37)$$

and

$$g_{i}^{k+1} = g_{i}^k + \tau \rho \left( t - \sum_{j \in N_i} w_{ij}^{(1)} - \sum_{j \in N_j} w_{ji}^{(2)} \right). \quad (2.38)$$

This concludes the ADMM formulation for Problem 2.8 using edges. The problem for larger cliques is very similar, with additional variables $w_{ij}^{(k)}$ for $k > 2$.

The costliest update in the proposed ADMM scheme is the projection of a matrix of dimension $O(n)$ to the positive semidefinite cone. The updates can be parallelized (across the nodes, for e.g.), only requiring one gather operation per iteration for the $w_{ij}^{(k)}$ updates.

**Convergence criterion**

Throughout all of our evaluations, our choice of convergence criterion mimics that in [60], and is defined as follows

**Definition 2.2** (Convergence criterion, $\eta$).

$$\eta = \max (\eta_P, \eta_D, \eta_K, \eta_K^*, \eta_P, \eta_P^*, \eta_{C1}, \eta_{C2})\quad (2.39)$$
with

\[
\eta_P = \frac{\|AX-B_e\|_F}{1+\sqrt{n}\|b_e\|} \quad \eta_D = \frac{\|C+ATY+SD^TZ+B_1^TW^{(1)}+B_2^TW^{(2)}\|_F}{1+\sqrt{n}\|b_e\|}
\]

\[
\eta_K = \frac{\|\Pi S_n - (X)\|_F}{1+\|X\|_F} \quad \eta_{K^*} = \frac{\|\Pi S_n - (S)\|_F}{1+\|S\|_F}
\]

\[
\eta_P = \frac{\|X-P_Kp(X)\|_F}{1+\|X\|_F} \quad \eta_{P^*} = \frac{\|Z-P_Kp(Z)\|_F}{1+\|Z\|_F}
\]

\[
\eta_{C1} = \frac{\|(X,S)\|}{1+\|X\|_F+\|S\|_F} \quad \eta_{C2} = \frac{\|(X,D^TZ)\|}{1+\|X\|_F+\|D^TZ\|_F}
\]

where each column of \(X\) is one of the variables \(x_{ij}\) (an analogous statement holds for \(S\) and \(Z\)).

### 2.3.3 An empirical note on ADMM convergence

Algorithm 1, originally described in [60], was the first provably convergent 3-block ADMM scheme to be described in literature (convergence of the standard 2-block scheme is well established, and a thorough description of the approach can be found in [17]).

However, a direct extension of 2-block ADMM (e.g., by updating blocks in 1-2-3 order) has been employed in the past. In [22], the authors show that this straightforward extension is not necessarily convergent. In agreement with this fact, our own evaluations show that direct extension can fail in a dramatic fashion for certain orderings of the blocks.

As an example, Figures 2.2a and 2.2b depict convergence curves for the TSPLIB problem gr21 ([54]) and QAPLIB problem chr20a ([19]), respectively, ran to 2000 iterations for both Conic-ADMM3c and direct extension. When directly extending 2-block ADMM to 3-block ADMM with blocks \((s_{ij},t)\), \((y_{ij})\), and \((z_{ij},w_{ij}^{(k)})\), performing the updates in order 1-2-3 fails to converge (although we observe convergence when updating in order 1-3-2).
Figure 2.2: Convergence criterion [2.39] as a function of the number of iterations for problems from the TSPLIB and QAPLIB for both the Conic-ADMM3c algorithm of [60] (in blue) and the direct extension of 2-block ADMM to a multi-block setting (red and yellow). Updating blocks in order 1-2-3 fails to converge. Updating in order 1-3-2 yields a convergence curve similar to the one obtained by using the algorithm in [60].

2.4 Evaluation of C-SDP

While C-SDP was developed with the goal of tackling NMR assignment, it is, at its core, a convex relaxation to the quadratic assignment problem. As such, we make use of the extensive benchmark tests available for both the QAP and instances of the traveling salesman problem (itself a particular instance of the QAP) to evaluate the tightness of the relaxation in various other scenarios. This section summarizes the results of applying C-SDP to sparse quadratic assignment problems described in the QAP library (QAPLIB, [19]), and traveling salesman problems described in the TSP library (TSPLIB, [54]).

While convex relaxations of the QAP are typically evaluated on the strength of their lower bounds, we have observed that stochastic projection of $P \in \text{DS}(n)$ to $\text{Perm}(n)$ (as described in section COMPLETE below) can often yield strong upper bounds on QAP instances. As such, we include both lower bound and upper bound comparisons in our results.

1The code used in this work is available at https://github.com/fsbravo/csdp.git
2.4.1 Lower bounds

In this subsection we discuss lower bound results for problems in the QAPLIB which feature some level of sparsity. Comparisons are provided against existing results published in literature for the matrix-lifting approach MSDR3 [28], the convex quadratic programming approach QPB [6], and the matrix-splitting approaches SDMRS [52], [53]. We also implement the eigenspace approach from [26], and test both C-SDP and this relaxation on problems in the TSPLIB with \( n \leq 150 \). The eigenspace relaxation is particularly interesting in this regard as it simplifies to a linear program with an additional second-order cone constraint, due to the spectral properties of the circulant matrix \( B \).

We note that the matrix splitting approach in [51], adapted specifically to problems with Hamming and Manhattan distance matrices, obtains stronger bounds on problems in the \( \text{esc} \) family, which we discuss below. The work in [27] leverages group sparsity to solve the relaxation of [68] for large problems in the \( \text{esc} \), which at the time produced new best lower bounds. However, our evaluation focuses primarily on general relaxations with PSD variables of \( O(n) \times O(n) \) size.

QAPLIB

Table 2.1 summarizes the results on problems from the QAP library. It can be seen that C-SDP performs particularly well on problems from \( \text{chr} \) family. This is reassuring, as the sparse matrix in this problem is the one which more closely resembles that in NMR assignment. Particularly remarkable is the fact that C-SDP recovers solutions that are close to rank 1, confirming that much of the permutation structure is preserved on the diagonal of \( Q \).

The eigenspace relaxation appears to dominate the remainder of the problems, competing with MSDR3 for problems in the \( \text{esc} \) family. Note that the majority
of these problems are not particularly sparse, and that the performance of C-SDP correlates strongly with sparsity.

An evaluation of the runtimes for both C-SDP and eigenspace (solved using interior point solver SeDuMi [59]) can be found in Figure 2.3. As all problem instances for C-SDP were ran on 20 processors, runtimes are comparatively high for small problems with \( n < 20 \) (in fact, for \( k = 2 \) an interior point solver is faster for problems of this size). However, 1000 iterations of ADMM (which is sufficient for convergence to \( \eta < 10^{-5} \) for most problem instances) run in less than 20 min for \( k = 2 \) and in less than 1 h for \( k = 4 \) for \( n = 128 \). The results highlight the potential of first-order methods such as ADMM. The interior point implementation of eigenspace remains competitive until about \( n = 20 \).

![Figure 2.3: Comparison of run times between C-SDP and eigenspace relaxations on problems from the QAP library (with \( n \leq 150 \)). C-SDP instances were ran for 1000 ADMM iterations or until convergence on 20 processors. Eigenspace instances were solved using SeDuMi [59].](image)

**TSPLIB**

We illustrate the results on all TSPLIB problems with \( n \leq 150 \) in Figure 2.4 below (full tabulated results can be found in 2.6 in the Appendix at the end of this chapter).
Table 2.1: Comparison of lower bounds on sparse QAP problems for five different relaxations. C-SDP instances were ran with components of size 4 on 20 processors for 1000 iterations or until convergence. Eigenspace [27], [26] and MSDR3 [28] were implemented in Matlab, with SeDuMi [59]. Results for SDMRS [52], [53] and QPB [6] were obtained from existing literature.

<table>
<thead>
<tr>
<th>Problem</th>
<th>NZ</th>
<th>OPT</th>
<th>C-SDP</th>
<th>λ-space</th>
<th>SDMRS</th>
<th>MSDR3</th>
<th>QPB</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr12a</td>
<td>0.1528</td>
<td>9552</td>
<td>9499</td>
<td>8581</td>
<td>8500</td>
<td>-12025*</td>
<td>-19669</td>
</tr>
<tr>
<td>chr12b</td>
<td>0.1528</td>
<td>9742</td>
<td>8492</td>
<td>7980</td>
<td>7341</td>
<td>3276*</td>
<td>-84209</td>
</tr>
<tr>
<td>chr12c</td>
<td>0.1528</td>
<td>11156</td>
<td>10919</td>
<td>9899</td>
<td>9832</td>
<td>-16762*</td>
<td>-23041</td>
</tr>
<tr>
<td>chr15a</td>
<td>0.1244</td>
<td>9896</td>
<td>9387</td>
<td>7860</td>
<td>7442</td>
<td>-39692*</td>
<td>-49102</td>
</tr>
<tr>
<td>chr15b</td>
<td>0.1244</td>
<td>7990</td>
<td>6736</td>
<td>5371</td>
<td>5167</td>
<td>-30435*</td>
<td>-50945</td>
</tr>
<tr>
<td>chr15c</td>
<td>0.1244</td>
<td>9504</td>
<td>9504</td>
<td>9899</td>
<td>9832</td>
<td>-16762*</td>
<td>-24021</td>
</tr>
<tr>
<td>chr18a</td>
<td>0.1049</td>
<td>11098</td>
<td>10526</td>
<td>9375</td>
<td>9078</td>
<td>-47168*</td>
<td>-65610</td>
</tr>
<tr>
<td>chr18b</td>
<td>0.1049</td>
<td>1534</td>
<td>1534</td>
<td>1534</td>
<td>1534</td>
<td>-159*</td>
<td>-746</td>
</tr>
<tr>
<td>chr18c</td>
<td>0.1049</td>
<td>6156</td>
<td>6100</td>
<td>5990</td>
<td>5964</td>
<td>-13777*</td>
<td>-20194</td>
</tr>
<tr>
<td>chr18d</td>
<td>0.095</td>
<td>2192</td>
<td>2160</td>
<td>2157</td>
<td>2157</td>
<td>5605*</td>
<td>-7998</td>
</tr>
<tr>
<td>chr18e</td>
<td>0.095</td>
<td>2298</td>
<td>2293</td>
<td>2237</td>
<td>2237</td>
<td>6656*</td>
<td>-7384</td>
</tr>
<tr>
<td>chr18f</td>
<td>0.095</td>
<td>14142</td>
<td>10182</td>
<td>9996</td>
<td>9996</td>
<td>35401*</td>
<td>-81212</td>
</tr>
<tr>
<td>chr20a</td>
<td>0.0868</td>
<td>6156</td>
<td>6100</td>
<td>5990</td>
<td>5964</td>
<td>-13777*</td>
<td>-20194</td>
</tr>
<tr>
<td>chr20b</td>
<td>0.0868</td>
<td>6194</td>
<td>6181</td>
<td>6048</td>
<td>6015</td>
<td>-16089*</td>
<td>-21186</td>
</tr>
<tr>
<td>chr20c</td>
<td>0.0868</td>
<td>14142</td>
<td>10182</td>
<td>9996</td>
<td>9996</td>
<td>35401*</td>
<td>-81212</td>
</tr>
<tr>
<td>chr22a</td>
<td>0.0768</td>
<td>6156</td>
<td>6100</td>
<td>5990</td>
<td>5964</td>
<td>-13777*</td>
<td>-20194</td>
</tr>
<tr>
<td>chr22b</td>
<td>0.0768</td>
<td>6194</td>
<td>6181</td>
<td>6048</td>
<td>6015</td>
<td>-16089*</td>
<td>-21186</td>
</tr>
<tr>
<td>chr22c</td>
<td>0.0768</td>
<td>14142</td>
<td>10182</td>
<td>9996</td>
<td>9996</td>
<td>35401*</td>
<td>-81212</td>
</tr>
<tr>
<td>chr25a</td>
<td>0.0768</td>
<td>6156</td>
<td>6100</td>
<td>5990</td>
<td>5964</td>
<td>-13777*</td>
<td>-20194</td>
</tr>
<tr>
<td>chr25b</td>
<td>0.0768</td>
<td>6194</td>
<td>6181</td>
<td>6048</td>
<td>6015</td>
<td>-16089*</td>
<td>-21186</td>
</tr>
<tr>
<td>chr25c</td>
<td>0.0768</td>
<td>14142</td>
<td>10182</td>
<td>9996</td>
<td>9996</td>
<td>35401*</td>
<td>-81212</td>
</tr>
<tr>
<td>chr32a</td>
<td>0.2969</td>
<td>68</td>
<td>21</td>
<td>57</td>
<td>37</td>
<td>50</td>
<td>55</td>
</tr>
<tr>
<td>chr32b</td>
<td>0.7188</td>
<td>292</td>
<td>42</td>
<td>284</td>
<td>272</td>
<td>276</td>
<td>250</td>
</tr>
<tr>
<td>chr32c</td>
<td>0.3984</td>
<td>160</td>
<td>49</td>
<td>135</td>
<td>88</td>
<td>123</td>
<td>95</td>
</tr>
<tr>
<td>chr32d</td>
<td>0.1641</td>
<td>16</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>chr32e</td>
<td>0.1641</td>
<td>28</td>
<td>13</td>
<td>18</td>
<td>5</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>chr32f</td>
<td>0.1641</td>
<td>28</td>
<td>13</td>
<td>18</td>
<td>5</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>chr32g</td>
<td>0.1641</td>
<td>28</td>
<td>13</td>
<td>18</td>
<td>5</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>chr32h</td>
<td>0.6875</td>
<td>996</td>
<td>465</td>
<td>927</td>
<td>639</td>
<td>906</td>
<td>708</td>
</tr>
<tr>
<td>chr32i</td>
<td>0.1172</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>chr32j</td>
<td>0.0938</td>
<td>14</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>chr32k</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32l</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32m</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32n</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32o</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32p</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32q</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32r</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32s</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32t</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32u</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32v</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32w</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32x</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32y</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>chr32z</td>
<td>0.0938</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>esc64a</td>
<td>0.0317</td>
<td>-243</td>
<td>-243</td>
<td>-243</td>
<td>-243</td>
<td>-243</td>
<td>-243</td>
</tr>
<tr>
<td>scr12</td>
<td>0.3889</td>
<td>31410</td>
<td>27407</td>
<td>28986</td>
<td>29133</td>
<td>18803</td>
<td>8585</td>
</tr>
<tr>
<td>scr15</td>
<td>0.3733</td>
<td>51140</td>
<td>42425</td>
<td>46379</td>
<td>46016</td>
<td>39399</td>
<td>12479</td>
</tr>
<tr>
<td>scr20</td>
<td>0.0076</td>
<td>64</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>-999</td>
</tr>
<tr>
<td>ste36a</td>
<td>0.2654</td>
<td>15852</td>
<td>5698</td>
<td>7893</td>
<td>5250</td>
<td>10155</td>
<td>5196407</td>
</tr>
<tr>
<td>ste36b</td>
<td>0.2654</td>
<td>15852</td>
<td>5698</td>
<td>7893</td>
<td>5250</td>
<td>10155</td>
<td>5196407</td>
</tr>
<tr>
<td>ste36c</td>
<td>0.2654</td>
<td>15852</td>
<td>5698</td>
<td>7893</td>
<td>5250</td>
<td>10155</td>
<td>5196407</td>
</tr>
</tbody>
</table>

*Instances in which the interior point solver failed to converge to the required precision.
The eigenspace relaxation simplifies to a linear program with an additional second-order cone constraint, so it can be solved remarkably fast. However, this relies on the spectral properties of the circulant matrix \( B \), and thus \( A \) and \( B \) cannot be freely swapped to obtain two distinct bounds.

![Graph showing lower bounds for selected problems from the TSP library (\( n \leq 150 \)). Lower bounds from C-SDP with 4 nodes per variable (blue) and for eigenspace (yellow) are shown. C-SDP consistently shows smaller gaps than eigenspace, although eigenspace can be used to quickly generate a lower bound, since it simplifies to a linear program due to the simple spectrum of \( B \).](image)

It can be seen that the simplified bound is weaker than C-SDP in every tested problem (although the eigenspace SDP bound provided by the eigenspace relaxation with \( B \) and \( A \) swapped produces comparable bounds in the small problems we got to test). Interior point solutions are not feasible for the SDP version once problem size increases beyond \( n = 40 \). C-SDP, on the other hand, produces solutions with an
optimality gap between 5-15% for most problems, including problems of size $n = 150$. However, as we shall see when evaluating upper bounds, not much of the structure of $P$ is preserved for the case of TSP.

A runtime comparison between the simplified bound (implemented with SeDuMi) and C-SDP can be found in Figure 2.5. Unlike in the case of QAPLIB, eigenspace is remarkably fast after simplification. But as the size of the problem grows, the distributed ADMM implementation of C-SDP again becomes competitive as it can be distributed across many processors.

![Figure 2.5: Comparison of run times between C-SDP and eigenspace (linear program) relaxations on problems from the TSP library (with $n \leq 150$). C-SDP instances were ran for 1000 ADMM iterations or until convergence on 20 processors. Eigenspace instances were solved using SeDuMi [59]. As the eigenspace relaxation simplifies to a linear program with a second-order cone constraint in the case of TSP, interior point solvers are competitive with the ADMM scheme used in C-SDP, even across multiple processors.](image)

2.4.2 Upper bounds

Unlike other relaxations, both the canonical relaxation of Zhao et al and C-SDP successfully preserve much of the structure of the original permutation $P$ within the PSD variable $Q$. To evaluate the extent to which this structure is preserved, we
use C-SDP to produce upper bounds on the same QAPLIB and TSPLIB problems studied in the previous subsection.

We compare the results obtained against the convex-concave method PATH [67], which has seen extensive use in problems in computer vision, and which has been known to produce reasonable upper bounds for QAP problems. Note, however, that PATH is several orders of magnitude faster than C-SDP.

**Recovering feasible solutions**

Recall that the variables $X_{ij}$ encode the columns of the relaxed $P$ within their diagonal. One can thus extract a doubly stochastic matrix, $D$, from the slack variables $p_i, i = 1, \ldots, n$ introduced in the ADMM algorithm. This doubly stochastic matrix can, in turn, be projected to Perm($n$) by solving the bipartite weighted matching problem we first described in 1.10.

Rather than producing a single solution, we add random noise to $D \in DS(n)$ before projecting, a total of 10000 times. This tends to produce a bound about 10% better than directly projecting $D$, as we have observed that $D$ converges to fractional solutions where weight is evenly distributed across a few entries in each row and column.

**QAPLIB**

Table 2.2 summarizes the results of projecting C-SDP solutions to the feasible non-convex set Perm($n$). It is clear that C-SDP generally produces stronger upper bounds than PATH. Remarkably, we observe that the stochastic projection method recovers the optimal solution to 9 of the QAP instances tested, illustrating that this type of upper bound can be immediately useful, especially since it comes at nearly no additional computational cost.
Table 2.2: Comparison between upper bounds given by C-SDP and PATH on selected problems from the QAP library with (relatively) sparse B. C-SDP instances were ran for 1000 ADMM iterations or until convergence on 20 processors.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Sparsity</th>
<th>OPT</th>
<th>C-SDP Value</th>
<th>C-SDP Gap (%)</th>
<th>PATH Value</th>
<th>PATH Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr12a</td>
<td>0.1528</td>
<td>9552</td>
<td>9552</td>
<td>0.0</td>
<td>13628</td>
<td>42.7</td>
</tr>
<tr>
<td>chr12b</td>
<td>0.1528</td>
<td>9742</td>
<td>10900</td>
<td>11.9</td>
<td>13450</td>
<td>38.1</td>
</tr>
<tr>
<td>chr12c</td>
<td>0.1528</td>
<td>11156</td>
<td>11414</td>
<td>2.9</td>
<td>13230</td>
<td>18.6</td>
</tr>
<tr>
<td>chr15a</td>
<td>0.1244</td>
<td>9896</td>
<td>10108</td>
<td>2.1</td>
<td>15038</td>
<td>52.0</td>
</tr>
<tr>
<td>chr15b</td>
<td>0.1244</td>
<td>7990</td>
<td>10090</td>
<td>26.3</td>
<td>20662</td>
<td>158.6</td>
</tr>
<tr>
<td>chr15c</td>
<td>0.1244</td>
<td>9504</td>
<td>9504</td>
<td>0.0</td>
<td>15522</td>
<td>63.3</td>
</tr>
<tr>
<td>chr18a</td>
<td>0.1049</td>
<td>11098</td>
<td>11118</td>
<td>0.2</td>
<td>19568</td>
<td>76.3</td>
</tr>
<tr>
<td>chr18b</td>
<td>0.1049</td>
<td>1534</td>
<td>1670</td>
<td>8.9</td>
<td>3058</td>
<td>99.3</td>
</tr>
<tr>
<td>chr20a</td>
<td>0.095</td>
<td>2192</td>
<td>3152</td>
<td>43.8</td>
<td>4284</td>
<td>95.4</td>
</tr>
<tr>
<td>chr20b</td>
<td>0.095</td>
<td>2298</td>
<td>2512</td>
<td>9.3</td>
<td>4186</td>
<td>82.2</td>
</tr>
<tr>
<td>chr20c</td>
<td>0.095</td>
<td>14144</td>
<td>28292</td>
<td>100.1</td>
<td>26712</td>
<td>88.9</td>
</tr>
<tr>
<td>chr22a</td>
<td>0.0868</td>
<td>6156</td>
<td>6340</td>
<td>3.0</td>
<td>8516</td>
<td>38.3</td>
</tr>
<tr>
<td>chr22b</td>
<td>0.0868</td>
<td>6194</td>
<td>6256</td>
<td>1.0</td>
<td>8696</td>
<td>40.4</td>
</tr>
<tr>
<td>chr22c</td>
<td>0.0768</td>
<td>3796</td>
<td>4752</td>
<td>25.2</td>
<td>6450</td>
<td>69.9</td>
</tr>
<tr>
<td>esc16a</td>
<td>0.2969</td>
<td>68</td>
<td>74</td>
<td>8.8</td>
<td>76</td>
<td>11.8</td>
</tr>
<tr>
<td>esc16b</td>
<td>0.7188</td>
<td>292</td>
<td>292</td>
<td>0.0</td>
<td>300</td>
<td>2.7</td>
</tr>
<tr>
<td>esc16c</td>
<td>0.3984</td>
<td>160</td>
<td>168</td>
<td>5.0</td>
<td>170</td>
<td>6.3</td>
</tr>
<tr>
<td>esc16d</td>
<td>0.1641</td>
<td>16</td>
<td>18</td>
<td>12.5</td>
<td>28</td>
<td>75.0</td>
</tr>
<tr>
<td>esc16e</td>
<td>0.1641</td>
<td>28</td>
<td>30</td>
<td>7.1</td>
<td>34</td>
<td>21.4</td>
</tr>
<tr>
<td>esc16f</td>
<td>0.1641</td>
<td>26</td>
<td>26</td>
<td>0.0</td>
<td>30</td>
<td>15.4</td>
</tr>
<tr>
<td>esc16h</td>
<td>0.6875</td>
<td>996</td>
<td>996</td>
<td>0.0</td>
<td>1164</td>
<td>16.9</td>
</tr>
<tr>
<td>esc16i</td>
<td>0.1172</td>
<td>14</td>
<td>14</td>
<td>0.0</td>
<td>22</td>
<td>57.1</td>
</tr>
<tr>
<td>esc16j</td>
<td>0.0938</td>
<td>8</td>
<td>8</td>
<td>0.0</td>
<td>14</td>
<td>75.0</td>
</tr>
<tr>
<td>esc32a</td>
<td>0.1445</td>
<td>130</td>
<td>278</td>
<td>113.8</td>
<td>252</td>
<td>93.8</td>
</tr>
<tr>
<td>esc32b</td>
<td>0.2109</td>
<td>168</td>
<td>352</td>
<td>109.5</td>
<td>316</td>
<td>88.1</td>
</tr>
<tr>
<td>esc32c</td>
<td>0.2559</td>
<td>642</td>
<td>724</td>
<td>12.8</td>
<td>692</td>
<td>7.8</td>
</tr>
<tr>
<td>esc32d</td>
<td>0.1758</td>
<td>200</td>
<td>272</td>
<td>36.0</td>
<td>242</td>
<td>21.0</td>
</tr>
<tr>
<td>esc32e</td>
<td>0.0117</td>
<td>2</td>
<td>2</td>
<td>0.0</td>
<td>14</td>
<td>600.0</td>
</tr>
<tr>
<td>esc32f</td>
<td>0.0176</td>
<td>6</td>
<td>6</td>
<td>0.0</td>
<td>28</td>
<td>366.7</td>
</tr>
<tr>
<td>esc64a</td>
<td>0.0317</td>
<td>116</td>
<td>178</td>
<td>53.4</td>
<td>240</td>
<td>106.9</td>
</tr>
<tr>
<td>esc128</td>
<td>0.0076</td>
<td>64</td>
<td>176</td>
<td>175.0</td>
<td>206</td>
<td>221.9</td>
</tr>
<tr>
<td>scr12</td>
<td>0.3889</td>
<td>31410</td>
<td>31884</td>
<td>1.5</td>
<td>49674</td>
<td>58.1</td>
</tr>
<tr>
<td>scr15</td>
<td>0.3733</td>
<td>51140</td>
<td>62574</td>
<td>22.4</td>
<td>92266</td>
<td>80.4</td>
</tr>
<tr>
<td>scr20</td>
<td>0.31</td>
<td>110030</td>
<td>153670</td>
<td>39.7</td>
<td>168216</td>
<td>52.9</td>
</tr>
<tr>
<td>ste36a</td>
<td>0.2654</td>
<td>9526</td>
<td>16218</td>
<td>70.2</td>
<td>16798</td>
<td>76.3</td>
</tr>
<tr>
<td>ste36b</td>
<td>0.2654</td>
<td>15852</td>
<td>45774</td>
<td>188.8</td>
<td>40994</td>
<td>158.6</td>
</tr>
<tr>
<td>ste36c</td>
<td>0.2654</td>
<td>8239110</td>
<td>13411140</td>
<td>62.8</td>
<td>15095338</td>
<td>83.2</td>
</tr>
</tbody>
</table>

**TSPLIB**

Both PATH and C-SDP fail to produce any meaningful bounds on TSP problems, as illustrated in Figure 2.6. This is not surprising in the case of C-SDP, as the symmetry in the problem leads to a flat doubly stochastic matrix $D$, which retains very little of
the structure of the original permutation. Fortunately, that symmetric is absent in the maximum path formulation of NMR assignment.

Figure 2.6: Upper bounds for selected problems in the TSP library ($n \leq 150$). Upper bounds from C-SDP with 4 nodes per variable (blue) and for PATH (yellow) are shown. C-SDP generally shows a smaller gap than PATH.


2.5 C-SDP for NMR assignment

Having seen that C-SDP can indeed recover good solutions to sparse QAP problems, it is now time to return to the application at hand. Recall that at the start of this chapter the NMR assignment problem was described in terms of the minimization of discrepancy along a path. The following formulation was proposed

$$\min_{\pi} \sum_{i=1}^{n-1} d(s_{\pi(i)}, s_{\pi(i+1)})$$

s.t. $\pi$ is a permutation

with $d(s_{\pi(i)}, s_{\pi(i+1)}) \propto -\log Pr[s_i \rightarrow s_j]$.

To tie this back to the quadratic assignment problem, let

$$D \triangleq \begin{pmatrix} d(s_1, s_1) & \cdots & d(s_1, s_n) \\ \vdots & \ddots & \vdots \\ d(s_n, s_1) & \cdots & d(s_n, s_n) \end{pmatrix} \quad (2.40)$$

and $B$ is the adjacency matrix for the path graph 2.2.4. Then the problem can be expressed in the following alternative form

$$\min_{\pi} \quad \text{Tr} (B^T PDP^T)$$

s.t. $P \in \text{Perm}(n)$

which amounts to finding a maximum likelihood reordering of the spin systems under a Gaussian noise model. This problem fits readily into the framework of C-SDP, due to the sparsity of $B$. 

64
2.5.1 Accounting for a prior

As discussed in 1.1.2, a crucial factor in accurate assignment is statistical typing. As we saw in much of the existing literature discussed in 1.3, this process is typically understood as matching to a prior. Letting \( r \) be a residue of the protein under study with amino acid type \( t \), the backbone chemical shifts are typically assumed to follow a Gaussian distribution, as follows:

\[
\begin{pmatrix}
N \\
H \\
C^\alpha \\
C^\beta \\
C^O
\end{pmatrix}
\sim \mathcal{N}(\mu_t, \Sigma_t),
\]

where \( \mu_t \) and \( \Sigma_t \) are assembled from deposition statistics in databases such as BMRB [61]. \( \Sigma_t \) is, in fact, usually assumed to be diagonal (although IPASS assumes a multivariate distribution for \( C^\alpha \) and \( C^\beta \) with non-diagonal covariance). Under this assumption, for each residue \( k \) in the molecule, one can compute a mapping score \( w_{ik} \) between spin system \( s_i = \begin{pmatrix} N_i & H_i & C^\alpha_i & C^\beta_i & C^O_i & C^\alpha_{i-1} & C^\beta_{i-1} & C^O_{i-1} \end{pmatrix} \) and the given position

**Definition 2.3 (Mapping score).**

\[
w_{ik} \triangleq - \log \Pr [s_i \mid r_k, r_{k-1}].
\]

In terms of the quadratic assignment problem, this information bears on \( P \) directly, and is best accounted for through a linear term in the objective \( \text{Tr} (W^T P) \), letting
\[ W \triangleq [w_{ik}], \text{ to yield} \]
\[
\min_{\pi} \quad \text{Tr} \left( B^T PDP^T \right) + \text{Tr} \left( W^T P \right) \\
\text{s.t.} \quad P \in \text{Perm}(n)
\]

Such a formulation of the QAP amounts to minimizing some pairwise penalty along a path of length \( n - 1 \), with an additional matching penalty given by the \( w_{ik} \)'s. While the formulation that we presented was inspired by probabilistic assumptions about NMR assignment data, there is no \textit{apriori} requirement that the penalties be defined directly as such. Therefore, we propose the following

\textbf{Problem 2.9} (QAP formulation of NMR assignment). Let \( k_1 : \mathbb{R} \rightarrow \mathbb{R}, k_2 : \mathbb{R} \rightarrow \mathbb{R} \) be arbitrary elementwise kernels.

\[
\min_{\pi} \quad \text{Tr} \left( B^T Pk_1(D)P^T \right) + \text{Tr} \left( k_1(W)^T P \right) \\
\text{s.t.} \quad P \in \text{Perm}(n)
\]

\textbf{Choosing a kernel} A kernel \( k(X) = X \) amounts to viewing NMR assignment through the distributional assumptions which we have described so far in this thesis. While this kernel shows good results, we observed that a kernel of the form

\[
k(X) = -\frac{1}{\gamma} \exp \left( \gamma \frac{X}{\|X\|_F} \right) \tag{2.42}
\]

tends to yield integral solutions in C-SDP more often than the identity kernel, and ultimately recovers accurate solutions more often as well. This kernel is assumed for the remainder of the chapter.

\textbf{Dealing with missing spin systems} In almost all NMR datasets, regardless of the quality of the spectra, some spin systems will be absent from the available data.
We introduce token spin systems (whose discrepancy and matching scores are derived from an expected log-probability) to deal with this mismatch.

### 2.5.2 Benchmark datasets

To assess the performance of C-SDP in NMR datasets we resorted to problems of size $n \leq 100$ in the benchmark dataset first described in [65], consisting of a number of proteins for which spin systems are created from existing assignments in a BMRB entry. To match the evaluation of IPASS [4], we construct each spin system from the assigned frequencies in the NMRStar file, as follows

\[
\mathbf{s}_i = \begin{pmatrix} 
N_i \\
H_i \\
C^\alpha_i \\
C_i^\beta \\
C_{i-1}^\alpha + \epsilon_i^\alpha \\
C_{i-1}^\beta + \epsilon_i^\beta 
\end{pmatrix} 
\]

where $\epsilon_i^\alpha \sim \mathcal{N}(0, \sigma^2_\alpha), \epsilon_i^\beta \sim \mathcal{N}(0, \sigma^2_\beta)$, except for residues where the base pair does not exist (such as all prolines). Two sets of $\sigma$’s were used: (1) low-noise, $\sigma_\alpha = 0.08, \sigma_\beta = 0.16$ and (2) high-noise, $\sigma_\alpha = 0.16, \sigma_\beta = 0.32$. Results are compared with other fully automated assignment tools: MARS [41], RANDOM [8], CISA [65], and IPASS [5].

**Evaluation:** Let $n_m$ be the number of spin systems assigned in the BMRB file, $n_a$ be the number assigned by the algorithm, and $n_c$ be the number of correctly assigned spin systems. We then define precision as $n_c/n_m$, and recall as $n_c/n_a$. These values as calculated for each of the chosen algorithms are presented in Tables 2.3 and 2.4 below. The results for C-SDP were averaged across 10 runs.
Table 2.3: Accuracy of assignment (precision/recall) of various algorithms and C-SDP on synthetic spin systems with noise level = (0.08, 0.16). Results for MARS [41], RANDOM [8], and CISA taken from [65]. Results for IPASS taken from [5].

<table>
<thead>
<tr>
<th>Protein ID</th>
<th>Length</th>
<th>MARS</th>
<th>RANDOM</th>
<th>CISA</th>
<th>IPASS</th>
<th>C-SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmr4391</td>
<td>66</td>
<td>100/76</td>
<td>67/63</td>
<td>97/97</td>
<td>93/90</td>
<td>99/99</td>
</tr>
<tr>
<td>bmr4752</td>
<td>68</td>
<td>100/97</td>
<td>40/35</td>
<td>96/94</td>
<td>100/94</td>
<td>100/100</td>
</tr>
<tr>
<td>bmr4144</td>
<td>78</td>
<td>100/91</td>
<td>36/33</td>
<td>100/99</td>
<td>98/85</td>
<td>100/100</td>
</tr>
<tr>
<td>bmr4579</td>
<td>86</td>
<td>99/98</td>
<td>54/51</td>
<td>98/98</td>
<td>100/98</td>
<td>100/100</td>
</tr>
<tr>
<td>bmr4316</td>
<td>89</td>
<td>100/100</td>
<td>42/36</td>
<td>100/99</td>
<td>99/98</td>
<td>99/99</td>
</tr>
</tbody>
</table>

Table 2.4: Accuracy of assignment (precision/recall) of various algorithms and C-SDP on synthetic spin systems with noise level = (0.16, 0.32). Results for MARS [41], RANDOM [8], and CISA taken from [65]. Results for IPASS taken from [5].

<table>
<thead>
<tr>
<th>Protein ID</th>
<th>Length</th>
<th>MARS</th>
<th>RANDOM</th>
<th>CISA</th>
<th>IPASS</th>
<th>C-SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmr4391</td>
<td>66</td>
<td>100/75</td>
<td>58/55</td>
<td>91/91</td>
<td>93/90</td>
<td>100/100</td>
</tr>
<tr>
<td>bmr4752</td>
<td>68</td>
<td>100/97</td>
<td>36/30</td>
<td>90/88</td>
<td>100/94</td>
<td>99/99</td>
</tr>
<tr>
<td>bmr4144</td>
<td>78</td>
<td>100/69</td>
<td>33/31</td>
<td>100/99</td>
<td>98/85</td>
<td>96/96</td>
</tr>
<tr>
<td>bmr4579</td>
<td>86</td>
<td>96/90</td>
<td>34/32</td>
<td>80/80</td>
<td>100/98</td>
<td>100/100</td>
</tr>
<tr>
<td>bmr4316</td>
<td>89</td>
<td>99/91</td>
<td>35/30</td>
<td>83/83</td>
<td>99/98</td>
<td>98/98</td>
</tr>
</tbody>
</table>

We see that C-SDP outperforms other methods in terms of both precision and recall, on average. Importantly, C-SDP vastly outperforms RANDOM, illustrating the importance of considering the prior during the construction of an optimal path. Like with IPASS and MARS, the increase in noise level does not impact the performance of C-SDP too much.

Note that since C-SDP always produces a full permutation matrix, it assigns all spin systems, such that $n_a = n_m$, where $n_m$ is assumed to be the number of assignable spin systems in the protein. As a result, precision and recall values for C-SDP are equal. In practice, matches where the prior does not satisfy a minimum threshold requirement could be eliminated, but we found that no such procedure was necessary in this dataset.

These results show great promise in using C-SDP to recover exact assignments from complete NMR spin system data in the presence of noise, and highlight the capacity of the relaxation to preserve enough of the structure of the permutation $P$. 

68
This adds to the observation in section 2.4 that C-SDP often recovers solutions that are close to rank 1 in sparse problems without symmetry.

2.5.3 Performance on experimental datasets

The simulated datasets, however useful for benchmarking, lack some of the challenges presented by experimental datasets. To evaluate the performance of C-SDP in these instances, we made use of spin system datasets provided by the authors of IPASS, and computed via the spin system construction procedure described in 1.3.2. The results are summarize in Table 2.5 below.

Table 2.5: Accuracy of assignment on four distinct spin system datasets provided by the authors of IPASS. Results for other algorithms were obtained from [4].

<table>
<thead>
<tr>
<th>Protein</th>
<th>Length</th>
<th>Manual1</th>
<th>Spins2*</th>
<th>RIBRA</th>
<th>MARS</th>
<th>IPASS</th>
<th>C-SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM1112</td>
<td>89</td>
<td>83</td>
<td>81(74)/85</td>
<td>40/54</td>
<td>55/63</td>
<td>71/72</td>
<td>50/85</td>
</tr>
<tr>
<td>VRAR</td>
<td>72</td>
<td>60</td>
<td>47(41)/47</td>
<td>4/13</td>
<td>6/17</td>
<td>30/37</td>
<td>19/47</td>
</tr>
<tr>
<td>HACS1</td>
<td>74</td>
<td>61</td>
<td>48(46)/61</td>
<td>5/11</td>
<td>15/16</td>
<td>37/50</td>
<td>19/61</td>
</tr>
</tbody>
</table>

1 Number of manually assigned residues in the BMRB file.
2 Correct/Total available spin systems

*The spin system lists available to us do not corroborate the numbers found in the original paper. The numbers in parentheses represent the total number of correct spin systems as identified by us through matching ground truth chemical shifts extracted from an NMRStar file to the list of provided spin systems.

Experimental datasets bring forth the main weaknesses of C-SDP in a way simulated datasets could not. We quickly notice that existence of missing and incorrect data greatly hinders the performance of the relaxation, since, by virtue of its formulation, C-SDP must assign these incorrect (or missing) spin systems to some location along the protein. It attempts to do so in a way that minimizes global discrepancy, but it struggles to detect and ignore outliers. While competitive, C-SDP, by itself, is unable to offer improvements over existing state-of-the-art automated assignment solutions.
2.6 Conclusion

This chapter introduced a formulation of NMR assignment as a quadratic assignment problem, and described a novel SDP relaxation called C-SDP, which further relaxes the SDR-3 relaxation of [68] to leverage the sparsity of graph $B$. Section 2.3 described a carefully derived ADMM scheme for the relaxation which is provably convergent and which distributes easily across many processors.

A Matlab implementation of the proposed algorithm was used in section 2.4 to obtain both lower and upper bounds on many problems in the QAPLIB and TSPLIB. It was observed that the C-SDP relaxation produces very strong lower bounds on sparse problems such as those in the chr family, and competitive results with other relaxations with similar-sized PSD variables on other sparse problems such as esc and ste. As expected, the performance of C-SDP is highly correlated with sparsity.

Surprisingly, it was also observed that much of structure of the original permutation $P$ is preserved in the diagonal of the SDP variables used by C-SDP. As a result, upper bounds obtained from projecting the doubly stochastic matrix produced by C-SDP to the set of permutation matrices $\text{Perm}(n)$ successfully recovers the optimal solutions in several of the tested problems, as long as the problem is not overly symmetric (as in the case of TSP).

C-SDP was turned to the problem of NMR assignment in section 2.5 under a formulation that accounts for both connectivity between spin systems, and a prior, via a linear term in the objective of the QAP. State-of-the-art results were observed on a benchmark simulated dataset first proposed in [64]. Results on real experimental data produced comparable results to other automated NMR assignment algorithms, but without improvement over the state-of-the-art, as C-SDP struggled to deal with outliers (erroneous data) and missing data.
## Appendix

Table 2.6: Comparison between lower bounds given by the C-SDP and eigenspace relaxations on problems from the TSP library (with \( n \leq 150 \)). C-SDP instances were ran for 1000 ADMM iterations or until convergence on 20 processors.

<table>
<thead>
<tr>
<th>Problem</th>
<th>OPT</th>
<th>C-SDP&lt;sub&gt;4&lt;/sub&gt;</th>
<th>( \lambda )-space</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Bound</td>
<td>Time (s)</td>
</tr>
<tr>
<td>att48</td>
<td>10628</td>
<td>9442</td>
<td>253.4</td>
</tr>
<tr>
<td>bayg29</td>
<td>1610</td>
<td>1513</td>
<td>106.8</td>
</tr>
<tr>
<td>bays29</td>
<td>2020</td>
<td>1876</td>
<td>122.0</td>
</tr>
<tr>
<td>berlin52</td>
<td>7542</td>
<td>6907</td>
<td>294.7</td>
</tr>
<tr>
<td>hier127</td>
<td>118282</td>
<td>106305</td>
<td>3549.0</td>
</tr>
<tr>
<td>brazil58</td>
<td>25395</td>
<td>19290</td>
<td>335.7</td>
</tr>
<tr>
<td>burma14</td>
<td>3323</td>
<td>3003</td>
<td>66.7</td>
</tr>
<tr>
<td>ch130</td>
<td>6110</td>
<td>5307</td>
<td>3692.8</td>
</tr>
<tr>
<td>ch150</td>
<td>6528</td>
<td>6297</td>
<td>8179.1</td>
</tr>
<tr>
<td>dantzig42</td>
<td>699</td>
<td>595</td>
<td>226.7</td>
</tr>
<tr>
<td>ell101</td>
<td>629</td>
<td>619</td>
<td>2153.2</td>
</tr>
<tr>
<td>ell51</td>
<td>426</td>
<td>404</td>
<td>354.7</td>
</tr>
<tr>
<td>ell76</td>
<td>538</td>
<td>515</td>
<td>1137.7</td>
</tr>
<tr>
<td>fri26</td>
<td>937</td>
<td>864</td>
<td>146.6</td>
</tr>
<tr>
<td>gr120</td>
<td>6942</td>
<td>6448</td>
<td>3980.3</td>
</tr>
<tr>
<td>gr137</td>
<td>69853</td>
<td>63434</td>
<td>5365.0</td>
</tr>
<tr>
<td>gr17</td>
<td>2085</td>
<td>1759</td>
<td>86.5</td>
</tr>
<tr>
<td>gr21</td>
<td>2707</td>
<td>2573</td>
<td>96.2</td>
</tr>
<tr>
<td>gr24</td>
<td>1272</td>
<td>1172</td>
<td>106.3</td>
</tr>
<tr>
<td>gr48</td>
<td>5046</td>
<td>4504</td>
<td>283.5</td>
</tr>
<tr>
<td>gr96</td>
<td>55209</td>
<td>50939</td>
<td>1469.8</td>
</tr>
<tr>
<td>hk48</td>
<td>11461</td>
<td>10634</td>
<td>249.3</td>
</tr>
<tr>
<td>kroA100</td>
<td>21282</td>
<td>18708</td>
<td>1917.8</td>
</tr>
<tr>
<td>kroA150</td>
<td>26524</td>
<td>25106</td>
<td>7231.9</td>
</tr>
<tr>
<td>kroB100</td>
<td>22141</td>
<td>19619</td>
<td>1708.2</td>
</tr>
<tr>
<td>kroB150</td>
<td>26130</td>
<td>25014</td>
<td>7463.9</td>
</tr>
<tr>
<td>kroC100</td>
<td>20749</td>
<td>18511</td>
<td>1797.5</td>
</tr>
<tr>
<td>lin105</td>
<td>14379</td>
<td>12009</td>
<td>1917.5</td>
</tr>
<tr>
<td>pr107</td>
<td>44303</td>
<td>27747</td>
<td>2015.5</td>
</tr>
<tr>
<td>pr124</td>
<td>59030</td>
<td>49482</td>
<td>3408.2</td>
</tr>
<tr>
<td>pr136</td>
<td>96772</td>
<td>94114</td>
<td>4701.3</td>
</tr>
<tr>
<td>pr144</td>
<td>58537</td>
<td>33308</td>
<td>5929.2</td>
</tr>
<tr>
<td>pr76</td>
<td>108159</td>
<td>92770</td>
<td>854.8</td>
</tr>
<tr>
<td>rat99</td>
<td>1211</td>
<td>1165</td>
<td>1648.8</td>
</tr>
<tr>
<td>rd100</td>
<td>7910</td>
<td>7137</td>
<td>1888.3</td>
</tr>
<tr>
<td>st70</td>
<td>675</td>
<td>595</td>
<td>610.1</td>
</tr>
<tr>
<td>swiss42</td>
<td>1273</td>
<td>1141</td>
<td>186.3</td>
</tr>
<tr>
<td>ulysses16</td>
<td>6859</td>
<td>5696</td>
<td>73.1</td>
</tr>
<tr>
<td>ulysses22</td>
<td>7013</td>
<td>5701</td>
<td>91.6</td>
</tr>
</tbody>
</table>
Table 2.7: Comparison between upper bounds given by C-SDP and PATH on problems from the TSP library (with $n \leq 150$). C-SDP instances were ran for 1000 ADMM iterations or until convergence on 20 processors.

<table>
<thead>
<tr>
<th>Problem</th>
<th>OPT</th>
<th>C-SDP Value</th>
<th>Gap (%)</th>
<th>PATH Value</th>
<th>Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>att48</td>
<td>10628</td>
<td>33267</td>
<td>213.0</td>
<td>45679</td>
<td>329.8</td>
</tr>
<tr>
<td>bayg29</td>
<td>1610</td>
<td>3450</td>
<td>114.3</td>
<td>4992</td>
<td>210.1</td>
</tr>
<tr>
<td>bays29</td>
<td>2020</td>
<td>4193</td>
<td>107.6</td>
<td>5349</td>
<td>164.8</td>
</tr>
<tr>
<td>berlin52</td>
<td>7542</td>
<td>17135</td>
<td>127.2</td>
<td>28702</td>
<td>280.6</td>
</tr>
<tr>
<td>bier127</td>
<td>118282</td>
<td>347523</td>
<td>193.8</td>
<td>395272</td>
<td>234.2</td>
</tr>
<tr>
<td>brazil58</td>
<td>25395</td>
<td>76379</td>
<td>200.8</td>
<td>110977</td>
<td>337.0</td>
</tr>
<tr>
<td>burma14</td>
<td>3323</td>
<td>4142</td>
<td>24.6</td>
<td>6495</td>
<td>95.5</td>
</tr>
<tr>
<td>ch130</td>
<td>6110</td>
<td>27641</td>
<td>352.4</td>
<td>44070</td>
<td>621.3</td>
</tr>
<tr>
<td>ch150</td>
<td>6528</td>
<td>27302</td>
<td>318.2</td>
<td>51528</td>
<td>689.3</td>
</tr>
<tr>
<td>dantzig42</td>
<td>699</td>
<td>1915</td>
<td>174.0</td>
<td>1272</td>
<td>82.0</td>
</tr>
<tr>
<td>eil101</td>
<td>629</td>
<td>2059</td>
<td>227.3</td>
<td>3382</td>
<td>437.7</td>
</tr>
<tr>
<td>eil51</td>
<td>426</td>
<td>1293</td>
<td>203.6</td>
<td>1467</td>
<td>244.4</td>
</tr>
<tr>
<td>eil76</td>
<td>538</td>
<td>1523</td>
<td>183.0</td>
<td>2304</td>
<td>328.2</td>
</tr>
<tr>
<td>fri26</td>
<td>937</td>
<td>1306</td>
<td>39.4</td>
<td>1327</td>
<td>41.6</td>
</tr>
<tr>
<td>gr120</td>
<td>6942</td>
<td>25100</td>
<td>261.6</td>
<td>49819</td>
<td>617.6</td>
</tr>
<tr>
<td>gr137</td>
<td>69853</td>
<td>223757</td>
<td>220.3</td>
<td>97057</td>
<td>38.9</td>
</tr>
<tr>
<td>gr17</td>
<td>2085</td>
<td>2761</td>
<td>32.4</td>
<td>3896</td>
<td>86.9</td>
</tr>
<tr>
<td>gr21</td>
<td>2707</td>
<td>4502</td>
<td>66.3</td>
<td>7735</td>
<td>185.7</td>
</tr>
<tr>
<td>gr24</td>
<td>1272</td>
<td>2212</td>
<td>73.9</td>
<td>2918</td>
<td>129.4</td>
</tr>
<tr>
<td>gr48</td>
<td>5046</td>
<td>14501</td>
<td>187.4</td>
<td>18691</td>
<td>270.4</td>
</tr>
<tr>
<td>gr96</td>
<td>55209</td>
<td>166591</td>
<td>201.7</td>
<td>80624</td>
<td>46.0</td>
</tr>
<tr>
<td>hk48</td>
<td>11461</td>
<td>35264</td>
<td>207.7</td>
<td>43736</td>
<td>281.6</td>
</tr>
<tr>
<td>kroA100</td>
<td>21282</td>
<td>121095</td>
<td>469.0</td>
<td>174544</td>
<td>720.2</td>
</tr>
<tr>
<td>kroA150</td>
<td>26524</td>
<td>135527</td>
<td>411.0</td>
<td>277376</td>
<td>945.8</td>
</tr>
<tr>
<td>kroB100</td>
<td>22141</td>
<td>91576</td>
<td>313.6</td>
<td>160339</td>
<td>624.2</td>
</tr>
<tr>
<td>kroB150</td>
<td>26130</td>
<td>118556</td>
<td>353.7</td>
<td>246856</td>
<td>844.7</td>
</tr>
<tr>
<td>kroC100</td>
<td>20749</td>
<td>113097</td>
<td>445.1</td>
<td>179055</td>
<td>763.0</td>
</tr>
<tr>
<td>lin105</td>
<td>14379</td>
<td>48137</td>
<td>234.8</td>
<td>50102</td>
<td>248.4</td>
</tr>
<tr>
<td>pr107</td>
<td>44303</td>
<td>124699</td>
<td>181.5</td>
<td>62752</td>
<td>41.6</td>
</tr>
<tr>
<td>pr124</td>
<td>59030</td>
<td>165420</td>
<td>180.2</td>
<td>98941</td>
<td>67.6</td>
</tr>
<tr>
<td>pr136</td>
<td>96772</td>
<td>256183</td>
<td>164.7</td>
<td>287028</td>
<td>196.6</td>
</tr>
<tr>
<td>pr144</td>
<td>58537</td>
<td>207781</td>
<td>255.0</td>
<td>93526</td>
<td>59.8</td>
</tr>
<tr>
<td>pr76</td>
<td>108159</td>
<td>316093</td>
<td>192.2</td>
<td>150780</td>
<td>39.4</td>
</tr>
<tr>
<td>rat99</td>
<td>1211</td>
<td>3167</td>
<td>161.5</td>
<td>6589</td>
<td>444.1</td>
</tr>
<tr>
<td>rd100</td>
<td>7910</td>
<td>37597</td>
<td>375.3</td>
<td>47973</td>
<td>506.5</td>
</tr>
<tr>
<td>st70</td>
<td>675</td>
<td>2706</td>
<td>300.9</td>
<td>3293</td>
<td>387.9</td>
</tr>
<tr>
<td>swiss42</td>
<td>1273</td>
<td>3351</td>
<td>163.2</td>
<td>3742</td>
<td>194.0</td>
</tr>
<tr>
<td>ullyses16</td>
<td>6859</td>
<td>8248</td>
<td>20.2</td>
<td>12533</td>
<td>82.7</td>
</tr>
<tr>
<td>ullyses22</td>
<td>7013</td>
<td>11012</td>
<td>57.0</td>
<td>15868</td>
<td>126.3</td>
</tr>
</tbody>
</table>
Chapter 3

Assignment through Linear Programming

This chapter describes our best attempts so far to address the limitations of existing NMR assignment tools, including (and perhaps especially so) those of the solution proposed in Chapter 2.

As we concluded, our semidefinite programming formulation of NMR assignment suffered from the following flaws:

- It applies to spin systems only, and as a result, in runs the risk of propagating errors introduced during spin system construction - most evident in the experiments with real data;

- The solution to the relaxation is not necessarily integral, and the projection step is poorly understood;

- There is no clean way to deal with either missing, or excess, spin systems, both of which are inevitable in real datasets;

- Large proteins $n > 150$ require access to a cluster for time-efficient solutions, which may not be available or easy to setup in all NMR facilities.
In this chapter we attempt to address all of these limitations. We begin this chapter by formulating NMR assignment as a problem of finding a simple path in a directed graph (sections 3.1 and 3.2), applicable to spin systems or to peak lists directly. A linear programming relaxation approach to finding feasible solutions to the problem is described in section 3.3. Finally, we test our proposed approach on both simulated and experimental data in section 3.4.

### 3.1 Modeling NMR assignment

There are two main considerations in assignment of both spin systems and peaks: (1) consistency with prior (defined by statistics extracted from deposition databases such as [61]) and (2) assignment consistency, whereby the chemical shifts assigned to the same atom should be within an experimental tolerance, or some small statistical deviation.

Different algorithms for assignment have taken distinct approaches to enforce such requirements. IPASS [4], as an example, enforces consistency explicitly within its fragments, such that matching becomes primarily a matter of fitting a prior. RANDOM [8] also focuses on building consistent paths first, and assignment based on a prior is left as a post-processing consideration. FLYA, [57], a current state-of-the-art representative, enforces consistency explicitly during the construction of solutions and local optimization routines, but it accounts for both statistical agreement with the prior and within each atom in its global score.

Our proposed approach attempts to bridge the gap between the two, while providing significant flexibility in formulating the probabilistic model and constraints to follow (which may vary from experiment to experiment). It is best illustrated with recourse to Figure 3.1, which we describe step-by-step in the following sections. For now, we summarize the steps as follows
(a) **Enumeration of valid assignments:** For each residue, all (or a large number of) valid assignments are enumerated to create nodes in a directed graph.

(b) **Statistical typing:** Each node is statistically typed against its residue to filter any assignments which are too unlikely to be deemed possible. The governing mechanism for node elimination is typically a thresholding function based on a statistical test.

(c) **Probabilistic modeling:** Edges are added to the graph. The weight of an edge between nodes in residues \( r_k \) and \( r_{k+1} \) is the log probability of assignment to the atoms of residue \( r_k \) given the values assigned in each of the nodes. This accounts for the fact that peaks associated with the base amide pair \(^1\text{H}–^{15}\text{N}\) of residue \( r_k \) may include atoms in residue \( r_{k-1} \), as discussed in 1.1.2.

(d) **Path finding:** A longest path in the directed graph is found, subject to any additional constraints of interest (e.g. that a spin system cannot be utilized more than once).

The next section summarizes the process by which the graph is built, while section 3.3 describes the approach to finding a maximal path.

### 3.2 Building an assignment graph

The goal of steps (a) through (c) of the pipeline is the creation of a graph where a path from start to end represents the full log-likelihood of the data. The process by which this graph is built involves enumeration of valid assignments, statistical typing, and the construction of appropriate edge weights to represent the underlying data. Since the processes involved are different depending on whether the available data is in the form of spin systems or peak lists, we describe the two separately.
Figure 3.1: Illustration of graph assignment model. (a) Possible chemical shift assignments are determined and enumerated for each residue, creating nodes in a graph. (b) Each node is statistically typed against its residue’s distribution, and very low likely nodes are eliminated. (c) Edges are placed between nodes $s_i, s_j$ in adjacent residues, $r_k, r_{k+1}$ with weight $w_{ij} = \log \Pr[r_k(s_i, s_j)]$, if they satisfy a minimum acceptable threshold. Empty nodes are added to each residue and connected to every node in the preceding and succeeding layers with edge weights equal to the threshold. (d) A longest path is found between the start and end nodes, subject to any additional constraints (e.g. that spin systems cannot be used more than once).
3.2.1 Graph construction for spin systems

(a) — Enumeration of valid assignments

The process of enumerating valid assignments is particularly simple for spin system datasets. Given a list of spin systems \( s_1, \ldots, s_m \), one can match each residue \( r_1, \ldots, r_n \) against each valid spin system, amounting to the creation of at most \( mn \) nodes (in practice, fewer nodes will be created as some residue types, such as proline, do not produce spin systems given the standard set of heteronuclear experiments described in 1.1.2).

Frequencies within a spin system are assigned to the corresponding atoms in the residue to create a node. As a result, at the end of this procedure, each residue \( r_k \) is associated with \( m \) nodes, representing all assignment possibilities of spin systems to residues. The optimization process will amount to selecting a single best node for each residue in order to produce a valid assignment.

(b) — Statistical typing

This step attempts to eliminate any enumerate assignments which are deemed too unlikely to be correct. We achieve this by defining a node score based on a distributional assumption on the chemical shifts for each residue, and eliminate all nodes that fall outside of a set node threshold.

**Definition 3.1** (Node score, spin system). Let \( r_k \) be a residue, with assignable atom set \( A_k \) (e.g. \( A_k = \{N_k, H_k, C_{\alpha}^a, C_{\beta}^a, C_{\gamma}^O\} \)). Let \( s_i \) be the set of chemical shift observations associated with spin system \( i \). Then the node score, \( g^i_k \), for residue \( r_k \) under spin system \( i \) is given by

\[
g^i_k \triangleq \sum_{a \in A_k} \log N(s_i(a) | \mu_a, \sigma_a), \quad (3.1)
\]
where $s_i$ denotes the chemical shift in spin system $s_i$ which would match atom $a$. If spin system $i$ does not include a valid chemical shift for atom $a$, then $s_i(a) = \mu_a + \delta \sigma_a$, where $\delta$ is a user-defined parameter.

The node threshold is defined along the same logic.

**Definition 3.2** (Node threshold, spin system).

$$
\delta_k \triangleq \sum_{a \in A_k} \log \mathcal{N}(\mu_a + \delta \sigma_a \mid \mu_a, \sigma_a) = |A_k| \log \mathcal{N}(\delta \mid 0, 1).
$$

(3.2)

Of course, any reasonable statistical test can be applied at this stage for a similar purpose. An example of another test we have observed to be well discriminative first normalizes all the variables to the standard normal and then adds their squares to form the random variable

$$
z^i_k \triangleq \sum_{a \in A_k} \frac{(\mu_a - s_i(a))^2}{\sigma^2_a} \sim \chi^2_{|A_k|}
$$

from which a standard p-test can be derived. However, as we shall see, our choice of node score intends to save some computational time by preemptively computing variables of interest for edge creation.

(c) — **Edge creation (probabilistic modeling)**

The last step in the creation of the graph is to set edges and edge weights such that a path from a node in the first layer to a node in the last represents the likelihood of the selected assignment. However, the development of an intuitive probabilistic model to represent spin systems is hindered by the fact that the construction of spin systems hides away much of the structure behind the experimental observations of the chemical shifts.
For the purposes of the experiments described in this work, the generative model used by the authors in [64] is adopted, and the following edge weight is used:

**Definition 3.3** (Edge weight, spin systems). Let $r_k, r_{k+1}$ be adjacent residues, and let $\mathcal{A}_k$ be the assignable atom set of $r_k$. Let $s_i, s_j$ be two spin systems corresponding to nodes $v_i$ and $v_j$ in the graph, for residues $r_k$ and $r_{k+1}$, respectively. Let $\mathcal{A}_k^{k+1}$ be the set of atoms in $r_k$ which also appear in peaks associated with residue $r_{k+1}$. Then

\[
 w_{ij} = \log \Pr[r_k(s_i, s_j)] 
 \begin{align*}
 \triangleq \sum_{a \in \mathcal{A}_k} \log \mathcal{N}(s_i(a) \mid \mu_a, \sigma_a) + \sum_{a \in \mathcal{A}_k^{k+1}} \log \mathcal{N}(s_j(a) \mid s_i(a), \sigma_e(a)) \end{align*}
\]

(3.3)

(3.4)

where $\sigma_e(a)$ is the standard deviation associated with atom $a$ under the given set of experiments.

In all spin system experiments described in this work, we have

\[
 s_i = \left(N_i \ H_i \ C_\alpha^i \ C_\beta^i \ C_\alpha^{i-1} \ C_\beta^{i-1}\right)
\]

where we abused notation to indicate that the last two elements correspond to observed chemical shifts corresponding to atoms in the preceding residue. As a result, we have

\[\mathcal{A}_k = \{N_k, H_k, C_\alpha^k, C_\beta^k\} \quad \text{and} \quad \mathcal{A}_k^{k+1} = \{C_\alpha^k, C_\beta^k\}\]

A similar thresholding procedure as that used for node filtering can be adopted in the context of edges as well.

**Definition 3.4** (Edge threshold, spin systems). The edge threshold between nodes $v_i$ and $v_j$, $\epsilon_{ij}$, is given by

\[
 \epsilon_{ij} \triangleq \left(\mathcal{A}_k \right) \mathcal{N}(\delta \mid 0, 1). \]

(3.5)
3.2.2 Graph construction for peak lists

(a) — Enumeration of valid assignments

One would wish the enumeration of valid assignments to be as thorough as possible. However, unlike spin systems, peak lists can be combined in an exponential number of ways to generate atom assignments, such that thorough enumeration is not computationally feasible, which is why algorithms like FLYA resort to local optimization routines and multiple solutions to heuristically explore the space of valid assignments. For the preliminary tests described in this work, we make use of a graph-based enumerator to explore maximum-completeness assignments:

1. Let $S_1, \ldots, S_L$ be a list of peak lists corresponding to different heteronuclear experiments. Choose some $S_k$ as a fingerprinting experiment (typically HSQC or HNCO). Given a set tolerance for the base amide pair (we use the FLYA tolerance), $(\delta_N, \delta_H)$, we build a graph $G_{NH}$ with edges between each peak in $S_k$ and all peaks in other spectra which satisfy the tolerance threshold. The maximal cliques of the graph $G_{NH}$ represent subsets of the peak lists which are consistent in the $^1H-^{15}N$ plane.

2. Let $C_k$ be such a clique. Let $G_{C_k}$ be a graph with edges between any measured peaks in $C_k$ which are consistent in the $^{13}C$ dimension according to some tolerance level $\delta_C$.

3. Construct an auxiliary graph, $G_E(C_k)$, where each node corresponds to a valid pairing between a measured peak and a hypothetical expected peak (for an arbitrary residue under the given set of heteronuclear experiments), $(m_i, e_j)$. For each two nodes, $(m_{i_1}, e_{j_1}), (m_{i_2}, e_{j_2}) \in G_E(C_k)$, an edge is inserted in $G_E(C_k)$ if $j_1 \neq j_2$ or if $(m_{i_1}, m_{i_2}) \in G_{C_k}$. Note this implies that an edge exists only between pairings whose measured peaks are assigning values to different carbon atoms, or if they are assigning consistent values to that atom.
The maximal cliques of this graph represent all valid assignments of the measured peaks in $C_k$ to the set of expected peaks associated with an arbitrary residue under the given set of heteronuclear experiments. We can therefore conclude that the given construction can enumerate all valid assignments of measured peaks to an arbitrary residue under the given tolerances ($\delta_N, \delta_H, \delta_C$).

In practice, the set is prohibitively large to consider, so for each clique $C_k$, we consider only the 4 largest cliques within $G_E(C_k)$ (a minimum of 2 is required to cover both valid assignments of $^{13}C_{\alpha_i}$ and $^{13}C_{\alpha_i-1}$ under the standard set of 7 heteronuclear experiments).

**(b) — Statistical typing**

A similar type of statistical test as used for spin systems could also be used for peak lists. However, an important distinction when matching peak lists is that multiple observations exist for the same atom. Recall (or see, e.g. [18]) that the product of $n$ univariate Gaussian probability density functions, $\mathcal{N}(\mu_1, \sigma_1), \ldots, \mathcal{N}(\mu_n, \sigma_n)$ is given by

$$\prod_{i=1}^{n} \mathcal{N}(\mu_i, \sigma_i) = S_{1\ldots n} \mathcal{N}(\mu_{1\ldots n}, \sigma_{1\ldots n})$$  \hspace{1cm} (3.6)$$

with

$$\sigma_{1\ldots n} = \left( \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \right)^{-1/2}$$ \hspace{1cm} (3.7)$$

$$\mu_{1\ldots n} = \left( \sum_{i=1}^{n} \frac{\mu_i}{\sigma_i^2} \right) \sigma_{1\ldots n}^2$$ \hspace{1cm} (3.8)$$

$$S_{i=1\ldots n} = \frac{1}{(2\pi)^{(n-1)/2}} \sqrt{\frac{\sigma_{1\ldots n}^2}{\prod_{i=1}^{n} \sigma_i^2}} \exp \left[ -\frac{1}{2} \left( \sum_{i=1}^{n} \frac{\mu_i^2}{\sigma_i^2} - \frac{\mu_{1\ldots n}^2}{\sigma_{1\ldots n}^2} \right) \right].$$ \hspace{1cm} (3.9)$$

Now consider an atom $a$ with $n$ observations, $v_1, \ldots, v_n$, and with prior distribution $\mathcal{D}_a = \mathcal{N}(\mu_a, \sigma_a)$. Let $x_a$ be the (unknown) true chemical shift of $a$. Then, making an
independence assumption and using symmetry

\[
\Pr[v_1, \ldots, v_n \mid x_a] = \prod_{i=1}^{n} \mathcal{N}(v_i \mid x_a, \sigma_i) = \prod_{i=1}^{n} \mathcal{N}(x_a \mid v_i, \sigma_i). \tag{3.10}
\]

With this result in hand, we define the node score for atom \( a \) as follows

**Definition 3.5 (Atom score).** Given a set of observations \( V_a \) for atom \( a \)

\[
h_{a}^{V_a} \triangleq \log \mathbb{E}_{x_a \sim \mathcal{D}_a} \left[ |V_a| \prod_{i=1}^{n} \mathcal{N}(x_a \mid v_i, \sigma_i) \right] \tag{3.11}
\]

is the log of the expected joint probability of the observations, under the distributional assumptions for the prior. Note that

\[
\mathbb{E}_{x_a \sim \mathcal{D}_a} \left[ |V_a| \prod_{i=1}^{n} \mathcal{N}(x_a \mid v_i, \sigma_i) \right] = \mathbb{E}_{x_a \sim \mathcal{D}_a} \left[ S_{1\ldots n} \mathcal{N}(x_a \mid v_{1\ldots n}, \sigma_{1\ldots n}) \right] = \int_{-\infty}^{+\infty} S_{1\ldots n} \mathcal{N}(x_a \mid v_{1\ldots n}, \sigma_{1\ldots n}) \mathcal{N}(x_a \mid \mu_a, \sigma_a) dx_a
\]

\[
= \int_{-\infty}^{+\infty} S_{1\ldots na} \mathcal{N}(x_a \mid \mu_{1\ldots na}, \sigma_{1\ldots na}) dx_a
\]

\[
= S_{1\ldots na}.
\]

The score is therefore just a simple function of the observations, \( v_1, \ldots, v_n \), and distributional properties of the prior and experiments. For the sake of completion, we have

\[
\sigma_{1\ldots na} = \left( \frac{1}{\sigma_a^2} + \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \right)^{-1/2} \tag{3.12}
\]

\[
\mu_{1\ldots na} = \left( \frac{\mu_a}{\sigma_a^2} + \sum_{i=1}^{n} \frac{v_i}{\sigma_i^2} \right) \sigma_{1\ldots na}^2 \tag{3.13}
\]

\[
S_{i=1\ldots na} = \frac{1}{(2\pi)^{n/2}} \sqrt{\frac{\sigma_{1\ldots na}^2}{\sigma_a^2 \prod_{i=1}^{n} \sigma_i^2}} \exp \left[ -\frac{1}{2} \left( \frac{\mu_a^2}{\sigma_a^2} + \sum_{i=1}^{n} \frac{v_i^2}{\sigma_i^2} - \frac{\mu_{1\ldots na}^2}{\sigma_{1\ldots na}^2} \right) \right]. \tag{3.14}
\]
We are now in a position to define the node score and threshold for the case of peak lists.

**Definition 3.6** (Node score, peak list). Let \( r_k \) be a residue, with assignable atom set \( \mathcal{A}_k \). Let \( \mathcal{V}_a^i \) be the set of observations of atom \( a \) for node \( i \). Then the node score for residue \( k \) under the observations of node \( i \), \( g_k^i \), is given by

\[
g_k^i \triangleq \sum_{a \in \mathcal{A}_k} \log h_{a}^{\mathcal{V}_a^i} \tag{3.15}
\]

with \( h_{a}^{\mathcal{V}_a^i} \) as per Definition 3.5. Missing observations (arising due to a missing measured peak in the experimental dataset, for example) are treated as an off-center observation. Let \( \mathcal{V}_a^i = \{v_1, \ldots, v_l\} \), and imagine that a total of \( l + m \) observations were expected for this atom. Then \( \mathcal{V}_a^i \) is augmented as follows:

\[
\mathcal{V}_a^{i,+} = \mathcal{V}_a^i \cup \left\{ \left( \frac{1}{l} \sum_{i=1}^{l} v_i \right) + \delta(-1)^j \sigma_j, j = 1, \ldots, m \right\}
\]

where \( \delta \) is a user-defined parameter.

The node threshold follows a similar logic.

**Definition 3.7** (Node threshold, peak list). Let \( a \) be an atom with \( m \) expected observations. Define

\[
\mathcal{V}_a^\delta \triangleq \left\{ \mu_a + \delta \sigma_a + \delta(-1)^j \sigma_j, j = 1, \ldots, m \right\} \tag{3.16}
\]

The node threshold, \( \delta_k \), is given by

\[
\delta_k \triangleq \sum_{a \in \mathcal{A}_k} \log h_{a}^{\mathcal{V}_a^\delta}. \tag{3.17}
\]

To summarize, a node score is defined based on an accepted probabilistic model, and nodes in the graph are filtered according to a threshold score. This threshold
score is constructed around a user-defined parameter, $\delta$, which denotes how many standard deviations away from the distributional mean is acceptable for assignment.

(c) — Edge creation (probabilistic modeling)

The machinery developed for calculating node scores and thresholds applies readily to the problem of calculating edge weights. The crucial distinction is that edge weights between nodes for residues $r_k$ and $r_{k+1}$ depend on chemical shift observations associated with peaks in both residues.

**Definition 3.8** (Edge weight, peak lists). Let $r_k$ be a residue, with assignable atom set $\mathcal{A}_k$. Let $\mathcal{V}_i^a, \mathcal{V}_j^a$ be the sets of observations of atom $a$ for nodes $s_i$ and $s_j$, respectively. Then the node score for residue $k$ under the observations of node $i$, $g_k^i$ is given by

$$w_{ij} = \Pr[r_k(s_i, s_j)] \triangleq \sum_{a \in \mathcal{A}_k} \log h_a^{\mathcal{V}_i^a \cup \mathcal{V}_j^a}$$

with $h_a^{\mathcal{V}_i^a \cup \mathcal{V}_j^a}$ as per Definition 3.5.

3.2.3 Choosing $\delta$

Choosing $\delta$ amounts to a tradeoff between recall and performance. It is important to point out that choosing $\delta = 2$, for example, does not imply that any one atom that lies 2 standard deviations away from the prior mean will cause the node to be filtered, since the threshold is defined using all atoms in a residue. As such, a value of $\delta = 2$ is not unreasonable (and is quite unlikely to cause issues due to false negatives).

We also note that hard-thresholds might be useful under a model such as the one employed in FLYA, where it is assumed that frequencies assigned to the same atom cannot violate a set threshold. In this instance, the atom score can be adjusted to return $-\infty$ if such a violation is encountered.
In practice, any reasonable threshold should be able to eliminate most of the statistically unlikely nodes.

### 3.2.4 Dummy, start, and end nodes

A single dummy is introduced at each layer, $k$, with incoming edges from all nodes in layer $k-1$ and outgoing edges to all nodes in layer $k+1$. Edge weights can be set to the edge threshold. This process, illustrated in Figure 3.1 part (c), guarantees that a feasible path can be found, and can also help diagnose where the best path is failing to connect.

A start node (with outgoing edges to all nodes in layer 1), and an end node (with incoming edges from all nodes in layer $n$), are also added to the graph, with weight 0. These are used for convenience, to find the best inner path through the $n$ residues.

### 3.3 Path finding (d)

The edge creation described above finalized the construction of a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the weight of a path from the start to the end node represents the log-likelihood of the data given the prior. In particular, we make the following observation

**Definition 3.9** (Log-likelihood path). Let $\mathcal{P} = (\text{start}, s_1, \ldots, s_n, \text{end})$ be a directed path from the start node to the end node. Then the weight of the path, $w(\mathcal{P})$, is given by

$$w(\mathcal{P}) \triangleq \sum_{i=1}^{n} w_{i,i+1} = \sum_{i=1}^{n} \log \Pr [r_i(s_i, s_{i+1})] = \log \Pr [r_1, \ldots, r_n] \quad (3.19)$$

under the independence assumptions. This applies to both spin systems and peak lists, under the models given.
As a result, our goal is that of finding the longest such path. Note that the shortest path problem described in [1.2.4] achieves this straightforwardly by setting

$$w_{ij} \leftarrow -w_{ij} + \max_{(i,j) \in E} w_{ij}$$

to avoid negative cycles. However, such an approach does not allow for constraints to be imposed on the path. In particular, assignment typically calls for uniqueness of data in the solution. We propose the following problem

**Problem 3.1 (NMR assignment).** Let $G = (V, E)$ be the graph created by the process described in section 3.2. Let $u_j \in \{0, 1\}^m$ be an indicator vector associated with node $s_j$ representing the subset of data utilized in that node. Solve

$$\max \left\{ \sum_{(i,j) \in E} w_{ij} x_{ij} \right\}
\text{s.t. } 
\sum_{j: (i,j) \in E} x_{ij} - \sum_{j: (j,i) \in E} x_{ji} = \begin{cases} 
1 & \text{if } i = \text{start} \\
-1 & \text{if } i = \text{end} \\
0 & \text{otherwise}
\end{cases}
\sum_{(i,j) \in E} x_{ij} u_j \leq 1_m
\quad \forall(i,j) \in E.
$$

In the case of spin system data, $m$ would be the number of available spin systems. For peak lists, $m$ would instead be the number of peaks available across all spectra. This constraint can be relaxed in instances where spectra are ambiguous (i.e. where it is believed that two peaks may exist instead of just one).

The problem as described above is an instance of the constrained shortest (longest) path problem, and is NP-hard [2]. However, as discussed in [1.2.4], linear programming
relaxations can often recover integral solutions in this kind of problem. We first propose the following relaxation

**Problem 3.2** (NMR assignment, R1). Let $G = (\mathcal{V}, \mathcal{E})$ be the assignment graph. Let $u_j \in \{0, 1\}^m$ be an indicator vector associated with node $s_j$ representing the subset of data utilized in that node. Solve

$$\max_{\{x_{ij}\}} \sum_{(i,j) \in \mathcal{E}} w_{ij} x_{ij}$$

subject to

$$\sum_{j: (i,j) \in \mathcal{E}} x_{ij} - \sum_{j: (j,i) \in \mathcal{E}} x_{ji} = \begin{cases} 1 & \text{if } i = \text{start} \\ -1 & \text{if } i = \text{end} \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{(i,j) \in \mathcal{E}} x_{ij} u_j \leq 1_m$$

$$0 \leq x_{ij} \leq 1 \quad \forall (i, j) \in \mathcal{E}.$$ 

Note that this relaxation is not guaranteed to produce an integral solution (as the feasible polytope is restricted by the utilization constraints in addition to the hypercube constraints). However, as we shall observe in evaluation, integral solutions are indeed obtained for many of the tested problems.

An alternative relaxation that can more often produce integral solutions without requiring further work is the following

**Problem 3.3** (NMR assignment, R2). Let $G = (\mathcal{V}, \mathcal{E})$ be the assignment graph. Let $u_j \in \{0, 1\}^m$ be an indicator vector associated with node $s_j$ representing the subset
of data utilized in that node. Solve

$$\max_{\{x_{ij}\}} \sum_{(i,j) \in E} w_{ij} x_{ij} + \lambda^T \epsilon_-$$

s.t. $$\sum_{j:(i,j) \in E} x_{ij} - \sum_{j:(j,i) \in E} x_{ji} = \begin{cases} 
1 & \text{if } i = \text{start} \\
-1 & \text{if } i = \text{end} \\
0 & \text{otherwise}
\end{cases}$$

$$\sum_{(i,j) \in E} x_{ij} u_j + \epsilon_+ - \epsilon_- = 1_m$$

$$\epsilon_+, \epsilon_- \geq 0$$

$$0 \leq x_{ij} \leq 1 \quad \forall (i,j) \in E$$

where we penalize overutilized data through the slack variable $\epsilon_-$. We observe that this relaxation is particularly useful in problem instances where datasets are of poor quality, and the reutilization of data is an advantage in finding a solution of higher likelihood.

A final relaxation that is often of use is the following

**Problem 3.4** (NMR assignment, R3). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the assignment graph. Let $u_j \in \{0,1\}^m$ be an indicator vector associated with node $s_j$ representing the subset
of data utilized in that node. Let \(0 \leq a < 1\). Solve

\[
\max_{\{x_{ij}\}} \sum_{(i,j) \in E} w_{ij} x_{ij}
\]

\[
\text{s.t.} \quad \sum_{j : (i,j) \in E} x_{ij} - \sum_{j : (j,i) \in E} x_{ji} = \begin{cases} 
1 & \text{if } i = \text{start} \\
-1 & \text{if } i = \text{end} \\
0 & \text{otherwise}
\end{cases}
\]

\[
\sum_{(i,j) \in E} x_{ij} u_j \leq 1_m
\]

\[
0 \leq x_{ij} \leq a \quad \forall (i, j) \in E.
\]

This is similar to the relaxation R1, except that a stricter upper bound is imposed on the decision variables \(x_{ij}\). The purpose of this is to force the solution to disperse across many nodes. We use this as a subgraph selection tool, to generate a smaller feasible assignment graph on which we can solve the original integer linear program 3.1 directly using branch-and-bound techniques.

The following section summarizes the results of our proposed pipeline on both simulated and experimental data.

### 3.4 Evaluation

This section summarizes the results of our assignment pipeline on both simulated and experimental data.

#### 3.4.1 Simulated data

**CISA**

As a first sanity check, we tested our approach (name LP for short) on the entirety of the benchmark dataset developed by the authors of CISA in [64], as it provides a useful
comparison to many other fully automated algorithms on problems of small, medium, and large scale. We note that the full dataset contains many proteins on which we never tested C-SDP. The results were obtained by using the LP-R3 relaxation \[3.4\] with \(a = 0.8\) and then solving the ILP on the subgraph induced by the solution.

The results are summarized in Tables 3.1 and 3.2 for the two distinct noise levels considered in [4].

Table 3.1: Accuracy of assignment (precision/recall) of various algorithms and LP on synthetic spin systems with noise level = (0.08, 0.16). Results for MARS [41] and CISA taken from [65]. Results for IPASS taken from [5].

<table>
<thead>
<tr>
<th>Protein ID</th>
<th>Length</th>
<th>N (^1)</th>
<th>MARS</th>
<th>CISA</th>
<th>IPASS</th>
<th>C-SDP</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmr4391</td>
<td>66</td>
<td>59</td>
<td>100/76</td>
<td>97/97</td>
<td>93/90</td>
<td>99/99</td>
<td>90/90</td>
</tr>
<tr>
<td>bmr4752</td>
<td>68</td>
<td>66</td>
<td>100/97</td>
<td>96/94</td>
<td>100/94</td>
<td>100/100</td>
<td>100/100</td>
</tr>
<tr>
<td>bmr4144</td>
<td>78</td>
<td>68</td>
<td>100/91</td>
<td>100/99</td>
<td>98/85</td>
<td>100/100</td>
<td>99/96</td>
</tr>
<tr>
<td>bmr4579</td>
<td>86</td>
<td>83</td>
<td>99/98</td>
<td>98/98</td>
<td>100/98</td>
<td>100/100</td>
<td>100/99</td>
</tr>
<tr>
<td>bmr4316</td>
<td>89</td>
<td>85</td>
<td>100/100</td>
<td>100/99</td>
<td>99/98</td>
<td>99/99</td>
<td>100/100</td>
</tr>
<tr>
<td>bmr4288</td>
<td>105</td>
<td>94</td>
<td>100/99</td>
<td>98/98</td>
<td>100/98</td>
<td>99/99</td>
<td></td>
</tr>
<tr>
<td>bmr4929</td>
<td>114</td>
<td>110</td>
<td>100/100</td>
<td>93/91</td>
<td>100/100</td>
<td>100/98</td>
<td></td>
</tr>
<tr>
<td>bmr4302</td>
<td>115</td>
<td>107</td>
<td>100/100</td>
<td>96/95</td>
<td>100/99</td>
<td>100/99</td>
<td></td>
</tr>
<tr>
<td>bmr4670</td>
<td>120</td>
<td>102</td>
<td>100/100</td>
<td>96/95</td>
<td>98/97</td>
<td>99/99</td>
<td></td>
</tr>
<tr>
<td>bmr4353</td>
<td>126</td>
<td>98</td>
<td>95/55</td>
<td>96/95</td>
<td>99/93</td>
<td>95/95</td>
<td></td>
</tr>
<tr>
<td>bmr4207</td>
<td>158</td>
<td>148</td>
<td>100/99</td>
<td>100/99</td>
<td>100/97</td>
<td>99/99</td>
<td></td>
</tr>
<tr>
<td>bmr4318</td>
<td>215</td>
<td>191</td>
<td>99/99</td>
<td>87/84</td>
<td>100/98</td>
<td>98/98</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) Number of assignable spin systems in the BMRB data.

It can be seen that the LP approach achieves results that are comparable to both IPASS and MARS (the top contenders on this dataset). Note, in particular, that LP performs strongly on \textit{bmr4144}, where both IPASS and MARS struggle on recall, and on \textit{bmr4353}, which is particularly challenging due to the large number of prolines and a smaller number of available spin systems.

Conversely, LP appears to perform poorly on the smallest protein, \textit{bmr4391}, although MARS and IPASS both also struggle to achieve good recall on this instance. An analysis of the assignment produced by LP revealed that the maximum-likelihood assignment under the chosen probabilistic model does not match the ground truth assignment. In particular, LP consistently found a higher likelihood assignment on
Table 3.2: Accuracy of assignment (precision/recall) of various algorithms and LP on synthetic spin systems with noise level = (0.16, 0.32). Results for MARS \[41\] and CISA taken from \[65\]. Results for IPASS taken from \[5\].

<table>
<thead>
<tr>
<th>Protein ID</th>
<th>Length</th>
<th>N(^1)</th>
<th>MARS</th>
<th>CISA</th>
<th>IPASS</th>
<th>C-SDP</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmr4391</td>
<td>66</td>
<td>59</td>
<td>100/75</td>
<td>91/91</td>
<td>93/90</td>
<td>100/100</td>
<td>86/86</td>
</tr>
<tr>
<td>bmr4752</td>
<td>68</td>
<td>66</td>
<td>100/97</td>
<td>90/88</td>
<td>100/94</td>
<td>99/99</td>
<td>100/100</td>
</tr>
<tr>
<td>bmr4144</td>
<td>78</td>
<td>68</td>
<td>100/69</td>
<td>100/99</td>
<td>98/85</td>
<td>96/96</td>
<td>96/94</td>
</tr>
<tr>
<td>bmr4579</td>
<td>86</td>
<td>83</td>
<td>96/90</td>
<td>80/80</td>
<td>100/98</td>
<td>100/100</td>
<td>100/99</td>
</tr>
<tr>
<td>bmr4316</td>
<td>89</td>
<td>85</td>
<td>99/91</td>
<td>83/83</td>
<td>99/98</td>
<td>98/98</td>
<td>99/99</td>
</tr>
<tr>
<td>bmr4288</td>
<td>105</td>
<td>94</td>
<td>100/99</td>
<td>98/98</td>
<td>100/98</td>
<td>99/99</td>
<td></td>
</tr>
<tr>
<td>bmr4929</td>
<td>114</td>
<td>110</td>
<td>100/100</td>
<td>93/91</td>
<td>100/100</td>
<td>100/98</td>
<td></td>
</tr>
<tr>
<td>bmr4302</td>
<td>115</td>
<td>107</td>
<td>100/100</td>
<td>96/95</td>
<td>100/99</td>
<td>99/99</td>
<td></td>
</tr>
<tr>
<td>bmr4670</td>
<td>120</td>
<td>102</td>
<td>100/100</td>
<td>96/95</td>
<td>98/97</td>
<td>98/97</td>
<td></td>
</tr>
<tr>
<td>bmr4353</td>
<td>126</td>
<td>98</td>
<td>95/55</td>
<td>96/95</td>
<td>99/93</td>
<td>95/95</td>
<td></td>
</tr>
<tr>
<td>bmr4207</td>
<td>158</td>
<td>148</td>
<td>100/99</td>
<td>100/99</td>
<td>100/97</td>
<td>99/99</td>
<td></td>
</tr>
<tr>
<td>bmr4318</td>
<td>215</td>
<td>191</td>
<td>99/99</td>
<td>87/84</td>
<td>100/98</td>
<td>98/98</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) Number of assignable spin systems in the BMRB data.

FLYA simulated framework

The simulated framework describe for the protein SH2 in \[57\] was used to generate noisy peak lists, as validation of the peak list graph model described in \[3.2\]. In particular, artificial peak lists were generated for HSQC, HN(CO)CACB, HN(CA)CO, and HNCA spectra at the positions specified by the reference chemical shifts as listed in the corresponding BMRB entry \[61\]. Peak positions were randomly shifted according to the tolerance levels specified in \[57\], with a maximal peak shift of 0.04 ppm for \(^1\)H atoms and 0.4 ppm for \(^{13}\)C and \(^{15}\)N atoms.

The node enumerator described in \[3.2.2\] was used with only the 4 largest maximal cliques for each connected component considered as a node. The results are summarized in Table \[3.3\], where we can see that LP appears to deliver compara-
table performance to FLYA. Further testing is required to evaluate how LP copes with missing and artifact data.

Table 3.3: Percentage of correct atom assignments for LP and FLYA on simulated SH2 peak list datasets.

<table>
<thead>
<tr>
<th>Protein ID</th>
<th>Length</th>
<th>FLYA</th>
<th>LP low</th>
<th>LP high</th>
<th>LP average</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH2</td>
<td>114</td>
<td>97.2%</td>
<td>94.5%</td>
<td>97.5%</td>
<td>95.4%</td>
</tr>
</tbody>
</table>

3.4.2 Experimental data

To validate the performance of LP on experimental data, we once again resort to the experimental dataset of IPASS, which we know presents significant challenges in the form of missing and erroneous data. The relaxation LP-R2 (3.3) was used throughout, with $\lambda_{\text{r}} = 5$. Results are summarized in Table 3.4.

Table 3.4: Accuracy of assignment on four distinct spin system datasets provided by the authors of IPASS. Results for other algorithms were obtained from [4].

<table>
<thead>
<tr>
<th>Protein</th>
<th>Length</th>
<th>Manual</th>
<th>Spins*</th>
<th>MARS</th>
<th>IPASS</th>
<th>C-SDP</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM1112</td>
<td>89</td>
<td>83</td>
<td>81(74)/85</td>
<td>55/63</td>
<td>71/72</td>
<td>50/85</td>
<td>74/81</td>
</tr>
<tr>
<td>CASKIN</td>
<td>67</td>
<td>54</td>
<td>47(43)/48</td>
<td>23/25</td>
<td>29/39</td>
<td>27/48</td>
<td>36/44</td>
</tr>
<tr>
<td>VRAR</td>
<td>72</td>
<td>60</td>
<td>47(41)/47</td>
<td>6/17</td>
<td>30/37</td>
<td>19/47</td>
<td>29/51</td>
</tr>
<tr>
<td>HACS1</td>
<td>74</td>
<td>61</td>
<td>48(46)/61</td>
<td>15/16</td>
<td>37/50</td>
<td>19/61</td>
<td>39/53</td>
</tr>
</tbody>
</table>

1 Number of manually assigned residues in the BMRB file.
2 Correct/Total available spin systems
*The spin system lists available to us do not corroborate the numbers found in the original paper. The numbers in parentheses represent the total number of correct spin systems as identified by us through matching ground truth chemical shifts extracted from an NMRStar file to the list of provided spin systems.

We see that LP delivers state-of-the-art performance on this dataset, beating IPASS on 3 out of 4 proteins in terms of recall, although it tends to assign more incorrect spin systems than other algorithms. The reasons for this may lie in the thresholds selected for these tests, as LP picked out erroneous spin systems instead of assigning dummy nodes to ambiguous residues.
An important observation provided by these experiments is that relaxation LP-R2 is particularly useful when datasets are of poor quality. In fact, we observe that the final solution in all these experiments reused several of the spin systems in multiple positions, which would not have been possible under the standard ILP formulation. This illustrates the importance of correctly characterizing the quality of the dataset through appropriate constraints on the problem.

3.5 Conclusion

In this chapter, we introduced a new formulation of NMR assignment as a problem of finding a path in a directed graph. The construction of the graph described in 3.1 and 3.2 is such that a path of length \( n + 2 \) from a start node to an end node represents the likelihood of a valid assignment.

Two distinct graph construction procedures were described, to handle both spin system and peak list data. Each procedure is characterized by a node creation mechanism, and node and edge scores, which reflect assumptions on the statistical mechanisms by which the data is generated. The node creation mechanism is trivial for spin system data, but quite intricate for peak list data, highlighting the difficulty of constructing accurate spin systems. An enumerator was described for this purpose, which is of relevance as it can produce all valid combinations of measured peaks given a particular set of tolerances.

Finally, three distinct linear programming relaxation approaches were described to tackle the problem of finding a maximum likelihood path in the graph. Results on both simulated and experimental data confirmed that the relaxations produce solutions that are close to optimal, with state-of-the-art results on all data sets.
Chapter 4

Conclusion

This work described two convex optimization approaches to the problem of NMR assignment, which represent forward progress towards achieving full assignment automation.

In our first approach, the assignment problem was formulated as an instance of the quadratic assignment problem (QAP), an NP-hard problem of great interest in theoretical computer science. A new semidefinite programming relaxation to the QAP was proposed, and solved for problems as large as $n = 150$, by using a carefully designed and provably convergent ADMM scheme. The relaxation was shown to yield strong lower bounds on several sparse QAP and TSP problems. Interestingly, it also proved to retain much of the structure of the problem within its PSD variables, such that strong upper bounds (and optimal solutions) were successfully derived from the lower bound solutions. Applications to NMR assignment were met with mixed success, as C-SDP successfully recovered exact assignments in simulated benchmark data, but failed to improve upon state-of-the-art on very noisy experimental data.

An integer programming formulation of NMR assignment was proposed as an alternative to resolve several of the limitations observed in the QAP formulation of the problem. By relaxing to a linear program, efficient solutions were obtained for NMR
assignment on both spin system and peak list data. In particular, improvements upon state of the art were obtained for both simulated and experimental spin system data. Preliminary results on peak list data showed comparable performance to existing state-of-the-art. Furthermore, to the best of our knowledge, our formulation represents the most probabilistically sound representation of NMR assignment to date, and we hope that subsequent tests will reveal improvement upon state-of-the-art for peak list data as well.

4.1 Future Work

Future work is required to evaluate the quality of the LP approach on peak list data. In particular, we hope to consider more efficient sampling mechanisms to create the original graph when provided with peak list data. We also hope to develop iterative methods that can automatically resolve ambiguities in original peak lists, as we have observed that the Lagrangian relaxation tends to violate utilization constraints on peaks which are near known missing peaks. We note that the solution to the LP relaxation is extremely efficient, such that another consideration is to impose flow constraints on the edges in order to force a fractional solution around the best nodes. Preliminary tests suggest such an approach can successfully reduce the graph to a size where a MILP formulation is not just feasible, but extremely fast using branch-and-bound techniques.

Ultimately, it is our goal to develop a fully automated pipeline for NMR structural reconstruction. Armed with powerful automated tools for NMR assignment, high quality constraint sets should be within reach. An important next step will be to make use of the constraints generated from our automated assignment tools as input to structural reconstruction algorithms.
Bibliography


