

A TABU SEARCH APPROACH FOR THE NUCLEAR MAGNETIC
RESONANCE PROTEIN STRUCTURE BASED ASSIGNMENT
PROBLEM

by
Gizem Çavuşlar

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of the requirements for the degree of
Master of Science

Sabancı University

July, 2011

A TABU SEARCH APPROACH FOR THE NUCLEAR MAGNETIC
RESONANCE PROTEIN STRUCTURE BASED ASSIGNMENT
PROBLEM

Approved by:

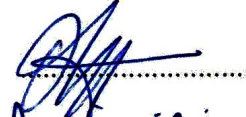
Assoc.Prof.Dr. Bülent Çatay
(Dissertation Co-Supervisor)



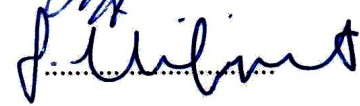
Assist.Prof.Dr. Mehmet Serkan Apaydın
(Dissertation Co-Supervisor)



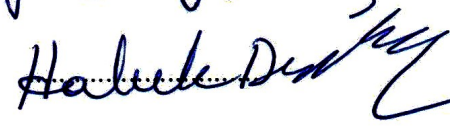
Assoc.Prof.Dr. Uğur Sezerman



Assoc.Prof.Dr. Tonguç Ünlüyurt



Clinical Prof.Dr. Haluk Demirkan



Date of Approval:04/07/2011.....

© Gizem Çavuşlar 2011

All Rights Reserved

A TABU SEARCH APPROACH FOR THE NUCLEAR MAGNETIC RESONANCE PROTEIN STRUCTURE BASED ASSIGNMENT PROBLEM

Gizem Çavuşlar

Industrial Engineering, Master's Thesis, 2011

Thesis Co-Supervisor: Bülent Çatay

Thesis Co-Supervisor: Mehmet Serkan Apaydın

Keywords: Nuclear Magnetic Resonance, Structure Based Assignments, Nuclear Vector Replacement, Metaheuristics, Tabu Search.

Abstract

Nuclear Magnetic Resonance (NMR) Spectroscopy is an experimental technique which exploits the magnetic properties of specific nuclei and enables the study of proteins in solution. The key bottleneck of NMR studies is to map the NMR peaks to corresponding nuclei, also known as the assignment problem. Structure Based Assignment (SBA) is an approach to solve this computationally challenging problem by using prior information about the protein obtained from a homologous structure. [17] used the Nuclear Vector Replacement (NVR) [29] framework to model SBA as a binary integer programming problem (NVR-BIP). In this thesis, we prove that this problem is NP-hard and propose a tabu search algorithm (NVR-TS) equipped with a guided perturbation mechanism to efficiently solve it. NVR-TS uses a quadratic penalty relaxation of NVR-BIP where the violations in the Nuclear Overhauser Effect constraints are penalized in the objective function. Experimental results indicate that our algorithm finds the optimal solution on NVR-BIP's data set which consists of 7 proteins with 25 templates (31 to 126 residues). Furthermore, for two additional large proteins, MBP and EIN (348 and 243 residues, respectively) which NVR-BIP failed to solve, it achieves 91% and 41% assignment accuracies. The executable and the input files are available for download at <http://people.sabanciuniv.edu/catay/NVR-TS/NVR-TS.html>.

NÜKLEER MANYETİK REZONANS PROTEİN YAPI TABANLI ATAMA PROBLEMİNE TABU ARAMA YAKLAŞIMI

Gizem Çavuşlar

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2011

Tez Eş Danışmanı: Bülent Çatay

Tez Eş Danışmanı: Mehmet Serkan Apaydın

Anahtar Kelimeler: Nükleer Manyetik Rezonans, Yapı Tabanlı Atama, Nükleer Vektör Değişirme, Metasezgisel Yaklaşımlar, Tabu Arama Sezgiseli.

Özet

Nükleer Manyetik Rezonans (NMR) spektroskopisi, belirli atomların manyetik özelliklerinden yararlanarak protein yapısının çözelti içinde çalışılmasını sağlayan deneysel bir yöntemdir. NMR çalışmalarında en büyük engel, NMR tepelerini karşılık gelen atomlara atama problemidir. Yapı Tabanlı Atama (YTA), bu hesaplama açısından zorlu problemi, benzer bir proteinden elde edilen bilgiyi kullanarak çözme yaklaşımıdır. [17]'de YTA problemi Nükleer Vektör Değişirme (NVD) [29] çerçevesinde ikili tam sayı programlama problemi (NVD-ITP) olarak modellenmiştir. Bu çalışmada, verilen problemin NP-zor olduğunu ispatlayıp, bu problemi verimli çözmek için yönlendirilmiş bir karıştırma mekanizmalı tabu arama sezgiseli (NVD-TA) öneriyoruz. NVD-TA'da, NVD-ITP modelindeki Nükleer Overhauser Etkisi kısıtlarının amaç işlevinde cezalandırılmasıyla elde edilen, NVD-ITB modelinin ikinci dereceden ceza gevşetilmesi kullanılmaktadır. Deneysel sonuçlar, algoritmamızın NVD-ITB'nin 25 kalıp 7 hedef proteinden oluşan (31 - 126 amino aside sahip) veri kümesi için en iyi sonucu verdiğini göstermektedir. Ayrıca, NVD-ITB'nin çözemediği iki büyük proteinden biri olan MBP için 91%, diğeri olan EIN için 41% doğrulukta (sırasıyla 348 ve 243 amino asitli) doğrulukta sonuçlar vermektedir. Çalıştırılabilir yazılım dosyası ve giriş verileri <http://people.sabanciuniv.edu/catay/NVR-TS/NVR-TS.html> adresinden elde edilebilir.

“to my family”

Acknowledgements

I would like to express my deepest gratitude to my co-supervisors, Assoc.Prof.Dr. Bülent Çatay and Assist.Prof.Dr. Mehmet Serkan Apaydın for their invaluable advice and supervision throughout this study. This thesis would not have been possible without their endless support and great patience.

I am thankful to my thesis committee, Assoc.Prof.Dr. Uğur Sezerman, Assoc.Prof.Dr. Tonguç Ünlüyurt and Clinical Prof.Dr. Haluk Demirkan for their review.

I would like to show my gratitude towards the Scientific and Technological Research Council of Turkey (TUBITAK) for the BIDEB scholarship that provided me the necessary financial resources throughout this thesis. This work was also supported by following grants to Assist.Prof.Dr. Mehmet Serkan Apaydın: TUBITAK research support program (program code 1001) [109E027] and EU Marie Curie Grant PIRG05-GA-2009-249267.

I am deeply thankful to my friends, Halit Erdoğan and Tansel Uras, for helpful discussions and comments on the complexity proof of the NMR-Resonance Assignment problem, as well as for their moral support.

I am indebted to all my friends from Sabancı University for their motivation and endless friendship. My special thanks go to Nurşen, Çetin, Mahir, Yeliz, Can, İbrahim, Nükte, Volkan, Semih, Birce, Gizem and Elif.

Last but not the least; I owe my deepest gratitude to my uncle Salih Güran for his invaluable support and guidance; to my parents Mehmet Güran and Fatma Güran and to my sister, Özge Çavuşlar for their unconditional love, support and persistent confidence in me.

Contents

1	Introduction	1
2	Nuclear Vector Replacement Framework	4
2.1	Problem Definition	4
2.2	NVR-EM	5
2.3	NVR-BIP Formulation	6
2.4	Complexity of NMR Resonance Assignment Problem	7
3	Proposed Tabu Search Solution Approach	10
3.1	Tabu Search	10
3.1.1	Heuristic Approaches	10
3.1.2	Basic Concepts in Tabu Search	11
3.1.3	Advanced Concepts in Tabu Search	13
3.2	Our Tabu Search Approach	14
3.2.1	Quadratic Penalty Relaxation of NVR-BIP	14
3.2.2	NVR-TS Algorithm	16
4	Experimental Study	19
4.1	Data Preparation	19
4.2	Experimental Design and Parameter Setting	20
4.3	Computational Results	21
4.4	Extension to k – <i>best</i> Solutions	23
5	Conclusion and Future Work	26
A	Pseudocode of NVR-TS Algorithm	31
B	Additional Accuracy Results	33

List of Figures

2.1	Illustrated example of NOE constraints.	5
3.1	Illustration of standard tabu list	12
3.2	Illustration of NVR-TS algorithm.	17

List of Tables

4.1	Parameter settings	20
4.2	Percent accuracy results on NVR-BIP's data set	21
4.3	Percent accuracy results on MBP and EIN.	21
4.4	Percent accuracy results of additional tests on MBP and EIN using different parameter values.	22
4.5	Percent accuracy comparison between the best solution and the most accurate solution in k – <i>best</i> solutions.	24
4.6	Percent accuracy comparison between the best solution and the most accurate solution in k – <i>best</i> solutions for MBP and EIN using different parameter values.	25
B.1	Percent accuracy results of 10 runs on Ubiquitin proteins	33
B.2	Percent accuracy results of 10 runs on SPG proteins	33
B.3	Percent accuracy results of 10 runs on Lysozyme Proteins	34
B.4	Percent accuracy results of 10 runs on the rest of the proteins	34
B.5	Percent accuracy results of 10 runs on MBP with several parameter sets	35
B.6	Percent accuracy results of 10 runs on EIN with several parameter sets	35

List of Abbreviations

BIP Binary Integer Programming

EIN Amino Terminal Domain of Enzyme I from Escherichia Coli

EM Expectation Maximization

ff2 The FF Domain 2 of human transcription elongation factor CA150 (RNA polymerase II C-terminal domain interacting protein)

hSRI Human Set2-Rpb1 Interacting Domain

ILP Integer Linear Programming

IP Integer Programming

LP Linear Programming

LS Local Search

MBP Maltose Binding Protein

NMR Nuclear Magnetic Resonance

NOE Nuclear Overhauser Effect

NVR Nuclear Vector Replacement

PDB Protein Data Bank

QAP Quadratic Assignment Problem

RA Resonance Assignment

RDC Residual Dipolar Coupling

RNA Ribonucleic Acid

SBA Structure-Based Assignments

SPG Streptococcal Protein G

SVD Singular Value Decomposition

TS Tabu Search

Chapter 1

Introduction

Proteins are one of the major macromolecules that are present in all biological organisms. They are composed of amino acids linked with each other by peptide bonds. When two amino acids form a peptide bond, a water (H_2O) molecule is released and the remainder of each amino acid is called amino acid residue. The resulting chain of amino acids, furthermore, are called polypeptide chains. Proteins are formed by one or more polypeptide chains.

Proteins are located within the cell, on the membrane of the cell, or outside of the cell, performing numerous functions such as catalyzing the biochemical reactions, transporting and storing chemical compounds, signaling and translating the information from other proteins, maintaining the structures of biological components (e.g. cells, tissues), converting chemical energy into mechanical energy causing muscular movement and generating immune responses to the harmful foreign bodies within the organism. Which function a protein assumes depends on its structure. Therefore, determining protein structure is of utmost importance for protein design studies. In pharmaceutical and biotechnological industry, as well as medicine, the ability to precisely engineer proteins to perform existing functions under a wider range of conditions, or to perform entirely new functions, has tremendous potential.

In order to determine protein structure, several experimental methods have been developed. X-Ray Crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy are the major experimental techniques for obtaining structural information.

X-Ray Crystallography is a method to determine the arrangement of atoms within a crystal and therefore requires crystallizing the protein. NMR is a widely used technique to determine the 3D structure of a protein in atomic detail as well as its dynamics. Unlike X-Ray Crystallography, protein structure is studied under nearly physiological conditions. This feature of NMR provides structural information about proteins that cannot be crystallized.

During an NMR spectroscopy experiment, the protein is applied electromagnetic signals

which cause the nuclei to absorb energy from the electromagnetic pulse and radiate this energy back. The resulting signals are recorded and converted into a spectrum. In the resulting spectrum, each peak corresponds to a tuple of atomic nuclei. The NMR resonance assignment problem consists of mapping the peaks to the corresponding atoms and is one of the major bottlenecks in NMR protein structure determination. Although computational methods are developed to overcome this bottleneck, NMR spectroscopists still rely on manual methods to perform the assignments as these methods are unreliable especially with large proteins.

Structure Based Assignment (SBA) is an approach to solve this challenging problem by using prior information about the protein obtained from a homologous (similar) structure. It resembles the molecular replacement technique, which solves the phase problem in X-Ray Crystallography by using a homologous structure and determines the structure rapidly and accurately [19]. An automated SBA procedure will similarly be helpful since it not only accelerates the structure determination but it is also able to reduce the amount of data needed for reliable assignments. In addition, it may be more accurate and robust.

There exists several SBA algorithms in the literature. CAP [24], which is a Ribonucleic Acid (RNA) assignment algorithm, performs an exhaustive search. Some algorithms use Residual Dipolar Coupling (RDCs) and triple resonance experiments [26, 34]. Nuclear Vector Replacement (NVR) [29, 12] is a molecular replacement-like approach for SBA. NVR does not use triple resonance experiments, but instead utilizes experimental data which can be obtained faster. In addition, its computation has a polynomial time complexity. [33] proposes a Branch-Contract-and-Bound search algorithm whereas [30] proposes a Genetic Algorithm to solve the SBA problem. NVR-BIP [17] is a tool which uses NVR's scoring function and data types to formulate a binary integer programming (BIP) model to the NMR-Resonance Assignment (NMR-RA) problem. It also incorporates CH RDCs into NVR. However, it is unable to solve the assignments for large proteins.

The primary purpose of the present study is to develop a tabu search (TS) approach to solve the NVR-BIP problem introduced in [17]. Specifically, the contributions are as follows:

- We prove that finding the set of assignments with the minimum total assignment score within the NVR framework is NP-hard.
- We propose an efficient TS (NVR-TS) algorithm equipped with a guided perturbation mechanism. To the best of our knowledge, this is the first application of TS to the NMR-RA problem.
- We add quadratic terms to the objective function to relax the Nuclear Overhauser Effect (NOE) constraints and penalize the NOE violations. By this way, we allow NVR-TS to

search through infeasible neighborhoods violating the NOE constraints in an attempt to reach improved feasible solutions.

- Finally, we test our algorithm on NVR-BIP's data set. Our results show that NVR-TS algorithm can efficiently find NVR-BIP's optimal solutions. In addition, we solve the assignments for two large proteins which cannot be solved with NVR-BIP.

The remainder of the thesis is organized as follows: In Chapter 2, we define the NMR-RA problem, and explain the NVR-BIP model proposed by [17]. Then, we prove that the problem is NP-hard and it cannot be solved by the exact solution methods for the larger proteins. Thus, a heuristic approach is indeed necessary. We present some background for TS metaheuristic. We then propose a Quadratic Penalty Relaxation of the NVR-BIP model that will enable us to design a TS heuristic for the NMR-RA problem and explain our TS algorithm in Chapter 3. Chapter 4 includes the experimental studies of our algorithm. Finally, we present our concluding remarks in Chapter 5.

Chapter 2

Nuclear Vector Replacement Framework

NMR-RA problem, which is the problem of correctly assigning the experimentally determined NMR resonances (peaks) to the correct amino acids, can be solved by exploiting the information from a homologous structure with NVR. In this chapter, we define the NMR-RA problem and present related work in the NVR framework. Then, we provide the complexity proof of the NMR-RA problem.

2.1 Problem Definition

NVR is an SBA framework in which the goal is to find a mapping between the set of peaks and the set of amino acids which minimizes the total mapping cost subject to the NOE constraints. NVR associates an assignment probability to each peak-amino acid matching which is converted into assignment cost. The interested reader is referred to [17] for detailed information.

The NOE constraints differentiate the NMR-RA problem from the General Assignment Problem which requires a set of entities to be mapped to another set of entities while minimizing the total mapping cost. In NMR-RA problem, each peak pair has a binary relation called NOE relation, i.e. for any given two peaks they either have an NOE relation or not. The amino acids also have a similar binary relation, i.e. for any given two amino acids, the distance between the (amide) protons of the amino acids is either less than a threshold value (NTH) or not. The NOE constraints imply that for any given a pair of peak - amino acid assignments (e.g. $p_i \rightarrow a_i$ and $p_j \rightarrow a_j$), if p_i and p_j have an NOE relation, then the distance of the protons of the amino acids that are assigned to those peaks, a_i and a_j , must be less than the threshold value.

Figure 2.1 is an illustration of NOE constraints. The arcs between the peak nodes repre-

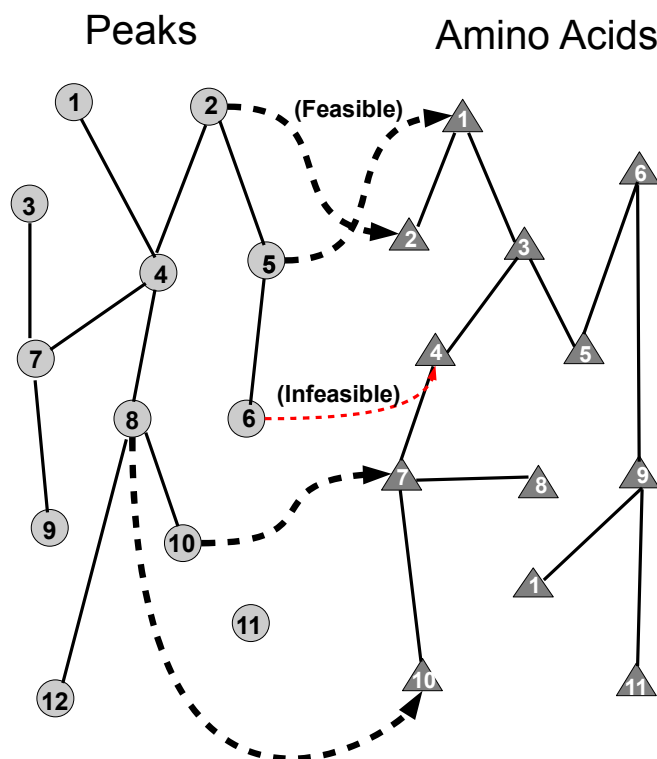


Figure 2.1: Illustrated example of NOE constraints. See the text for the explanation.

sent the NOE relation between the corresponding a pair of peaks. Similarly, two amino acid nodes have an arc in between if the distance of their amide protons is less than the threshold value. For example, peaks 2 and 5 have an arc in between implying they have an NOE relation, and they are mapped to amino acids 2 and 1, respectively. Amino acids 2 and 1 also have an arc in between implying their distance from each other is under the threshold value. Hence, assigning peaks 2 and 5 to amino acids 2 and 1, respectively, is feasible. On the other hand, peaks 5 and 6 also have an NOE relation. However, the amino acids that are assigned to them, amino acids 1 and 4 do not have an arc in between which means the distance between their amide protons is more than the threshold value. Thus, the assignments of peak 5 to amino acid 1 and peak 6 to amino acid 4 cause infeasibility.

2.2 NVR-EM

The NVR-Expectation Maximization (NVR-EM, [12]) is an approach to solve the NMR-RA problem under the NVR framework. In [12], the authors represent the problem as a maximum

bipartite matching problem with one set of nodes in the bipartite graph corresponding to the peaks and the other set of nodes corresponding to the amino acids. The probability of assigning a peak to an amino acid is represented as the weight of the edges in the bipartite graph. Based on these assignment probabilities, they generate the NVR scoring function, which they use in their expectation maximization algorithm.

2.3 NVR-BIP Formulation

Linear Programming (LP) is an optimization technique which optimizes a linear objective function subject to linear equality or inequality constraints. The objective function and the constraints are constructed by the decision variables and the parameters. Although both the decision variables and the parameters are numbers, parameters are known to the decision maker and must be taken as constant points (i.e. they do not change), whereas the value of the decision variables are determined throughout the optimization process ([4], [1]).

Integer Programming (IP) or Integer Linear Programming (ILP) is a special case of LP where all decision variables are required to have integer values. If the decision variables are allowed to be only 0 and 1, it is called Binary Integer Programming (BIP).

In [17], the NMR-RA problem is formulated as BIP. The formulation introduced is as follows:

Parameters

P: Set of peaks

A: Set of amino acids

s_{ij} : Score associated with assigning peak i to amino acid j

N : Number of peaks to be assigned ($N \leq |P|$)

d_{jl} : Distance between amide protons of amino acids j and l

$NOE(i)$: Set of peaks that have an NOE with peak i

NTH : Distance threshold for an NOE interaction

$$b_{jl} = \begin{cases} 1 & \text{if } d_{jl} \geq NTH \\ 2 & \text{otherwise} \end{cases}$$

Decision Variables:

$$x_{ij} = \begin{cases} 1 & \text{if peak } i \text{ is assigned to amino acid } j \\ 0 & \text{otherwise} \end{cases}$$

Mathematical Model:

$$\text{Min} \sum_{i \in P} \sum_{j \in A} s_{ij} x_{ij} \quad (2.1)$$

$$\text{s.t.} \sum_{i \in P} x_{ij} \leq 1 \quad \forall j \in A \quad (2.2)$$

$$\sum_{j \in A} x_{ij} \leq 1 \quad \forall i \in P \quad (2.3)$$

$$\sum_{i \in P} \sum_{j \in A} x_{ij} = N \quad (2.4)$$

$$x_{ij} + x_{kl} \leq b_{jl} \quad \forall j, l \in A, \forall i \in P, \forall k \in \text{NOE}(i) \quad (2.5)$$

$$x_{ij} \in (0, 1) \quad \forall i \in P, \forall j \in A \quad (2.6)$$

In the NVR-BIP model, the objective is to minimize the total score associated with mapping NMR peaks to amino acids. Constraints (2.2) make sure that each amino acid is assigned to at most one NMR peak while constraints (2.3) ensure that each NMR peak is mapped to at most one amino acid. Constraint (2.4) determines the number of NMR peak-amino acid assignments. Although N is usually equal to the number of peaks, in rare cases, mapping all of the peaks could be infeasible. In such cases, partial solutions can be obtained by using N as a control parameter. Constraints (2.5) are the NOE constraints. Finally, constraints (2.6) ensure that the decision variables take only binary values.

2.4 Complexity of NMR Resonance Assignment Problem

To prove that NMR-RA problem is NP-hard, we first define the feasibility problem as finding a feasible solution, i.e. a list of assignments that satisfy the NOE constraints. We prove that the feasibility problem is NP-complete by a reduction from the 3 – *Coloring* problem which is known to be NP-complete ([10]). Then, we will illustrate that feasibility problem can be easily reduced to the NMR-RA problem. Therefore, finding an optimal solution to the NMR-RA problem is NP-hard.

The NMR-RA Feasibility Problem: Given two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $|V_1| = |V_2|$, find a bijective function $f : V_1 \rightarrow V_2$ s.t. for any $v, u \in V_1$; if $(v, u) \in E_1$, then $(f(v), f(u)) \in E_2$.

The NMR-RA Problem: Given two undirected graphs $G'_1 = (V'_1, E'_1)$ and $G'_2 = (V'_2, E'_2)$ with $|V'_1| = |V'_2|$, the weight function $w : V'_1 \times V'_2 \rightarrow R^+$, and the bijective function $f' : V'_1 \rightarrow V'_2$ s.t. for any $v, u \in V'_1$; if $(v, u) \in E'_1$, then $(f'(v), f'(u)) \in E'_2$. Find a bijective function f' such that $\kappa = \sum_{v \in V'_1} w(v, f'(v))$ is minimized.

3-Coloring Problem: Given an undirected graph $G = (V, E)$, find a function $c : V \rightarrow \{Red, Green, Blue\}$ s.t. for any $v, u \in V$, if $(v, u) \in E$, then $c(v) \neq c(u)$.

Proposition 1. *The NMR-RA problem under the NVR framework is NP-hard.*

Proof. *The NMR-RA feasibility problem is in NP (membership).* It is in NP since we can guess a candidate function g and verify in polynomial time that g is a solution.

Reduction (hardness) We will reduce the 3-Coloring problem to our problem. Given a coloring problem with an undirected graph $G = (V, E)$, the NMR-RA feasibility problem with the graphs G_1 and G_2 is constructed as follows:

$$V_1 = V \cup V_D \text{ where } V_D \text{ is a set of (dummy) vertices such that } |V_D| = 2|V|,$$

$$E_1 = E,$$

$$V_2 = V_R \cup V_G \cup V_B \text{ where } V_R = \{red_1, red_2, \dots, red_{|V|}\}, V_G = \{green_1, green_2, \dots, green_{|V|}\}, \\ V_B = \{blue_1, blue_2, \dots, blue_{|V|}\}.$$

$$E_2 = \{(v, u) : v \in V_R, u \in V_G\} \cup \{(v, u) : v \in V_R, u \in V_B\} \cup \{(v, u) : v \in V_B, u \in V_G\}$$

The reduction is polynomial since we use $6|V|$ vertices and $3|V|^2 + |E|$ edges.

We need to show that 3-Coloring problem has a solution if and only if the NMR-RA feasibility problem has a solution.

- The NMR-RA feasibility problem has a solution \rightarrow 3-Coloring problem has a solution

Given a solution to the NMR-RA feasibility problem, we can extract the solution to the respective 3-Coloring problem as follows:

$$c(v) = \begin{cases} red & \text{if } f(v) \in V_R \\ green & \text{if } f(v) \in V_G \\ blue & \text{otherwise} \end{cases}$$

where $v \in V_1 \setminus V_D$

We know that all vertices are labeled and no adjacent vertices are assigned to the same color.

- 3-Coloring problem has a solution \rightarrow the NMR-RA feasibility problem has a solution

Given a solution to 3-Coloring problem, we can extract the solution to the respective NMR-RA feasibility problem as follows. For each $v_1 \in V_1 \setminus V_D$ where $c(v_1) = \textit{red}$, select a vertex from V_R to assign v_1 ; for each $v_2 \in V_1 \setminus V_D$ where $c(v_2) = \textit{green}$, select a vertex from V_G to assign v_2 ; for each $v_3 \in V_1 \setminus V_D$ where $c(v_3) = \textit{blue}$, select a vertex in V_B to assign v_3 . Since $|V_R| = |V_G| = |V_B| = |V|$ it is possible to have a one-to-one assignment. For each vertex $v_d \in V_d$, assign random vertices from $V_R \cup V_G \cup V_B$ which are not assigned to the vertices in $V_1 \setminus V_D$. Then, given any two vertices $u, v \in V_1$, if $(u, v) \in E_1$, obviously $(f(u), f(v)) \in E_2$ since $c(u) \neq c(v)$.

For an instance of the NMR-RA feasibility problem, new graphs $G'_1 = (V'_1, E'_1)$ and $G'_2 = (V'_2, E'_2)$ where $G'_1 = G_1$ and $G'_2 = G_2$ with the functions $f' = f$ and $w : V'_1 \times V'_2 \rightarrow 1$ are constructed. The NMR-RA feasibility problem has a solution if and only if the NMR-RA problem has a solution. If the NMR-RA feasibility problem has a solution, then there exists a solution for NMR-RA problem where the minimum $\kappa = |V'_1|$. Similarly, if NMR-RA problem has a solution, then there exists a solution for NMR-RA feasibility problem. Hence, finding a solution for the NMR-RA problem where $\kappa < k$ and $k \in R^+$ is NP-complete and finding a solution with the minimum κ is NP-hard. \square

Chapter 3

Proposed Tabu Search Solution Approach

Large NP-hard optimization problems cannot be solved by exact solution algorithms which spend excessive amount of time and memory to establish the optimal solution. Hence, for an NP-hard optimization problem such as NMR-RA, the NVR-BIP approach [17] which uses an exact solution algorithm to solve the proposed BIP model will fail to solve larger proteins. Therefore, an efficient heuristic/metaheuristic solution approach is necessary to establish the assignments for larger instances.

In the following sections, first we will present preliminaries for TS metaheuristic. Then we propose a quadratic penalty relaxation of NVR-BIP model which allows us to design a more effective TS algorithm. Finally, we describe the mechanisms of the developed TS algorithm.

3.1 Tabu Search

3.1.1 Heuristic Approaches

Over the past decades, heuristic methods have been developed to solve computationally challenging combinatorial optimization problems. Regardless of the method employed, the basic concepts are common to every heuristic technique. Representation of a potential solution has a great impact on the search space (the set of all possible solutions) and its size. A poorly defined search space may have several infeasible or duplicate solutions which leads to low quality search results. Another common concept is the evaluation function. It measures the fitness of a particular solution to the objective of the search. By allowing to compare one solution to another, it enables the algorithm to differentiate the good solutions from the bad ones. Finally, the concept of neighborhood remains the same in almost every heuristic

method even though its definition may vary. The neighborhood of a given solution is the set of solutions which are generated by a partial change of the given solution. The solutions in the neighborhood set of the given solution are the neighbors of the given solution. Most of the heuristic methods work by moving from one solution to its neighbor while searching for the solution which fits the evaluation function the highest. The interested reader may refer to [13] for more information.

Local Search (LS) is an heuristic method which starts with an initial solution and moves to neighbor solution if the neighbor solution's score (computed by the evaluation function) is better than the incumbent solution's score. If the incumbent solution has a better solution than the neighbor solution, then the search terminates with the incumbent solution as the local optimum solution. Algorithm 1 is the pseudocode of the generic LS.

Algorithm 1 Generic Local Search Algorithm

```
Obtain an initial solution  $x$ 
while Local optimum has not been reached do
  Generate set of neighbors  $N(x)$ 
   $x' \in N(x)$ 
  if  $f(x') < f(x)$  then
     $x \leftarrow x'$ 
  else
    Local optimum has been reached
  end if
end while
Return  $x$ 
```

LS algorithms terminate with the first solution that has a better score than all of its neighbors. This local best solution, however, may not be better than another local best solution which the search would return if the initial solution changed. In other words, the search gets stuck on a local optimum point in a search space which may not be a satisfactory enhancement over the initial solution.

3.1.2 Basic Concepts in Tabu Search

Glover proposed TS ([6] and [7]) to allow LS methods to avoid getting stuck on local optima. TS uses an LS procedure to iteratively move from one solution to its neighbor until a stopping criterion has been reached. One of the major differences between generic LS and TS is that the TS moves from the incumbent solution to its best neighbor even though its best neighbor is worse than the incumbent solution. Then, the move becomes tabu for a limited number of iterations. The tabu moves, as the name suggests, are forbidden to prevent the search from

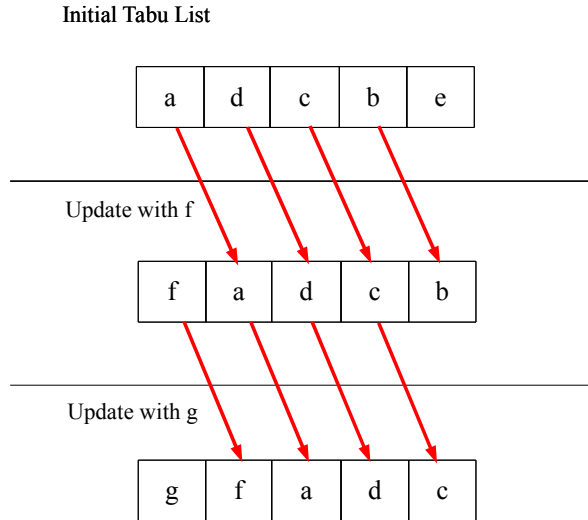


Figure 3.1: Illustration of standard tabu list

cycling to the previously visited solution and enables the search to move away from the local optimal solution ([5]).

Tabu moves are recorded in a *short term memory* or *tabu list* of the search for the limited number of iterations. The information stored on the tabu list should be sufficient to differentiate the previously visited solution from the solutions that have not been explored. The standard tabu lists are usually implemented as a circular list with a fixed length.

In Figure 3.1, a standard tabu list implementation is presented. The length of the tabu list (*tabu tenure*) represents the amount of memory allocated for the tabu list. The list in the figure is capable of carrying 5 moves which means a move stays tabu for 5 consecutive iterations when it becomes tabu. Every time the algorithm proceeds to the best neighbor, the information related to the move is kept in the tabu list. In the figure, the initial tabu list is a tabu list from a random iteration during the search. At the end of each iteration, a new move becomes tabu and enters the list while shifting all existing items on the list. The size of the tabu list is kept constant by removing the last item during the shifting process.

Keeping moves tabu may sometimes be disadvantageous since they may prohibit attractive moves while there is no risk of cycling and may cause the termination of the search with a poor quality solution. To subdue the negative effect of the tabu list, *Aspiration Criteria*, which provide the conditions to allow a move when it is still tabu, are implemented to TS. One of the most frequently encountered aspiration criteria is to allow a move if it provides a better solution than the best solution obtained so far. More complicated aspiration criteria implementations have been proposed by [2] and [9].

Algorithm 2 Generic Tabu Search Algorithm

Obtain an initial solution x
while Stopping criterion has not been reached **do**
 Generate set of neighbors $N(x)$
 $x' \in N(x)$
 if $x' = \min\{x_i : x_i \in N(x) \forall i \in |N(x)|\}$ and x' is not tabu **then**
 $x \leftarrow x'$
 Update the tabu list
 end if
end while

3.1.3 Advanced Concepts in Tabu Search

The generic TS may perform adequately for difficult problems. However, in general, additional strategies are implemented to boost the performance of the TS algorithm. Some of these strategies are as follows:

Intensification is a procedure of focusing on the more promising areas of the search space in order not to skip high quality solutions that lay on those areas. Intensification procedure is usually based on some *long term memory* such as *recency memory* which keeps the number of consecutive iterations that the particular solution components are observed in the current solution. The higher the ranking of a particular component in the memory, the more promising the component becomes to the search. A characteristic approach for intensification is to fix the high ranking components in the best solution found so far and restart to search with that solution. Although it is commonly employed in the literature, intensification is not always mandatory since the search may explore the attractive areas thoroughly enough to obtain the local best solution of the specific area.

As most of the local search based heuristics, TS is only able to search a very limited portion of the search space, missing more interesting parts with more promising solutions. Thus, although the search results in a considerably good solution, the solution may likely be still far away from the optimal solution. To remedy this drawback, implementation of the appropriate diversification (also referred as perturbation) strategies is of paramount importance. The diversification procedure enables the search to jump into the unexplored areas of the search space by making a partial change in a specific solution and start the search from that solution. It is usually based on a *long term memory* called *frequency memory* which records the number of times a particular component changes. A high rank of the component in the memory indicates that component is a “crack filler”, which sways back and forth into the solution during the search. Other types of long term memories can also be implemented to extract information. The interested reader may refer to [8] and [5] for further information.

Diversification can also be fulfilled continuously during the search. In the continuous diversification procedure, diversification is integrated in the regular search process by adding a memory based term into the evaluation function.

A third way of accomplishing diversification is the dynamic oscillation which enables exploration of the new portions of the search space by constraint relaxation. Thus, a larger search space that can be explored by simpler neighborhood structures is achieved. The constraint relaxation is performed by removing the chosen constraints from the search space and adding them into the evaluation function with a weighted penalty. The penalty term can be determined at the beginning or can be implemented as a self adjusting mechanism during the search.

3.2 Our Tabu Search Approach

3.2.1 Quadratic Penalty Relaxation of NVR-BIP

In this study, we develop a TS algorithm for solving the NMR-RA problem. The implementation of the algorithm is based on the relaxation of the NOE constraints in the NVR-BIP model, which we refer to as quadratic penalty relaxation formulation. In NVR-BIP, the NOE relations are considered as hard constraints which prohibit any solution with NOE violations. In contrast, in the relaxed formulation, NOE violations are allowed by adding terms to the objective function that penalize the NOE violations. NOE constraints (2.5) are removed and a penalty term associated with the violation of the NOE constraints is inserted into the objective function. Since we have a minimization objective, any solution with an NOE violation is discouraged by the positive penalty term. The relaxed formulation is as follows:

NVR-Quadratic Penalty Formulation:

$$\text{Min} \sum_{i \in P} \sum_{j \in A} s_{ij} x_{ij} + \sum_{i \in P} \sum_{k \in \text{NOE}(i)} \sum_{j \in A} \sum_{l \in A} p_{jl} x_{ij} x_{kl} \quad (3.1)$$

$$\text{s.t.} \sum_{i \in P} x_{ij} = 1 \quad \forall j \in A \quad (3.2)$$

$$\sum_{j \in A} x_{ij} = 1 \quad \forall i \in P \quad (3.3)$$

$$x_{ij} \in (0, 1) \quad \forall i \in P, \forall j \in A \quad (3.4)$$

In the model above, the objective function (3.1) minimizes the total score associated with the assignment of NMR peaks to amino acids and the additional score (penalty) resulting from NOE relation violations. Constraints (3.2) guarantee that each amino acid is assigned

to one NMR peak and constraints (3.3) ensure that each peak is assigned to one amino acid. If the number of peaks and the number of amino acids are not equal we introduce dummy peaks or amino acids. An assignment containing a dummy peak (or amino acid) does not have a corresponding assignment score, and it does not violate any NOE constraints. Since infeasibility due to NOE constraints is no longer possible, constraint (2.5) in NVR-BIP is not needed. Finally, constraints (3.4) define the decision variables as binary.

The penalty parameter p_{jl} determines the weight of the NOE violation penalty in the objective function value. If it is too large the search will focus on satisfying the NOE constraints which may lead to less accurate solutions. On the other hand, if it is too low the search may favor solutions with NOE violations which lead to infeasible assignments. Therefore, establishing the value of the parameter p_{jl} is important. After a series of preliminary tests, we determined the p_{jl} value as follows ²:

$$p_{jl} = \begin{cases} \bar{s} & \text{if } d_{jl} > NTH \\ 0 & \text{otherwise} \end{cases}$$

where $\bar{s} = \max\{s_{ij} : i \in P, j \in A\}$.

The advantage of using the relaxed formulation approach in the TS implementation is twofold. First, the absence of the NOE constraints makes it easy to find an initial solution to start the TS algorithm since any one-to-one assignment of peaks to residues is a feasible solution. Second, it allows TS to explore NOE relation violating solutions in an attempt to find better solutions obeying the NOE constraints at the end. The relaxed problem resembles the quadratic assignment problem (QAP) as they both have the same feasible region. TS has been extensively investigated in the operations research literature for solving the QAP. Skorin-Kapov [20], [25], [21], Taillard [22], Battiti and Tecchiolli [18] are the pioneers of tabu search implementation for QAP. Their studies are followed by Misevicius [14]. Hybrid methods which combine TS with genetic algorithms are applied to QAP in [15], [3]. Recently, Tabitha *et. al.* presented a multi start TS and diversification strategies for QAP in [32]. [23] proposes a neighborhood generation structure for very large scale QAP which can be implemented as a part of TS algorithm. Also, [27] defines diversification strategies for the QAP for general metaheuristic strategies including TS.

²Note that the penalty term may also be formulated as follows: $\sum_{i \in P} \sum_{k \in NOE(i)} \sum_{j \in A} \sum_{l \in A} p * \max\{(x_{ij} + x_{kl}) - b_{jl}, 0\}$, where p is the penalty associated with violating constraints (5) and may be determined as $p = \bar{s}$.

3.2.2 NVR-TS Algorithm

In NVR-TS algorithm, we implement a dynamic tabu list structure. The tabu tenure is randomly determined between the interval of $[t_{min}, t_{max}]$ where t_{min} and t_{max} correspond to the minimum and maximum tabu tenures, respectively. Since the number of iterations that a particular move stays tabu is changing for each move, the tabu list is called a dynamic tabu structure. A similar dynamic tabu list approach was also used by [22].

We developed a novel perturbation mechanism which enables the algorithm to reach larger portions of the search space and enhances its performance considerably. The search procedure and the perturbation mechanism are illustrated in Figure 3.2. The nodes represent the set of assignments (solutions) and the plane that the nodes are located in represents the search space. The nodes are placed according to their distance (in terms of the number of moves), not according to their scores. The search begins with randomly generated solution in neighborhood 1 and moves from one solution to another by only making one move at each iteration (as shown with short arrows). During this walk, the local best score is updated whenever the search visits a solution that has a lower score than the incumbent local best score. After several non-improvements on the local best score, the local best solution is perturbed which causes a jump in the search space (as shown with the long arrows). The local search is repeated in the new region and the local best solutions of the two regions are compared to determine the global best solution. In Figure 3.2, the global best solution lies in neighborhood 3. After the perturbation of the local best (also the global best in this illustration) solution in this neighborhood, the search jumps to several regions. If the global best solution has not improved after a pre-determined number of jumps, the search returns to the global best solution (as shown with the red arrows) and makes another jump by perturbing the global best solution. Since the perturbation is stochastic, the probability of tracking the same path is very low. Returning to the global best solution after a given number of non-improving jumps prevents the algorithm from spending too much time in non-promising neighborhoods. As a result, NVR-TS is able to find high quality solutions in reasonable computational times.

We represent a feasible solution as an array s of size n which is equal to the maximum of $|P|$ and $|A|$ and $s[i] = j$ refers to peak i being assigned to amino acid j . We use a swap (pairwise) neighborhood search operator to generate the neighbor solutions. Our tabu list includes peaks in pairs. Swapping peak i with peak k (where the solution consists of $s[i] = j$ and $s[k] = l$) is *tabu* if the pair $i - k$ is in the tabu list. The number of iterations that a swap move stays in the tabu list is randomly determined between the interval of $[t_{min}, t_{max}]$ for each move.

Our algorithm has nested structures to implement the proposed perturbation mechanism.

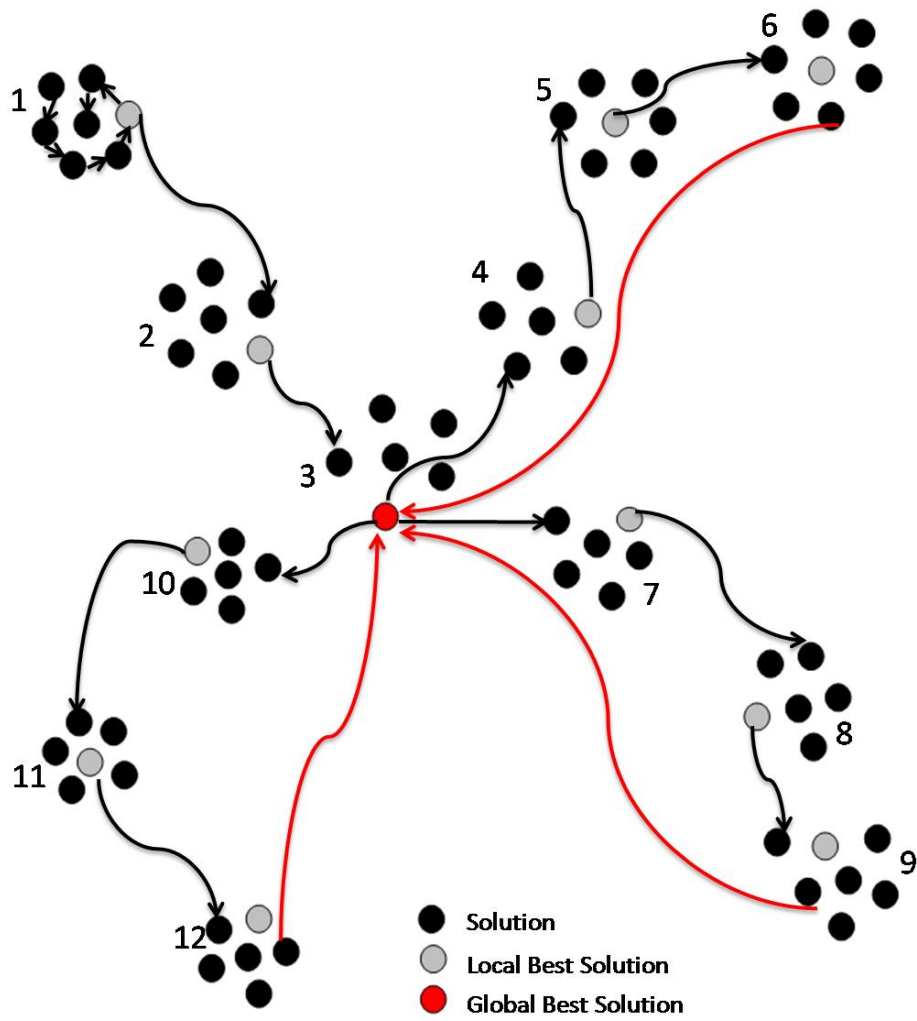


Figure 3.2: Illustration of NVR-TS algorithm.

Each nested structure has its own best-so-far solution. The innermost structure is the basic TS structure which starts with an initial solution and moves from one solution to another until its local optimum solution has not improved for a given number of consecutive iterations ($Iter_1$). The procedure is as follows: Let s be the incumbent solution and s' be the current best solution. All neighbors of s are generated by considering all possible combinations of pairwise exchanges of peaks in solution s . Let s_l be the neighboring solution with the lowest score. s is updated with s_l if the move is not tabu or it meets the aspiration criterion, i.e. $score(s_l) < score(s')$. Otherwise, the algorithm determines the non-tabu next lowest scoring solution and updates s with that solution. s' is updated whenever $score(s_l) < score(s')$. This basic search and the following perturbation can be considered as a compact mechanism which is repeated until the mechanism cannot improve the local optimum solution, s'' , for a specified number of consecutive iterations ($Iter_2$). When the mechanism fails to improve s'' , it applies a perturbation procedure to s'' and restarts the search-perturbation procedures with the perturbed s'' as the initial solution. This returning policy is also repeated until the global best solution, s^* , has not improved for $Iter_3$ consecutive iterations. The simplified pseudocode of the algorithm is presented in the Appendix A.1.

The perturbation is performed by reassigning a pre-determined ratio (r) of the current solution without violating constraints (3.2-3.4) in the quadratic penalty formulation. Which peaks will be changed is determined based on a long term memory called *transition memory* ([8]). The selected peaks are reassigned from a set of amino acids that have been assigned to these peaks. If the number of amino acids is larger than the number of peaks (i.e. there exists unassigned amino acids), those amino acids are also added to the set. The reassignments are made in a similar way to the roulette wheel method. Each assignment has a probability determined based on its score. The assignment probability and its score are inversely proportional.

Chapter 4

Experimental Study

In this chapter, we present the data preparation, parameter tuning and the computational performance of our NVR-TS implementation.

4.1 Data Preparation

We tested the performance of NVR-TS on the data set used in NVR-BIP since the scores that NVR-BIP reports are optimal. Furthermore, we tested our algorithm on two novel proteins which were not included in NVR-BIP's data set: Maltose Binding Protein (MBP) and Amino Terminal Domain of Enzyme I from Escherichia Coli (EIN).

The NOE data for MBP is simulated by selecting all pairs of amide hydrogens that are closer than 5\AA from the pdb structure (1DMB, an X-ray structure) and generating an NOE for this pair of protons. This generated a total of 574 NOE constraints for MBP. The chemical shifts and RDCs were acquired from MARS distribution [34] and NVR was extended to accept N-C and C- C_α RDCs. The HD-exchange data was simulated analogously to the proteins in NVR-BIP's data set as described in [17] for EIN. For EIN, we extracted the unambiguous NOEs from the NOE data as described in [31]. We obtained the chemical shifts from the BMRB and simulated the RDCs using the pdb structure (1ZYM, an X-ray structure). For EIN and MBP we did not simulate TOCSY data as this experiment is not plausible for large proteins. We added the protons using MOLMOL [28] for both 1DMB and 1ZYM.

The preparation of the data files for the remaining proteins is described in [12] and [17].

Parameter	Value	Experimental Design
t_{max}	αn^2	$\alpha = 0.04, 0.06, 0.08$
t_{min}	βt_{max}	$\beta = 0.5, 0.7, 0.9$
$Iter_1$	γt_{max}	$\gamma = 1.2, 1.5, 2$
r	δm	$\delta = 5\%, 10\%, 20\%, 50\%$

Table 4.1: Parameter settings

4.2 Experimental Design and Parameter Setting

Similar to most TS algorithms, NVR-TS also has several parameters such as t_{max} , t_{min} , $Iter_1$ and r where t_{max} is the maximum tabu tenure, t_{min} is the minimum tabu tenure, $Iter_1$ is the number of consecutive non-improving iterations in the innermost loop and r is the percentage of the actual assignments to be perturbed. Each of those parameters has a considerable effect on the performance of the algorithm. Therefore, after a series of preliminary tests, we designed an experimental framework to determine the best performing parameters. We call the combination of those four parameters (t_{max} , t_{min} , $Iter_1$, r) a parameter set. In Table 4.1, we present the parameter values investigated in the preliminary analysis. $Iter_2$ and $Iter_3$ are determined as 3 and 5, respectively.

In our problem, the solution size n is equal to the maximum of $|P|$ and $|A|$. Excluding r , we set the values of the parameters as a function of n^2 since the search space grows with the square of the solution size. The perturbation ratio r is determined as a function of m , where m is the minimum of $|P|$ and $|A|$. By doing so, we ensure that the actual assignments that do not contain any dummy peaks or dummy amino acids will be reassigned. To determine the best parameter values for NVR-TS we performed an initial experimental study on a subset of proteins (namely GB1, 1GB1, 2GB1, 1PGB, hSRI, pol η , ff2, 1AAR, 1G6J, 1AKI, 2LYZ) using all combinations of $(\alpha, \beta, \gamma, \delta)$. We performed 10 runs for each parameter set (i.e. 1080 runs for each protein) due to the stochastic nature of the algorithm and considered the average of the total scores to evaluate the performances. All computational tests were performed on an 2x Quad Core Xeon E7430 2.33 GHz processor with 128 GB of RAM. Based on these initial experiments, we established a parameter set $t_{max}, t_{min}, Iter_1$ and r as a function of $\alpha=4\%$, $\beta=0.7$, $\gamma=1.2$ and $\delta=10\%$, respectively, as illustrated in Table 4.1, and applied NVR-TS for all the proteins included in the NVR-BIP’s data set.

Protein Family	No of Residues	PDB ID	NVR-BIP		NVR-TS			
			without RDC (%)	with RDC (%)	Accuracy of Best Solution		Average Accuracy	
					without RDC (%)	with RDC (%)	without RDC (%)	with RDC (%)
Ubiquitin	72	1UBI	87	97	87	97	90	97
		1UBQ	87	97	87	97	91	97
		1G6J	87	97	87	97	80	81
		1UD7	81	97	81	97	83	97
		1AAR	79	97	79	97	55	86
SPG	55	1GB1	100	100	100	100	100	100
		2GB1	100	100	100	100	100	100
		1PGB	96	100	96	100	96	100
Lysozyme	126	193L	78	100	78	100	74	98
		1AKI	78	98	78	98	76	96
		1AZF	74	94	74	94	72	90
		1BGI	75	97	75	97	69	93
		1H87	77	100	77	100	72	96
		1LSC	74	100	74	100	73	98
		1LSE	75	98	75	98	73	94
		1LYZ	79	82 ^a	79	68 ^a , 82 ^b	76	65 ^a , 81 ^b
		2LYZ	75	91	75	91	75	89
		3LYZ	79	90	79	90	77	87
		4LYZ	75	91	75	91	70	87
5LYZ	75	91	75	91	72	87		
6LYZ	75	96	75	96	73	95		
The rest	80	ff2	85	93	85	93	57	93
	96	hSRI	73	89	73	89	33	62
	31	pol η	100	100	100	100	100	100
	55	GB1	96	100	96	100	96	100

^a: With one set of RDCs, ^b: With two set of RDCs.

Table 4.2: Percent accuracy results on NVR-BIP’s data set

Protein Name	No of Residues	Accuracy of Best Solution		Average Accuracy	
		without RDC (%)	with RDC (%)	without RDC (%)	with RDC (%)
MBP	348	69	79	40	62
EIN	243	5	28	2	8

Table 4.3: Percent accuracy results on MBP and EIN.

4.3 Computational Results

We report our results in terms of the accuracies of the high quality solutions. We define accuracy as the ratio of the number of correctly assigned peaks to the total number of assigned peaks.

The results for NVR-BIP’s data set are presented in Table 4.2. The column “NVR-BIP” shows the accuracies obtained in [17]. “NVR-TS (Accuracy of Best Solution)” column reports the accuracy of the best solution (the solution with the lowest score) among the 10 runs. As in [17], we report the accuracies both without and with RDC’s. Note that the results reported by NVR-BIP are optimal with respect to the assignment score and benchmarking the performance of a metaheuristic against the optimal solutions is of paramount importance. We

	Parameter Set				Accuracy of Best Solution		Average Accuracy	
	α	β	γ	δ	without RDC (%)	with RDC (%)	without RDC (%)	with RDC (%)
MBP	6%	0.5	1.5	20%	72	87	60	71
	8%	0.9	2	5%	80	90	54	66
	6%	0.7	1.2	20%	76	91	60	75
EIN	6%	0.5	1.5	20%	25	9	7	7
	8%	0.9	2	5%	16	41	3	18
	6%	0.7	1.2	20%	0	24	2	15

Table 4.4: Percent accuracy results of additional tests on MBP and EIN using different parameter values.

also report the average of the accuracies corresponding to the best solutions of 10 runs in the last two columns of Tables 4.2 and 4.3. The accuracy results of the 10 runs for every protein can be found in the Appendix.

For all of the proteins in NVR-BIP’s data set, NVR-TS finds the same assignment accuracies as NVR-BIP. In that sense, NVR-TS shows a remarkable performance as it is able to reach optimal solutions in every instance. The average computational times for Lysozyme (126 residues) and Ubiquitin (72 residues) families are 106 seconds and 17 seconds, respectively. The running time of NVR-TS is shorter or comparable to the running time of NVR-BIP for these proteins. For some proteins, the average accuracies associated with the best solutions in 10 runs is considerably high, sometimes even higher than the optimal solution’s accuracy. This indicates that close-to-best (near optimal) solutions yield higher accuracies than that of the best (optimal) solution for those proteins.

In Table 4.3, we also present the results for MBP and EIN which have 348 and 243 residues, respectively. NVR-BIP is not capable of solving the assignments for those proteins due to their sizes and NVR-EM finds a solution with 0% accuracy both for MBP and EIN. On the other hand, NVR-TS solves MBP with 79% accuracy. The average computational time is 3.8 hours. For EIN, 28% accuracy is achieved within 1 hour.

These results show the superior performance of NVR-TS on these novel proteins. However, since the accuracies were lower for EIN and MBP compared to the other proteins in our data set and NVR-TS terminated with solutions allowing NOE violations, we decided to perform additional tests for these two proteins using three other parameter sets that we observed to perform well in our preliminary parameter analysis. The results for these tests are presented in Table 4.4 and detailed results are provided in Appendix B. For MBP, an NOE feasible solution is obtained with an accuracy 91%. EIN’s accuracy remains lower even though it increases to 41% and the solution is still NOE infeasible. Although NVR-TS is able to obtain solutions with high accuracies, it may be unable to provide NOE feasible solutions

for the given $Iter_2$ and $Iter_3$ parameter values.

4.4 Extension to $k - best$ Solutions

In NVR-SBA problem, an ideal scoring function has to have a perfect correlation with the assignment accuracies, in other words, the lowest scored solution should have the highest accuracy. However, in practice, the optimal solution does not always have the highest accuracy due to the imperfection of the score functions. To remedy this, we design our algorithm to return an elite group of solutions, the $k - best$ solutions that the search has visited, instead of just one global best solution. The accuracies of those elite solutions may be higher than the accuracy of the global best or even the optimal solution. The parameter k is the number of elite solutions to be returned by NVR-TS and it is determined by the user.

During our tests on NVR-TS, we kept $k - best$ solutions where k is 30. We investigate whether any of these 30 elite solutions is more accurate than the global optimal solution that the algorithm returns. The results are reported in Table 4.5 and Table 4.6. The accuracy values shown in **bold** indicate that the accuracy of a solution in the $k - best$ list surpasses the accuracy of the global best solution. Note that in Table 4.6, the two results with 92% accuracy for MBP correspond to feasible assignments. The remaining results for both MBP and EIN in this table involve NOE infeasibilities.

We plan to use the $k - best$ solutions to analyze the frequency of individual peak-amino acid assignments to observe the common assignments in those solutions. This can potentially provide us with a confidence score for each assignment. In addition, this information can further be used for intensification purposes, i.e. fixing the common assignments in the $k - best$ solutions and restarting the search from that point.

Protein Family	No of Residues	PDB ID	Accuracy of Best Solution		Best Accuracy in $k - best$	
			without RDC (%)	with RDC (%)	without RDC (%)	with RDC (%)
Ubiquitin	72	1UBI	87	97	98	100
		1UBQ	87	97	100	100
		1G6J	87	97	100	100
		1UD7	81	97	97	100
		1AAR	79	97	96	100
SPG	55	1GB1	100	100	100	100
		2GB1	100	100	100	100
		1PGB	96	100	100	100
Lysozyme	126	193L	78	100	83	100
		1AKI	78	98	83	100
		1AZF	74	94	87	95
		1BGI	75	97	87	98
		1H87	77	100	87	100
		1LSC	74	100	84	100
		1LSE	75	98	85	100
		1LYZ	79	$68^a, 82^b$	84	81^a, 89^b
		2LYZ	75	91	85	95
		3LYZ	79	90	85	94
		4LYZ	75	91	86	93
5LYZ	75	91	86	93		
6LYZ	75	96	93	97		
The rest	80	ff2	85	93	93	96
	96	hSRI	73	89	80	96
	31	pol η	100	100	100	100
	55	GB1	96	100	100	100
	348	MBP	69	79	69	79
243	EIN	5	28	7	28	

^a: With one set of RDCs, ^b: With two set of RDCs.

Table 4.5: Percent accuracy comparison between the best solution and the most accurate solution in $k - best$ solutions. See the text for the explanation.

	Parameter Set				Accuracy of Best Solution		Best Accuracy in $k - best$	
	α	β	γ	δ	without RDC (%)	with RDC (%)	without RDC (%)	with RDC (%)
MBP	6%	0.5	1.5	20%	72	87	75	87
	8%	0.9	2	5%	80	90	83	92
	6%	0.7	1.2	20%	76	91	76	92
EIN	6%	0.5	1.5	20%	25	9	28	18
	8%	0.9	2	5%	16	41	16	46
	6%	0.7	1.2	20%	0	24	23	27

Table 4.6: Percent accuracy comparison between the best solution and the most accurate solution in $k - best$ solutions for MBP and EIN using different parameter values.

Chapter 5

Conclusion and Future Work

In this study, we implemented a TS algorithm, NVR-TS, to find the assignments for a given protein. The algorithm is based on the relaxed formulation of the model introduced in [17] where the NOE constraints are removed and their violations are penalized in the objective function. By relaxing the NOE constraints, we allow TS to explore NOE violating neighborhoods in the search space in order to reach the global best solution.

We used NVR-BIP's scoring function and tested our algorithm on NVR-BIP's data set. Our assignment accuracies were the same as or better than those of NVR-BIP's. Additionally, we used MBP and EIN to test the performance of our algorithm on large proteins and obtained an assignment accuracy of 79% and 28% for MBP and EIN, respectively. Their accuracies increase to 91% and 41%, respectively when we use different parameter sets. Although feasible assignments could be obtained for MBP, NVR-TS was unable to reach a feasible solution for EIN. It is noteworthy that NVR-BIP cannot even find a feasible solution for these proteins due to the memory problems. Also, note that the results for 1LYZ were reported with one set of RDCs as in [17]. With both set of RDCs, the problem was infeasible in [17] due to a noisy experimental RDC value, whereas we were able to compute a solution with NVR-TS. As future work, we plan to increase the robustness of NVR-TS with respect to the noise in the data. We also plan to continue our tests on large proteins to further investigate the performance of the algorithm.

Note that the assignment accuracies presented in this thesis are computed using an ideal alignment tensor which is computed using Singular Value Decomposition (SVD) based on the knowledge of the correct assignments. As such, this work provides a proof of principle. In practice, one can compute the alignment tensor using grid search as [11] and then iterate the computation of the assignments using the alignment tensor computed using SVD and the previous assignments. Our future work includes integrating this grid search with the score

matrix computation.

Another area of future research is to analyze the occurrence frequency of individual peak-residue assignments among the k -best solutions. This analysis may be used as a reliability measure similar to the confidence value in [16]. Another possible use of the frequency analysis is that it allows us to fix the most frequent assignments and to solve the remaining subset of peaks and amino acids with NVR-TS or NVR-BIP. The frequency analysis may be especially useful for large proteins. Our current research efforts also focus on improving the quality of the scoring function so as to better represent the relationship between the score value and the assignment accuracy. Hence, the best accuracy solution may rank higher in our set of extracted solutions.

Further research on the heuristic approach may focus on the variable neighborhood search method within the TS framework, which allows the use of multiple neighborhood search structures, or other efficient LS approaches. Those approaches can be combined into a hybrid method that may provide more accurate solutions faster, especially for large problems.

Bibliography

- [1] Sherali H. D. Bazaraa, M. S. and J. J. Jarvis. *Linear Programming and Network Flows*. Wiley, New York, 2nd ed. edition, 1990.
- [2] D. de Werra and A. Hertz. Tabu search techniques: A tutorial and an application to neural networks. *OR Spektrum*, 11:131–141, 1989.
- [3] Z. Drezner. Extensive experiments with hybrid genetic algorithms for the solution of the quadratic assignment problem. *Comput. Oper. Res.*, 35(3):717–736, 2008.
- [4] Sandblo C. L. Eiselt, H. A. *Linear Programming and Its Applications*. Springer-Verlag Berlin Heidelberg, 2007.
- [5] M. Gendreau. An introduction to tabu search, 1998.
- [6] F. Glover. Tabu search—part I. *RSA Journal on Computing*, 1(3):190–206, 1989.
- [7] F. Glover. Tabu search—part II. *RSA Journal on Computing*, 2(1):4–32, 1990.
- [8] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1 edition, 1997.
- [9] A. Hertz and D. de Werra. The tabu search metaheuristic: How we used it. *Annals of Mathematics and Artificial Intelligence*, 1:111–121, 1991.
- [10] R. M. Karp. Reducibility among combinatorial problems. In R. E Miller and J.W Thatcher, editors, *Complexity of Computer Computations*, pages 85–10. Plenum, New York, 1972.
- [11] C. J. Langmead and B. R. Donald. 3D-Structural Homology Detection via Unassigned Residual Dipolar Couplings. *Proc. IEEE Computer Society Bioinformatics Conference (CSB), Stanford University, Palo Alto, CA*, pages 209–217, 2003.
- [12] C. J. Langmead and B. R. Donald. An expectation/maximization nuclear vector replacement algorithm for automated nmr resonance assignments. *Journal of Biomolecular NMR*, 29(2):111–138, 2004.

- [13] Fogel D.B. Michalewicz, Z. *How to solve it: modern heuristics*. Springer-Verlag Berlin Heidelberg, 2004.
- [14] A. Misevicius. A tabu search algorithm for the quadratic assignment problem. *Comput. Optim. Appl.*, 30(1):95–111, 2005.
- [15] A. Misevicius. A fast hybrid genetic algorithm for the quadratic assignment problem. In *The 8th annual conference on Genetic and evolutionary computation (GECCO)*, pages 1257–1264, 2006.
- [16] Apaydın *et al.* Structure-based protein nmr assignments using native structural ensembles. *Journal of Biomolecular NMR*, 40(4):263–276, 2008.
- [17] Apaydın *et al.* Nvr-bip: Nuclear vector replacement using binary integer programming for nmr structure-based assignments. *The Computer Journal*, 2010.
- [18] Battiti R. and Tecchiolli G. The reactive tabu search. *INFORMS Journal on Computing*, 6(2):126–140, 1994.
- [19] M. G. Rossmann and D. M. Blow. The detection of sub-units within the crystallographic asymmetric unit. *Acta Crystallographica*, 15(1):24–31, 1962.
- [20] J. Skorin-Kapov. Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, 2(1):33–45, 1990.
- [21] J. Skorin-Kapov. Extensions of a tabu search adaptation to the quadratic assignment problem. *Comput. Oper. Res.*, 21(8):855–865, 1994.
- [22] E. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991.
- [23] Ahuja *et al.* Very large-scale neighborhood search for the quadratic assignment problem. *INFORMS Journal on Computing*, 19(4):646–657, 2007.
- [24] Al-Hashimi *et al.* Towards structural genomics of rna: Rapid nmr resonance assignment and simultaneous rna tertiary structure determination using residual dipolar couplings. *J. Comp. Bio.*, 318(3):637–649, 2002.
- [25] Chakrapani *et al.* Massively parallel tabu search for the quadratic assignment problem. *Annals of Operations Research*, 41(1-4):327–341, 1993.
- [26] Hus *et al.* Assignment strategy for proteins of known structure. *J. Mag. Res.*, 157(1):119–125, 2002.

- [27] Kelly *et al.* A study of diversification strategies for the quadratic assignment problem. *Comput. Oper. Res.*, 21(8):885–893, 1994.
- [28] Koradi *et al.* Molmol: A program for display and analysis of macromolecular structures. *Journal of Molecular Graphics*, 14(1):51–55, 1996.
- [29] Langmead *et al.* A polynomial-time nuclear vector replacement algorithm for automated nmr resonance assignments., booktitle = the seventh annual international conference on research in computational molecular biology (RECOMB), year = 2003, pages = 176–187,.
- [30] Lin *et al.* Gana-a genetic algorithm for nmr backbone resonance assignment. *Nucleic Acids Research*, 33(14), 2005.
- [31] Stratmann *et al.* Noenet-use of noe networks for nmr resonance assignment of proteins with known 3d structure. *Bioinformatics*, 25(4):474–481, 2009.
- [32] Tabitha *et al.* Multistart tabu search and diversification strategies for the quadratic assignment problem. *Trans. Sys. Man Cyber. Part A*, 39(3):579–596, 2009.
- [33] Vitek *et al.* Reconsidering complete search algorithms for protein backbone NMR assignment. *Bioinformatics*, 21(2), 2005.
- [34] Jung Y. and Zweckstetter M. Backbone assignment of proteins with known structure using residual dipolar couplings. *Journal of Biomolecular NMR*, 30(1):25–35, 2004.

Appendix A

Pseudocode of NVR-TS Algorithm

Notation

s : initial solution

s_l : non-tabu neighbor solution with the lowest score

s' : the best solution that the innermost loop can achieve

s'' : the best solution that the middle loop can achieve

s^* : the best solution that the outermost loop can achieve (global best solution)

ctr' : the consecutive non-improving iteration counter of the innermost loop

ctr'' : the consecutive non-improving iteration counter of the middle loop

ctr^* : the consecutive non-improving iteration counter of the outermost loop

$Iter_1$: The parameter that determines the number of consecutive non-improving iterations that the innermost loop will terminate after

$Iter_2$: The parameter that determines the number of non improving consecutive perturbations that will be performed until the middle loop terminates

$Iter_3$: The parameter that determines how many times that the search will return and perturb s''

NVR-TS algorithm consists of 3 nested loops. The innermost loop is a basic tabu search implementation which moves from one solution to its lowest scoring non-tabu neighbor while updating the tabu list in every iteration. This structure alone fails to provide highly accurate solutions. Therefore, we implemented the outer loop which includes both the basic tabu search and perturbation mechanism. With this implementation, the search jumps to another part of the search space when it can not improve s' any longer. After the implementation of this loop, the accuracy of the resulting solution has increased dramatically. However, jumping through the search space may cause forgetting the lower scoring parts of the search space.

Algorithm A.1 Tabu Search Algorithm (NVR-TS)

Initialization: Obtain an initial solution s , $s' \leftarrow s$, $s'' \leftarrow s$, $s^* \leftarrow s$, $ctr' \leftarrow 0$, $ctr'' \leftarrow 0$, $ctr^* \leftarrow 0$

while $ctr^* < Iter_3$ **do**
 while $ctr'' < Iter_2$ **do**
 while $ctr' < Iter_1$ **do**
 Move from s to s_l
 if $score(s_l) < score(s')$ **then**
 Update tabu list; $s' \leftarrow s_l$
 $ctr' \leftarrow 0$
 else
 $ctr' \leftarrow ctr' + 1$
 end if
 end while
 Perturb(s')
 if $score(s') < score(s'')$ **then**
 $s'' \leftarrow s'$
 $ctr'' \leftarrow 0$
 else
 $ctr'' \leftarrow ctr'' + 1$
 end if
 end while
 if $score(s'') < score(s^*)$ **then**
 $s^* \leftarrow s''$
 $ctr^* \leftarrow 0$
 else
 $ctr^* \leftarrow ctr^* + 1$
 end if
 $s \leftarrow$ Perturb(s'')
end while
Return s^*

Therefore, we implemented the outermost loop, or the third loop, which enables the search to continue from a high quality solution when it can no longer enhance s'' . Thus, the search is able to explore diverse parts of the search space while remembering the lower scoring regions. The outermost loop repeats until s^* (the global best solution) cannot be improved for $Iter_3$ times. NVR-TS algorithm is presented in Algorithm A.1 in simplified form.

Appendix B

Additional Accuracy Results

	Protein	1	2	3	4	5	6	7	8	9	10
With RDC	1UBI	97	97	97	97	97	97	97	97	97	97
	1UBQ	97	97	97	97	97	97	97	97	97	97
	1G6J	6	97	27	97	97	97	97	97	97	97
	1UD7	97	97	97	97	97	97	97	97	97	97
	1AAR	97	97	56	97	97	97	31	97	97	97
Without RDC	1UBI	87	87	87	87	96	96	96	87	87	87
	1UBQ	87	87	87	87	96	96	96	87	87	96
	1G6J	87	6	87	87	87	87	87	87	94	87
	1UD7	81	81	79	79	81	90	90	81	81	90
	1AAR	89	89	4	79	43	79	79	63	23	7

Table B.1: Percent accuracy results of 10 runs on Ubiquitin proteins

	Protein	1	2	3	4	5	6	7	8	9	10
With RDC	1GB1	100	100	100	100	100	100	100	100	100	100
	2GB1	100	100	100	100	100	100	100	100	100	100
	1PG1	100	100	100	100	100	100	100	100	100	100
Without RDC	1GB1	100	100	100	100	100	100	100	100	100	100
	2GB1	100	100	100	100	100	100	100	100	100	100
	1PG1	96	96	96	96	96	96	96	96	96	96

Table B.2: Percent accuracy results of 10 runs on SPG proteins

	Protein	1	2	3	4	5	6	7	8	9	10
With RDC	193L	100	100	100	98	98	93	98	100	98	100
	1AKI	95	98	98	96	98	83	98	98	98	92
	1AZF	88	88	94	88	88	82	94	91	94	94
	1BGI	88	90	97	97	90	97	97	97	90	88
	1H87	100	100	90	91	98	98	100	87	98	100
	1LSC	100	100	100	100	93	100	100	100	90	93
	1LSE	98	98	90	96	96	98	98	65	98	98
	1LYZ	67 ^a ,81 ^b	68 ^a ,82 ^b	56 ^a ,75 ^b	57 ^a ,82 ^b	60 ^a ,79 ^b	67 ^a ,83 ^b	67 ^a ,83 ^b	71 ^a ,79 ^b	64 ^a ,78 ^b	75 ^a ,83 ^b
	2LYZ	89	86	87	89	87	90	87	92	90	92
	3LYZ	89	89	85	85	87	89	91	87	85	84
	4LYZ	89	86	91	92	83	84	85	88	83	89
	5LYZ	85	89	83	85	90	87	87	91	88	87
6LYZ	96	96	96	96	96	94	97	87	94	94	
Without RDC	193L	65	77	73	78	71	77	78	62	78	78
	1AKI	77	77	75	77	77	78	73	78	73	77
	1AZF	63	68	74	75	75	75	74	75	75	69
	1BGI	59	71	71	63	75	69	74	67	76	61
	1H87	77	77	68	64	66	77	67	65	77	77
	1LSC	76	75	76	71	66	74	75	72	71	74
	1LSE	77	75	79	71	77	63	69	79	78	56
	1LYZ	77	69	71	79	79	75	77	70	78	83
	2LYZ	76	67	75	75	70	79	75	81	75	79
	3LYZ	76	81	76	75	80	79	70	78	79	79
	4LYZ	65	67	73	71	77	67	78	56	72	75
	5LYZ	72	72	64	75	69	75	67	75	75	75
6LYZ	74	75	71	67	78	72	60	75	75	78	

^a: With one set of RDCs, ^b: With two set of RDCs.

Table B.3: Percent accuracy results of 10 runs on Lysozyme Proteins

	Protein	1	2	3	4	5	6	7	8	9	10
With RDC	ff2	93	93	93	93	93	93	93	93	93	93
	hSRI	65	33	89	89	23	40	89	86	42	66
	polη	100	100	100	100	100	100	100	100	100	100
	GB1	100	100	100	100	100	100	100	100	100	100
Without RDC	ff2	76	76	25	85	22	71	24	25	76	85
	hSRI	31	40	15	37	24	25	73	16	51	15
	polη	100	100	100	100	100	100	100	100	100	100
	GB1	96	96	91	96	96	96	96	96	96	96

Table B.4: Percent accuracy results of 10 runs on the rest of the proteins

	Parameter Set				Runs									
	α	β	γ	δ	1	2	3	4	5	6	7	8	9	10
With RDC	4%	0.7	1.2	0.2	59	48	57	67	55	79	71	60	67	53
	6%	0.5	1.5	0.2	61	71	87	63	75	79	81	58	65	73
	6%	0.7	1.2	0.2	71	57	89	90	87	62	61	79	63	67
	8%	0.9	2	0.05	79	84	84	90	91	84	72	75	86	87
Without RDC	4%	0.7	1.2	0.2	54	52	69	38	34	30	1	57	8	54
	6%	0.5	1.5	0.2	64	57	52	51	61	72	59	65	66	57
	6%	0.7	1.2	0.2	56	72	53	76	53	58	53	51	71	53
	8%	0.9	2	0.05	70	72	66	66	57	64	80	67	82	53

Table B.5: Percent accuracy results of 10 runs on MBP with several parameter sets

	Parameter Set				Runs									
	α	β	γ	δ	1	2	3	4	5	6	7	8	9	10
With RDC	4%	0.7	1.2	0.2	4	7	1	11	15	9	2	28	7	0
	6%	0.5	1.5	0.2	9	9	9	8	1	15	13	15	8	4
	6%	0.7	1.2	0.2	22	6	24	24	9	20	17	23	21	13
	8%	0.9	2	0.05	4	41	16	5	1	21	43	20	45	33
Without RDC	4%	0.7	1.2	0.2	2	0	0	1	0	2	6	5	0	0
	6%	0.5	1.5	0.2	1	6	2	4	25	4	5	2	1	4
	6%	0.7	1.2	0.2	0	0	3	18	2	8	1	2	3	1
	8%	0.9	2	0.05	1	3	3	1	5	2	0	0	16	10

Table B.6: Percent accuracy results of 10 runs on EIN with several parameter sets