ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

HEURISTIC ALGORITHMS FOR SOLVING CHEMICAL SHIFT ASSIGNMENT PROBLEM IN PROTEIN STRUCTURE DETERMINATION

Ph.D. THESIS

Emel MADEN YILMAZ

Department of Computer Engineering

Computer Engineering Programme

SEPTEMBER 2021

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

HEURISTIC ALGORITHMS FOR SOLVING CHEMICAL SHIFT ASSIGNMENT PROBLEM IN PROTEIN STRUCTURE DETERMINATION

Ph.D. THESIS

Emel MADEN YILMAZ
(504152507)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Ayşe Şima ETANER UYAR
Thesis Co-Advisor: Prof. Dr. Peter GÜNTERT

SEPTEMBER 2021

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

SEZGİSEL ALGORİTMALAR İLE PROTEİN YAPI BELİRLEMESİNDEKİ KİMYASAL KAYMA ATAMA PROBLEMİNİN ÇÖZÜMÜ

DOKTORA TEZİ

Emel MADEN YILMAZ
(504152507)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Ayşe Şima ETANER UYAR
Eş Danışman: Prof. Dr. Peter GÜNTERT

EYLÜL 2021

Emel MADEN YILMAZ, a Ph.D. student of ITU Graduate School student ID 504152507, successfully defended the thesis entitled "HEURISTIC ALGORITHMS FOR SOLVING CHEMICAL SHIFT ASSIGNMENT PROBLEM IN PROTEIN STRUCTURE DETERMINATION", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Ayşe Şima ETANER UYAR** Istanbul Technical University	.....
Co-advisor :	**Prof. Dr. Peter GÜNTERT** ETH Zürich	.....
Jury Members :	**Prof. Dr. Mustafa Ersel KAMAŞAK** Istanbul Technical University	.....
	Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ Istanbul Technical University	.....
	Assoc. Prof. Dr. Arzucan ÖZGÜR Boğaziçi University	.....
	Assoc. Prof. Dr. Yusuf YASLAN Istanbul Technical University	.....
	Asst. Prof. Dr. Teppei IKEYA Tokyo Metropolitan University	.....

Date of Submission : 03 August 2021
Date of Defense : 06 September 2021

To my husband, Sadun Yılmaz…

FOREWORD

First of all, I would like to express my sincere thanks and gratitude to my advisor Prof. Dr. A. Şima ETANER UYAR for her continuous support during all phases of my Ph.D. journey and even beforehand. She has started inspiring me with her creative attitude and productive thinking even in my bachelor years. She encouraged me to start my Ph.D. and supported my academic work with her valuable comments, ideas, and suggestions during my Ph.D. study. I appreciate our discussions and detailed brainstorming sessions that led the direction of my thesis.

I would also like to thank my co-advisor, Prof. Dr. Peter GÜNTERT, who has a significant role in my academic life starting from my master's studies. He also encouraged me while I was planning to start my Ph.D. journey. Even if we worked remotely, he could always find time for me and even blocked his whole days during my visit to ETH Zürich. I thank him for his time, patience, understanding and kindness during our numerous scientific discussions that have greatly improved this thesis.

I would like to express my appreciation to my steering committee members: Prof. Dr. Mustafa Ersel KAMAŞAK, Assoc. Prof. Dr. Arzucan ÖZGÜR and Assoc. Prof. Dr. Yusuf YASLAN for their valuable contributions and important suggestions during our progress review meetings. It was an honor for me to discuss my academic study in this group with these valuable professors.

Last but not least, I would like to thank my whole family, especially my mother, my father, and my sister, for their never-ending trust and unconditional love. My special thanks are for my husband, Sadun YILMAZ, without whom none of this would be possible. In the end, I would like to thank my sons: Kaan, who has always cheered me up and gave me the motivation for going forward, and Arda, who is on the way to this world.

June 2021

Emel MADEN YILMAZ
(Computer Engineer, M.Sc.)

TABLE OF CONTENTS

Page

FOREWORD ..... **ix**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xv**
SYMBOLS ..... **xvii**
LIST OF TABLES ..... **xix**
LIST OF FIGURES ..... **xxi**
SUMMARY ..... **xxv**
ÖZET ..... **xxix**
1. INTRODUCTION **1**
1.1 Purpose of the Thesis 2
1.2 Contribution of the Thesis 4
1.3 Outline of the Thesis 5
2. BACKGROUND AND RELATED WORK **7**
2.1 Heuristic Algorithms 7
2.1.1 Optimization 7
2.1.1.1 Problem types 7
2.1.1.2 Objectives and constraints 9
2.1.1.3 Solution evaluations 9
2.1.1.4 Local and global optimum 10
2.1.2 Search types 10
2.1.2.1 Systematic and local search 10
2.1.2.2 Perturbative and constructive search 11
2.1.2.3 Complete and incomplete methods 12
2.1.3 Stochastic local search 12
2.1.3.1 Iterative improvement (hill-climbing) 12
2.1.3.2 Simple stochastic local search methods 13
2.1.3.3 Hybrid stochastic local search methods 14
2.1.3.4 Population-based stochastic local search methods 14
2.1.4 Evolutionary algorithms 16
2.1.4.1 Terminology 16
2.1.4.2 Methodology 17
2.1.4.3 Genetic algorithms 19
2.1.5 Multi-objective evolutionary algorithms 20
2.1.5.1 Terminology 21
2.1.5.2 Methodology 22
2.1.5.3 Non-dominated sorting genetic algorithm II (NSGA2) 23
2.1.5.4 Strength pareto evolutionary algorithm 2 (SPEA2) 24
2.1.5.5 Non-dominated sorting genetic algorithm III (NSGA3) 25
2.2 Nuclear Magnetic Resonance (NMR) 25
2.2.1 NMR hardware 27

2.2.2 NMR principles 28
2.2.3 NMR experiment steps 30
2.2.4 Chemical shift 34
2.2.5 Chemical shift assignment problem 35
2.2.6 Protein structure 38
2.3 Literature Review 40
3. NEW METHOD ON CHEMICAL SHIFT ASSIGNMENT 45
3.1 Algorithms 45
3.1.1 NSGA2 algorithm 46
3.1.2 NSGA3 algorithm 47
3.1.3 GA algorithm 47
3.1.4 Hill climbing and hybrid algorithms 47
3.2 Problem Model 48
3.3 Objectives and Constraints 49
3.3.1 Objectives 50
3.3.1.1 Shift normality 50
3.3.1.2 Alignment 50
3.3.1.3 Completeness 50
3.3.1.4 Low degeneracy 50
3.3.2 Constraints 51
3.3.3 Evaluation 51
3.4 Score Function 52
3.5 Problem Representation 54
3.5.1 Representation on multiple spectra 54
3.5.2 Chromosome structure 56
3.5.3 Chemical shift frequency calculation 57
3.6 Input and Output Files 58
3.6.1 Input files 58
3.6.1.1 Sequence file 58
3.6.1.2 CYANA library file 59
3.6.1.3 Expected peaks file 62
3.6.1.4 Measured peaks file 63
3.6.2 Output files 64
3.6.2.1 Chemical shift frequency values file 64
3.6.2.2 Consolidation file 66
3.6.2.3 Graphical representation file 67
3.7 NMR Specific Behaviour 67
3.7.1 Atom order 68
3.7.2 Multiple occurrences of atoms 69
3.7.3 Shift reference values from pseudo atoms 69
3.8 Problem Challenges 71
3.8.1 NMR data challenges 71
3.8.2 Search space challenges 72
3.9 Improvements on the Algorithms 72
3.9.1 Crossover and mutation rates 72
3.9.2 Population initialization 73
3.9.3 Population diversity 74
3.9.4 Population size 75
3.9.5 Repetition parameter 75
3.9.6 Hill-climbing algorithm 75

3.9.7 Hybrid algorithms....75
4. RESULTS AND DISCUSSIONS 77
4.1 Experimental Design....77
4.2 Results on the Fes SH2 Protein....78
4.2.1 Dataset details....79
4.2.2 Empirical results....80
4.2.3 Graphical representation....81
4.2.4 Repetition parameter tuning....82
4.2.5 Iteration count tuning....83
4.2.6 Evaluation....84
4.3 Results on the Sso7d Protein....84
4.3.1 Dataset details....85
4.3.2 Empirical results....86
4.3.3 Graphical representation....87
4.3.4 Evaluation....88
4.4 Further Analysis Results....89
4.4.1 Objective analysis....89
4.4.1.1 Results....90
4.4.1.2 Evaluation....95
4.4.2 Overlapping atoms....95
4.4.2.1 Results....96
4.4.2.2 Evaluation....100
4.4.3 Hierarchical Model....100
4.4.3.1 Results....100
4.4.3.2 Evaluation....103
4.5 Discussion....103
5. CONCLUSION AND RECOMMENDATIONS 107
REFERENCES 111
APPENDICES....117
APPENDIX A....119
APPENDIX B....121
APPENDIX C....123
CURRICULUM VITAE....125

ABBREVIATIONS

ADC : Analog-to-digital Converter
CAMRA : Computer-Aided Magnetic Resonance Assignment
COSY : Correlation Spectroscopy
FID : Free Induction decay
FLYA : Fully Automated Structure Determination Algorithm
GA : Genetic Algorithm
He : Helium
MOEA : Multi-objective Evolutionary Algorithm
NMR : Nuclear Magnetic Resonance
NOE : Nuclear Overhauser Effect
NOESY : Nuclear Overhauser Effect Spectroscopy
NP : Nondeterministic Polynomial
NSGA2 : Non-dominated Sorting Genetic Algorithm II
NSGA3 : Non-dominated Sorting Genetic Algorithm III
PASTA : Protein Assignment by Threshold Accepting
PDB : Protein Data Bank
ppm : Parts Per Million
RF : Radio Frequency
Recvr : Receiver
SLS : Stochastic Local Search
SPEA : Strength Pareto Evolutionary Algorithm
TATAPRO : Tracked Automated Assignments in Proteins
TMS : Tetramethylsilane
Xmtr : Transmitter

SYMBOLS

N'	**:** Random population
g	**:** Number of generations
$\mathbb{P}'$	**:** Initial population
$\mathbb{E}'$	**:** External set of population
PF	**:** Pareto Front
Z^s	**:** Structured reference points
Z^a	**:** Supplied aspiration points
Z^r	**:** Reference set
P_t	**:** Parent population at generation *t*
N	**:** Parent population size
S_t	**:** Population at generation *t*
Q_t	**:** Offspring population at generation *t*
R_t	**:** Union population of P_t and Q_t
F_i	**:** Sorted population according to different non-domination levels
$_{1}\mathbf{H}$	**:** Hydrogen stable isotope
$_{13}\mathbf{C}$	**:** Carbon stable isotope
$_{15}\mathbf{N}$	**:** Nitrogen stable isotope
$_{19}\mathbf{F}$	**:** Fluorine stable isotope
$_{31}\mathbf{P}$	**:** Phosphorus stable isotope
$_{57}\mathbf{Fe}$	**:** Iron stable isotope
B	**:** Magnetic field
α	**:** Lower Energy State
β	**:** Higher Energy State
h	**:** Max Planck constant
v	**:** Frequency of a photon light
E_1	**:** Energy level 1.
δ	**:** Chemical shift of the compound
G	**:** Global score
w	**:** Weight
N_a	**:** The set of expected peaks of atom *a*

N'_a : The set of the assigned expected peaks of atom *a*

b_n : Ambiguity of the assignment, degeneracy

$Q_1(a)$: Shift normality of the assigned atoms

$Q_2(a,n)$: Alignment of the chemical frequency of atom *a*

d_n : Total number of assigned atoms

LIST OF TABLES

Page

Table 4.1 : Experimental dataset details.....**79**
Table 4.2 : Chemical shift assignment results of our algorithms NSGA2, NSGA3 and GA, and the state-of-the-art programs FLYA and PINE on the seven fragments of the Fes SH2 protein. **81**
Table 4.3 : Chemical shift assignment results on the 4th fragment of the Fes SH2 protein.....**83**
Table 4.4 : Chemical shift assignment results on the 2nd and 4th fragments of the Fes SH2 protein with different iteration numbers of GA algorithm. **84**
Table 4.5 : Experimental dataset details.....**85**
Table 4.6 : Chemical shift assignment results of our algorithms NSGA2, NSGA3 and GA, and the state-of-the-art programs FLYA and PINE on the two datasets of the Sso7d protein. **86**
Table 4.7 : Objective values after each evaluation. **90**
Table 4.8 : The objective relations with respect to trade-off.....**95**
Table 4.9 : Atom counts that were assigned correctly. **96**
Table 4.10 : FLYA, GA, and hierarchical model results on the Fes SH2 protein. .**101**
Table 4.11 : PINE, GA, and hierarchical model results on the Fes SH2 protein. ...**102**
Table 4.12 : The percentages of the atoms that are assigned correctly on the seven fragments of the Fes SH2 protein. **104**
Table B.1 : Chemical shift assignment results on the 2nd and 4th fragments of the Fes SH2 protein with different iteration numbers of NSGA2, NSGA3 and GA algorithms.....**121**

LIST OF FIGURES

Page

Figure 2.1 : Local optima and global optima on a search space [19]. **10**
Figure 2.2 : One chromosome of 0-1 Knapsack problem with 10 items. **16**
Figure 2.3 : Basic workflow of an evolutionary algorithm. **18**
Figure 2.4 : Basic workflow of the standard genetic algorithm. **19**
Figure 2.5 : One-point crossover operator representation on binary chromosomes. **20**
Figure 2.6 : Mutation operator representation on binary chromosomes. **21**
Figure 2.7 : Pareto front and dominated solutions on a two-dimensional objective space. **22**
Figure 2.8 : 950 MHz Liquid State NMR Spectrometer in BMRZ, Goethe University, Frankfurt am Main, Germany [32]. **27**
Figure 2.9 : NMR spectrometer hardware [33]. **28**
Figure 2.10 : Two quantum energy states: lower energy state “α” and higher energy state “β” [33]. **29**
Figure 2.11 : The energy difference between the two energy states is observed as a line on the NMR spectrum [34]. **30**
Figure 2.12 : Distance restraints that connect the hydrogen atoms are in greed and the protein atoms are in black. This sample is obtained from $_{13C}$ and $_{15N}$ NOESY spectra [43]..... **33**
Figure 2.13 : NMR experiment steps [35]. **34**
Figure 2.14 : NMR spectrum as a function of frequency is the one above. The chemical shift scale is used in general as an NMR spectrum [34]. **35**
Figure 2.15 : Two-dimensional NMR spectrum measured peaks, where the picked peaks are marked with x [47]. **36**
Figure 2.16 : The chemical shift assignment problem finds mapping from expected peaks to measured peaks [44]. **37**
Figure 2.17 : Amino acid general formula [48]..... **38**
Figure 2.18 : Formation of a dipeptide [48]. **39**
Figure 2.19 : Polypeptide formation formula [48]. **39**
Figure 3.1 : The formulation of the chemical shift assignment problem [15]. **48**
Figure 3.2 : The objectives of the chemical shift assignment problem: a) Shift normality b) Alignment c) Completeness d) Low degeneracy [72]. ... **50**
Figure 3.3 : The assignment in three spectra, HNCA, HNcaCO, and CBCANH. .. **54**
Figure 3.4 : Multiple spectrum atoms, expected peaks and measured peaks before the assignment is done. **55**
Figure 3.5 : The chemical shift assignment problem representation of the expected peaks onto the measured peaks on three different spectra. .. **56**
Figure 3.6 : Solution candidate shows the chromosome structure of one spectrum. **56**
Figure 3.7 : The chromosome of the evolutionary algorithm for three spectra..... **57**
Figure 3.8 : Chemical shift assignment representation on fifteen spectra. **57**
Figure 3.9 : Atoms of an expected peak. The above part shows spectrum-specific atoms, atoms along with their expected peaks are shown at the bottom. **58**

Figure 3.10 : Input file for the sequence of amino acids of the protein, *demo.seq*..**59**
Figure 3.11 : CYANA library file, *cyana.lib*. **61**
Figure 3.12 : Example expected peaks for the HNCA spectrum, HNCA_*exp. peaks.* **62**
Figure 3.13 : Expected peak patterns in the HNCA spectrum. **63**
Figure 3.14 : Expected peaks example for HNCA spectrum. **63**
Figure 3.15 : Input file measured peaks example for HNCA spectrum, HNCA.*peaks.* **64**
Figure 3.16 : Output file chemical shift frequency values, *a01_5.prot.* **65**
Figure 3.17 : The details of the chemical shift assignment, *consolidate.tab.* **66**
Figure 3.18 : Graphical representation of the assignment results, *flya.pdf* **67**
Figure 3.19 : N15NOESY.peaks measured peak file of N15NOESY spectrum. ... **68**
Figure 3.20 : N15NOESY_exp.peaks expected peak file of N15NOESY spectrum **69**
Figure 3.21 : CSTABLE entries are showing the general chemical shift statistics and the deviation in rectangles. **70**
Figure 3.22 : Four steps exist in the process of finding shift reference values in the CYANA library. **71**
Figure 3.23 : Assignment on several spectra. The above part shows the number of the expected peaks and measured peaks for each spectrum. The part below shows the total number of solution candidates for the whole problem. **73**
Figure 4.1 : The fragments from the 114-residue Src homology 2 domain from the human feline Fes SH2 ..... **79**
Figure 4.2 : The graphical representation of assignments done by the GA. **82**
Figure 4.3 : The residues on the datasets from the 65-residue Sso7d protein. **85**
Figure 4.4 : The graphical representation of assignments done by the GA. **88**
Figure 4.5 : The trend of the objectives “shift normality” and “alignment”. **91**
Figure 4.6 : The trend of the objectives “shift normality” and “degeneracy”. **91**
Figure 4.7 : The trend of the objectives “shift normality” and “completeness”. **92**
Figure 4.8 : The trend of the objectives “alignment” and “degeneracy” **92**
Figure 4.9 : The trend of the objectives “alignment” and “completeness”. **93**
Figure 4.10 : The trend of the objectives “degeneracy” and “completeness”. **93**
Figure 4.11 : The trend of the objectives “shift normality”, “alignment”, “degeneracy”, and “completeness” in the first 1.000 iterations. **94**
Figure 4.12 : The trend of the objectives “shift normality”, “alignment”, “degeneracy” and “completeness” in 10,000 iterations. **94**
Figure 4.13 : The percentages of the correct assignments on the 1st fragment of the Fes SH2 protein. **97**
Figure 4.14 : The percentages of the correct assignments for GA on the 1st fragment of the Fes SH2 protein. **98**
Figure 4.15 : The percentages of the correct assignments for NSGA2 on the 1st fragment of the Fes SH2 protein. **98**
Figure 4.16 : The percentages of the correct assignments for NSGA3 on the 1st fragment of the Fes SH2 protein. **99**
Figure 4.17 : The percentages of the NSGA2, NSGA3, and GA algorithms on the 1st fragment of the Fes SH2 protein **99**
Figure A.1: The graphical representation of assignments done by the (a) GA, (b) FLYA, and (c) PINE algorithms on the **Fes** SH2 protein. **119**

Figure C.1 : The graphical representation of assignments done by the (a) GA, (b) FLYA, and (c) PINE algorithms on the Sso7d protein..... **123**

HEURISTIC ALGORITHMS FOR SOLVING CHEMICAL SHIFT ASSIGNMENT PROBLEM IN PROTEIN STRUCTURE DETERMINATION

SUMMARY

Heuristic algorithms have been widely used in several different hard optimization problems not only in computer science but also in several other disciplines, including natural sciences, bioinformatics, electronics, and operational research, where computational methods are needed. Heuristic algorithms search for optimal solutions by maximizing or minimizing the given objectives depending on the need while satisfying the given conditions. Heuristic algorithms find solutions in a huge search space where many different possible solution candidates exist. Due to these conditions of the search space, systematic search techniques are not feasible for such kinds of problems. In this thesis, we applied several different heuristic approaches and their combinations on the chemical shift assignment problem of the Nuclear Magnetic Resonance (NMR) spectroscopy.

NMR spectroscopy is one of the methods to determine the three-dimensional structure of proteins. The three-dimensional structure of proteins provides crucial information to detect the shape, structure and function of biological macromolecules. The protein structure also demonstrates the function of proteins by illustrating the interactions of the macromolecules with other proteins or small ligands. Therefore, the three-dimensional structure of a protein can form a basis for drug design against human diseases. NMR has many advantages compared to other techniques; however, NMR spectroscopy needs very advanced computational techniques for providing the protein structure. The chemical shift assignment of the atoms is one of the most challenging problems in NMR spectroscopy. It needs a considerable amount of time by an experienced spectroscopist if the determination is done manually or by a semi-automated method. Additionally, even if the remaining parts of the structure determination methods work perfectly, it is impossible to create the protein structure if the chemical shift assignments are not done correctly. Due to this complexity, the total number of protein structures obtained from NMR spectroscopy is very few compared to its alternative methods, such as X-ray crystallography.

Due to its importance in NMR experiments, the chemical shift assignment problem has recently become one of the most critical research areas in the computational techniques of NMR spectroscopy. There have been many types of research on this problem; however, they are far from perfect. Some of these techniques can provide only partial solutions by assigning only the backbone atoms or only the sidechain atoms. Some of these methods require a very long computation time. Additionally, the results of many of the existing methods have a great area for improvement. In this thesis, we developed a novel method with the heuristic algorithms that provides a fully automatic assignment of the chemical shift values of NMR experiments. First, we studied the background of the problem along with the existing methods. Secondly, we proposed our methods that solve the problem with evolutionary algorithms. Thirdly,

we performed experiments on several different datasets, compared the success of our methods against the state-of-the-art solutions of the problem, and continuously improved our methods. Finally, we performed further analysis on the results and proposed further work.

First, the background of the chemical shift assignment problem is comprehensively studied from the computer science point of view. The optimization processes in heuristic algorithms, stochastic local search methods, iterative improvement, simple stochastic local search methods, hybrid, and population-based stochastic local search methods are discussed in detail. The ant colony optimization and the evolutionary algorithms are analyzed as the population-based stochastic local search methods. After these evaluations, the evolutionary algorithms appeared to be a suitable candidate for solving this problem since they already work with a population, which is a set of solution candidates. We also analyzed the NMR spectroscopy hardware, principles, and experiment steps in detail because the problem is a real application from NMR spectroscopy in natural sciences. Furthermore, we had a deep dive into the chemical shift assignment problem and into the protein structure and peptide formation areas, which are the basis for the NMR spectroscopy calculations. Afterwards, the existing methods for solving this problem are discussed with their drawbacks.

Secondly, we proposed our methods for solving the problem with heuristic algorithms. Our method comprises several different evolutionary algorithms and their combinations with hill climbing, with each other, and constructive heuristic methods. More conventional approach genetic algorithm, GA, and multi-objective evolutionary algorithms, NSGA2 and NSGA3, are applied to the problem. The multi-objective evolutionary algorithms investigated each objective parameter separately, whereas the genetic algorithm followed a conventional way, where all objectives are combined in one score function. While defining the methods, we first defined the problem model, along with the existing conditions and the score function. We modeled the problem as a combinatorial optimization problem, where expected peaks are mapped onto the measured peaks. The chromosome of the algorithm is an array of the expected peaks and the values inside represent their mapped measured peaks. The objectives of the problem are defined in a score function. The constraints are not separately evaluated because they are already fulfilled by the problem model implicitly. Additional fine-tuning and changes are implemented on the algorithms to apply the NMR-specific behaviors to the problem model. Then, the following improvements are realized on the algorithms: We optimized the probability of applying crossover and mutation in the methods. The population initialization is optimized with a constructive initialization algorithm, which minimizes the search space to find better initial individuals. Furthermore, we optimized the population's diversity to find the optimum solutions by escaping from local optima. We also implemented hybrid algorithms by combining a hill-climbing algorithm with our proposed algorithms.

Thirdly, we performed experiments on several datasets with a set of commonly used spectra. We also compared the results of our methods with the two state-of-the-art algorithms: FLYA and PINE. In almost all of these datasets, our algorithm GA yielded better results than PINE. Our algorithm NSGA2 produced better results than PINE in almost half of the datasets. Our NSGA3 algorithm yielded less than 10% correct assignments because only two objectives out of four objectives of our problem model create trade-off. NSGA3 algorithms are known to be successful on problems with more than three objectives. Additionally, our algorithms had better runtime performance than FLYA in more than half of the datasets. Our algorithms could assign all of the

atoms in all datasets, which creates a huge completeness success of the problem, whereas FLYA and PINE algorithms could not provide a complete assignment. Furthermore, we observed in our results that splitting a large protein into smaller fragments improved our algorithms' results dramatically.

Finally, we performed further analysis on our results. These analyses showed us that our algorithms often assigned different atoms than FLYA and PINE. Primarily the GA algorithm can provide good results on some parts of datasets where the state-of-the-art algorithms cannot make any assignment. In order to leverage this success of our algorithms, we proposed a hierarchical method. This method combines FLYA and our algorithm GA to benefit from the different success factors of each algorithm. The results showed that this approach improved the overall success of the algorithms.

In future work, the three algorithms could be combined to achieve better results. Additionally, one can focus on distinguishing atoms that can be assigned consistently and more reliably than others. The assignment is only tentative so that fewer wrong assignments are done. Furthermore, the objective function of the problem can be remodeled to improve the performance of the algorithms. Additionally, our method can be extended in further work so that large proteins are split into smaller fragments before applying our algorithms, which will improve the overall results.

In this thesis, we successfully implemented a fully automatic algorithm for solving the chemical shift assignment problem of NMR spectroscopy. Our method can automatically assign a significant part of the sidechain and backbone atoms without any parameter changes or manual interactions. We produced results that are comparable to the two very well know state-of-the-art algorithms. Our approaches could provide around a 70% success rate on these datasets and assign many atoms that other methods could not assign. Our algorithm outperformed at least one of these two state-of-the-art methods almost in all of our experiments. Additionally, the whole methods are implemented on the MOEA framework, enabling the further implementation of new algorithms easily.

SEZGİSEL ALGORİTMALAR İLE PROTEİN YAPI BELİRLEMESİNDEKİ KİMYASAL KAYMA ATAMA PROBLEMİNİN ÇÖZÜMÜ

ÖZET

Sezgisel algoritmalar zor eniyileme problemlerinin çözümü için sadece bilgisayar bilimlerinde değil; doğal bilimler, biyoenformatik, elektronik ve yöneylem araştırması gibi hesaplamalı yöntemlere ihtiyaç duyulan diğer bilim alanlarında da sıklıkla kullanılır. Sezgisel algoritmalar, verilen amaç fonksiyonlarını ihtiyaca göre enazaltarak ya da ençoklayarak ve verilen kısıtları karşılayarak eniyi çözümü ararlar. Sezgisel algoritmalar çok fazla farklı olası çözüm adayı bulunan çok büyük arama uzaylarında çözüm bulmak için kullanılırlar. Arama uzayındaki bu şartlar sebebiyle, bu tür problemlerde sistematik arama tekniklerini uygulamak olanaklı değildir. Biz bu tezde, birçok farklı sezgisel algoritma ve onların birbirleriyle birleşimini Nükleer Manyetik Rezonans (NMR) spektroskopisinde yer alan kimyasal kayma atama probleminin çözümünde uyguladık.

NMR spektroskopisi proteinlerin üç boyutlu yapılarının tespiti için kullanılan yöntemlerden biridir. Proteinlerin üç boyutlu yapısı, biyolojik büyük moleküllerin şekli, yapısı ve görevlerinin tespiti için çok önemli bilgiler sağlar. Protein yapısı ayrıca büyük moleküllerin diğer proteinlerle ve küçük ligandlarla olan etkileşimlerini göstererek proteinlerin görevlerini anlamamızda yardımcı olur. Bu yüzden, proteinlerin üç boyutlu yapısı insan hastalıklarına karşı ilaç tasarımında temel oluşturur. NMR diğer yöntemlerle karşılaştırıldığında birçok avantaja sahiptir. Örneğin, NMR alternatifi bir yöntem olan X-ışınları örütbilimi ile proteinlerin üç boyutlu yapılarını belirlemek için, proteinlerin kristalleştirilmesi gerekir. Kristalleştirme hem uzun zaman alan hem de proteinlerin doğal ortamı olan sıvı ortamdan çıkmalarına sebep olan bir yöntemdir. NMR spektroskopisinde hem kristalleştirme gibi zaman alan bir işleme ihtiyaç olmaz hem de proteinlerin doğal ortamları olan sıvı ortamda üç boyutlu yapılarının tespiti sağlanır. Fakat NMR spektroskopisi protein yapısının oluşturulması için çok ileri hesaplama tekniklerine ihtiyaç duyar. NMR spektroskopisi protein yapılarını doğrudan üretmez, onun yerine NMR spektrumu olarak bilinen sinyaller üretir. İleri hesaplama teknikleri kullanılarak, bu sinyaller üzerinden hesaplamalar yapılması ve üç boyutlu yapının bilgisayar yöntemleri ile oluşturulması gerekir. Yapılan hesaplama adımları içinde, en zor adım atomların kimyasal kayma frekanslarının atanması problemidir. Bu işlem, deneyimli bir spektroskopist tarafından elle ya da yarı otomatik bir metotla yapıldığında ciddi miktarda zamana ihtiyaç duyulur. Zor bir problem olmasının yanı sıra, kimyasal kayma problemi proteinlerin üç boyutlu yapısının doğru bulunmasında çok önemli bir yere sahiptir. Protein yapı belirleme metotları mükemmel olsa da, kimyasal kayma atamaları doğru yapılmadığında, protein yapısını belirlemek imkânsızdır. Bu yöntemin bu kadar zor olması sebebiyle, NMR spektroskopisinden elde edilen protein yapı sayısı, X-ışınları örütbilimi gibi diğer alternatif metotlarından üretilenlere göre çok düşüktür.

NMR deneylerindeki öneminden dolayı, kimyasal kayma atama problemi son zamanlarda NMR spektroskopisi hesaplamalı teknikleri konusundaki en önemli araştırma alanlarından biri olmuştur. Bu problem üzerinde yapılmış birçok araştırma vardır, ancak bu yöntemler mükemmelden uzaktır. Bu yöntemlerden bazıları sadece yan zincir atomlarını ya da sadece omurga atomlarını atayarak sadece parçalı çözüm sunabilmişlerdir. Bu yöntemlerden bazıları da çok uzun hesaplama zamanlarına ihtiyaç duyarlar. Ek olarak, var olan yöntemlerin çoğunda sonuçların iyileştirilmesi gerekir. Bu tezde, sezgisel algoritmaları kullanarak NMR deneylerinde tam otomatik kimyasal kayma ataması yapan yeni bir metot geliştirdik. Tezin ilk aşamasında, problemin detaylarını ve bu konuda yapılan geçmiş araştırmaları inceledik. İkinci aşamada, evrimsel algoritmalar ile bu problemi çözecek metotlarımızı tasarladık. Üçüncü aşamada, farklı veri kümeleri üzerinde deneyler gerçekleştirerek, metotlarımızın sonuçlarını bu alandaki en gelişmiş algoritmalarla karşılaştırdık ve kendi metotlarımız üzerinde iyileştirmeler yaptık. Son olarak, elde ettiğimiz sonuçlar üzerinde detaylı analiz yaparak, ileride yapılacak çalışmalar için çeşitli öneriler oluşturduk.

İlk aşamada, kimyasal kayma atama probleminin detaylarını ve bu konuda yapılan geçmiş araştırmaları bilgisayar bilimleri bakış açısı ile inceledik. Sezgisel algoritmalardaki eniyileme süreçleri, olasılıksal yerel arama yöntemleri, tekrarlayan iyileşme, basit olasılıksal yerel arama yöntemleri, melez ve toplum temelli olasılıksal yerel arama yöntemleri detaylı olarak incelendi. Toplum temelli olasılıksal yerel arama yöntemleri olan karınca kolonisi optimizasyonu ve evrimsel algoritmaları detaylarıyla değerlendirdik. Bu incelemeler sonrasında, evrimsel algoritmaların zaten çözüm adaylarını içeren bir toplumla çalıştıkları için, bu problemin çözümü için uygun bir aday olduğu kararına vardık. Algoritmalarımız ile çözeceğimiz problem doğal bilimlerden NMR spektroskopisinin gerçek bir uygulaması olduğu için, NMR spektroskopisinin donanımını, prensiplerini ve deney adımlarını da detaylı olarak inceledik. Ek olarak, kimyasal kayma problemi ile NMR spektroskopi hesaplarının temeli olan protein yapısı ve peptit oluşumunu da detaylı olarak analiz ettik. Bu analizler sayesinde problemin modellenmesi ve kullanılacak metotların iyileştirilmesi için gereken ön hazırlıkları tamamladık. Daha sonra, bu problemin çözümü için var olan metotları inceledik. Bu metotların her biri için, içerdikleri güçlü yanları ve hesaplamalarda karşılaştıkları zorlukları ele alarak, detaylı analiz yaptık.

İkinci aşamada, bu problemin çözümü için sezgisel algoritmalar kullanarak kendi metotlarımızı oluşturduk. Bizim metodumuz birçok farklı evrimsel algoritma ile onların hem kendileriyle hem de tepe tırmanma algoritması ile yapıcı sezgisel yöntemlerle birleşmesinden oluşmaktadır. Yöntemleri belirlerken, öncelikli olarak problem modelini ve modeldeki kısıtları ve skor fonksiyonunu belirledik. Kısıtları ve amaçları içeren problem modelini her algoritma için aynı olarak belirledik ve yönteme göre problem modelimizde değişiklik uygulamadık. Biz problemi beklenen zirvelerin ölçülen zirvelere eşleştirildiği bir birleşimsel eniyileme problemi olarak modelledik. Bu problem modelini çözmek için, daha geleneksel bir yöntem olan genetik algoritma, GA, ile NSGA2, NSGA3 gibi çok amaçlı evrimsel algoritmaları probleme uyguladık. Genetik algoritma geleneksel bir yöntem izleyerek, tüm amaç parametrelerini tek bir skor fonksiyonunda birleştirerek çözüm bulmaya çalışır. Diğer taraftan, çok amaçlı evrimsel algoritmalar olan NSGA2 ve NSGA3 algoritmaları her amaç parametresini diğerlerinden bağımsız olarak ele alır. Problemin amaçlarını skor fonksiyonunda tanımladık. Problem modelinde kısıtları ayrıca hesaplamadık çünkü problemi

tanımlama yöntemimizle tüm kısıtlar dolaylı olarak kapsanmış ve gerçekleştirilmiş oldu.

Problemi çözmek için kullanacağımız evrimsel algoritmalarının kalıtım ipliğini, beklenen zirvelerin bir dizisi olarak belirledik. Bu dizinin içindeki değerler ilgili beklenen zirvenin atandığı ölçülen zirveyi gösterecek şekilde kalıtım ipliğini modelledik. Problem modelinde NMR'a özel davranışların uygulanması için, algoritmalarda çeşitli değişiklikler yaptık. Öncelikli olarak, problemin birden fazla spektrumda çalışabilmesi için, kalıtım ipliğinin her spektrum için genişletilmesini tasarlayarak modelimizi güncelledik. Atom sıralarının her spektrum için beklenen ve ölçülen zirvelerde farklı olduğunu ve bir zirvede aynı atomun birden fazla kez bulunabildiğini testlerimizde görerek, modelimizi güncelledik. NMR hesaplamaları sırasında birden fazla atomu temsil eden sahte atom olarak adlandırılan atomların referans değerlerinin hesaplanması algoritmasını da yaptığımız deneyler yardımıyla oluşturarak, yöntemimizi bu hesaplamayı kapsayacak şekilde geliştirdik.

Modelimizi ve algoritmalarımızı tamamladıktan sonra, farklı veri kümeleri üzerinden testler yaparak, algoritmalarımızda çeşitli iyileştirmeler yaptık. Öncelikle, geçiş ve mutasyon olasılıklarını hem sabit, hem de arama uzayının büyüklüğü ile orantılı olarak değişen değerler vererek; bizim problemimiz için en uygun olan değerlerini belirledik. Problemin en büyük zorluklarından birisinin arama uzayının çok büyük olması olduğunu zaten önceden de belirlemiştik. Algoritmalarımızı başlatırken toplumun tüm bireylerini rastgele oluşturmak yerine daha iyi bireylerin oluşturulmasını sağlayan bir algoritma geliştirdik. Bu algoritma ile arama uzayını küçülterek, yapıcı ve sezgisel bir şekilde toplumun bireylerini oluşturduk. Ayrıca, yerel eniyiden kaçarak global eniyi çözümleri bulmak için toplumdaki çeşitlilik değerini belirledik. Toplumu oluşturan birey sayısını farklı veri kümeleri üzerinde denemeler yaparak, uygun bir değeri varsayılan değer olarak belirledik. Aynı şekilde, algoritmalarımızın içinde yer alan tekrarlama parametresi değerini de eniyileştirerek, varsayılan bir değer belirledik. Ek olarak, bizim evrimsel algoritmalarımızın her tekrarlamada buldukları bireyleri, arama uzayında bulundukları alanın en iyisi olarak iyileştirmek için, tepe tırmanma algoritması geliştirdik. Evrimsel algoritmalarımız ile tepe tırmanma algoritmasını birleştirerek oluşturduğumuz melez algoritmaları da problem üzerinde uyguladık.

Tezin üçüncü aşamasında, deneylerimizi, yaygın bir şekilde kullanılan spektrum kümelerini içeren çeşitli veri kümeleri üzerinde gerçekleştirdik. Ayrıca sonuçlarımızı FLYA ve PINE gibi bu konudaki en gelişmiş algoritmaların sonuçları ile karşılaştırdık. Bizim GA algoritmamız, bu veri kümelerinin neredeyse tamamında PINE'dan daha iyi sonuçlar üretti. Bizim NSGA2 algoritmamız da bu veri kümelerinin neredeyse yarısında PINE'dan daha iyi sonuçlar üretti. Bizim NSGA3 algoritmamız bu veri kümelerinde %10'dan fazla doğru atom ataması yapamadı. Bunun sebebini incelemek için amaç fonksiyonlarının değerlerini her tekrar sonrasında inceledik ve problem modelimizdeki dört amaçtan sadece iki tanesinin birbiriyle çelişki oluşturduğunu belirledik. NSGA3 algoritmaları üçten fazla birbiriyle çelişen amacın bulunduğu problemlerde başarı elde ederler. Ek olarak, bizim algoritmalarımız veri kümelerinin yarıdan fazlasında FLYA'dan daha hızlı çalışma performansı gösterdiler. Bizim algoritmalarımız, veri kümesindeki tüm atomların atamasını yaparak çok yüksek *tamamlık* başarı oranına ulaştılar. Öte yandan, FLYA ve PINE algoritmaları tam bir atama yapamadılar. Ek olarak, yaptığımız deneylerde büyük proteinlerin küçük parçalara bölünmesi ile bizim metodumuzun yaptığı doğru atama sayısının %36 oranında arttığını ve yapılan yanlış atamanın da %36 oranında azaldığını gözledik.

Tezin son aşamasında, sonuçlarımız üzerinden problem modeli ve atanan atomları içeren analizler yaptık. Yaptığımız analizlerde bizim algoritmalarımızın FLYA ve PINE'ın hiç atama yapamadığı birçok atomu başarıyla atayabildiğini gözledik. Özellikle GA algoritması veri kümesinde, en gelişmiş algoritmaların hiç atama yapamadıkları yerlerde iyi sonuçlar üretti. Bizim algoritmalarımızın bu başarısından faydalanmak için sıra düzenli yeni bir yöntem tasarladık. Bu yöntem ile FLYA ve bizim GA algoritmaları birbirinin başarılarından faydalanarak sonuca ulaştılar. Sıra düzenli yaklaşım ile elde ettiğimiz sonuçlar, tüm veri kümeleri üzerinde en gelişmiş algoritmaların sonuçlarını geçerek, genel başarı oranlarını yükseltmiştir.

İlerleyen çalışmalarda, bizim bu tezde önerdiğimiz üç algoritma birleştirilerek daha iyi çözümler üretilebilir. Ayrıca, atomların tutarlı olarak ataması ve daha güvenilir atamasına odaklanarak, yanlış yapılan atama sayılarının düşürülmesi üzerinde çalışma yapılabilir. Ek olarak, problemin amaç fonksiyonu yeniden modellenerek performans iyileştirilebilir. Ayrıca, ilerleyen çalışmalarda bizim metodumuz, büyük proteinlerin küçük parçalara bölünmesi ve sonra algoritmaların uygulanması şeklinde genişletilirse, daha iyi sonuçlar elde edilebilir.

Bu tezde, kimyasal kayma problemini çözen ve tamamen otomatik çalışan bir çözüm geliştirdik. Bizim yöntemimiz, herhangi bir parametre değişikliği ya da elle etkileşim olmadan hem yan zincir atomlarının hem de omurga atomlarının önemli bir kısmını başarıyla atayabilmektedir. Bizim sonuçlarımız bu konudaki en gelişmiş iki algoritmaların sonuçları ile karşılaştırılabilir seviyededir. Bizim yöntemimiz ile bu veri kümelerinde ortalama %70 oranında başarı elde edildi ve bizim yöntemimiz diğer yöntemlerin atama yapamadığı birçok atomun başarıyla atamasını yapabilmiştir. Bizim yöntemimiz, yaptığımız deneylerin hemen hemen hepsinde, bu alandaki en iyi algoritmaların en az birinden daha iyi sonuçlar elde etmiştir. Bizim yöntemimiz, çok amaçlı evrimsel algoritmalar kullanılarak modellendiği için, yeni bir amaç ya da kısıt eklenmesi ya da çıkarılması çok pratik bir şekilde problem modeli güncellenerek geliştirilebilir. Son olarak, bizim yöntemimiz MOEA çerçevesi üzerinde geliştirildiği için, bu çerçeve üzerinde geliştirilen yeni algoritmaların bu problemde uygulanması çok kolay bir şekilde yapılabilecektir.

1. INTRODUCTION

Heuristic algorithms have been widely used not only in computer science but also in some other disciplines, including natural sciences, bioinformatics, electronics, and operational research, where computational methods are needed. Heuristic algorithms search for optimal solutions by grouping, ordering or assigning from a finite number of items to maximize or minimize given objectives by satisfying given conditions. Heuristic algorithms have many application areas in computer science and other science fields such as planning, scheduling, finding the shortest or optimum path, internet data packet routing, and map coloring. Heuristic algorithms find solutions in a huge search space where many different possible solution candidates exist. Due to these conditions of the search space, systematic search techniques are not feasible for such kinds of problems. We applied several different heuristic approaches and their combinations on the chemical shift assignment problem of the natural sciences in this thesis. The chemical shift assignment problem is an optimization problem, where a mapping from expected peaks set to measured peaks set of a Nuclear Magnetic Resonance (NMR) spectroscopy is searched. When both of these lists are represented with graphs, this problem corresponds to the graph homomorphism problem, which is a nondeterministic polynomial (NP) complete problem [1].

NMR is one of the methods to determine three-dimensional protein structure. In natural sciences, three-dimensional structure analysis of macromolecules is a very crucial method to detect the shape, classification, and domain structure of macromolecules. The chemical shift assignment of $_{1}H$, $_{13}C$, and $_{15}N$ nuclei resonance frequencies is a significant step of the NMR experiments and it plays an essential role in the success of NMR experiments [2]. The computer programs that calculate the protein structure from NMR data require almost entirely assigned chemical shifts of the atoms to produce a qualified protein structure. Even if there are some software programs that automate the chemical shift assignment process, a considerable amount of time by an experienced spectroscopist is needed if the determination is done manually or by a semi-automated method.

In this thesis, we present a novel algorithm that provides a solution for the fully automatic assignment of the chemical shift frequency values of NMR experiments to their respective atoms. We analyzed the chemical shift assignment problem from the computer science point of view. We investigated the objectives of the problem along with its nature to create a suitable problem model. We applied several different types of heuristic algorithms, including the multi-objective optimization methods, to the problem. Our solution achieves a complete assignment for the whole protein in the NMR experiment, where all of the sidechain and backbone atoms are taken into consideration for a complete resonance frequency assignment. Furthermore, our method assigns chemical shift values to the hydrogen, carbon, and nitrogen nuclei without any manual interactions and any additional input data required from the user.

1.1 Purpose of the Thesis

The field of three-dimensional structure analysis of macromolecules is a rapidly growing interdisciplinary research area in natural sciences. Three-dimensional structures of proteins are essential pieces of information to determine the molecular functions occurring in living organisms. The three-dimensional structure plays a vital role in detecting the shape, classification, and domain structure of macromolecules. It also depicts the interactions of proteins with other proteins or small ligands such as inhibitors, nucleotides, and cofactors. Additionally, it is a crucial source of knowledge for the enzyme mechanism. The protein structure can demonstrate the function of proteins and form a basis for drug development against human diseases.

Recent developments resulted in two complementary methods for protein structure determination: X-ray crystallography and Nuclear Magnetic Resonance (NMR). These two methods are competing against each other in several aspects. X-ray crystallography is a kind of high-resolution microscopy of the crystallized macromolecule. However, NMR spectroscopy has the advantage that the macromolecules do not need to be crystallized; they occur in a solution instead, which provides a similar environment to the flexible structure of macromolecules as in their natural circumstances. On the contrary, the crystal creation for the X-ray crystallography may take an extended amount of time, such as several months [3]. As a result, NMR spectroscopy arises as a powerful technique for demonstrating the

molecular dynamics of the macromolecule for intramolecular dynamics and reaction kinetics [4].

On the other hand, NMR spectroscopy is challenging due to the need for advanced computational techniques to produce the three-dimensional structure of macromolecules. One of the most challenging problems in NMR spectroscopy is the chemical shift assignment process. Due to this complexity, the overall number of results obtained from the X-ray crystallography is much more than the ones derived from the NMR spectroscopy. As per the Protein Data Bank repository [5], there exist around 125,000 molecule structures derived from X-ray crystallography, whereas this number is around 12,500 molecules for the NMR spectroscopy.

In order to deal with this challenge, the chemical shift assignment problem has been recently turned into one of the most critical research areas in NMR spectroscopy. There have been many pieces of research on this problem; however, they are far from perfect. Some of these researches focused only on a partial solution of the problem, the backbone assignment, and exempted chemical shift assignment of the atoms on the sidechain of the molecules [6-9]. Both of the sidechain and backbone atoms' chemical shift assignments are required for a complete three-dimensional structure of a protein though. Even if an assignment of both sidechain and backbone atoms is carried out, some of the existing methods [10-14] have a long computation time, and they do not apply to the large size of proteins. They can only produce results for small or medium-sized macromolecules. Additionally, the results of the existing methods [15,16] have a great area for improvement.

Existing approaches demonstrate different aspects by using different methodologies of computer science. Some of these methods used simulated annealing, such as in the research [6]. Exhaustive search algorithms have been employed in some of these researches [10,11] for small and medium-sized proteins. Some of the existing approaches [13,14,17] adopted heuristic algorithms. Some hybrid solutions exist [15,16] where an evolutionary algorithm is incorporated with a local search. As explained above, these solutions have limitations depending on the size of the macromolecule, whereas some of them cannot even produce a complete solution.

This thesis provides a novel method for solving the chemical shift assignment problem of NMR spectroscopy. Firstly, we analyzed the problem from the computer science

point of view and defined the problem model along with the objectives and constraints. Then, we applied several heuristic algorithms, including genetic algorithms and multi-objective optimization approaches, on the chemical shift assignment problem. The eventual purpose of the thesis is to produce a fully automatic solution for solving the chemical shift assignment problem of NMR spectroscopy for both backbone and sidechain atoms of a macromolecule. The scope of this thesis includes the necessary steps for the chemical shift assignment problem. The remaining parts of the NMR spectroscopy are out of the scope of this thesis.

1.2 Contribution of the Thesis

Chemical shift assignment is a very cumbersome and time-consuming process in NMR spectroscopy. This thesis develops a fully automatic algorithm for solving the chemical shift assignment problem by determining the chemical shift resonance frequency for the atoms in a protein without any manual interactions. Replacing this cumbersome manual assignment process for NMR spectra will accelerate the investigations of the structure, dynamics, and interactions of proteins.

The functionality of evolutionary algorithms has been analyzed on many different problems. Additionally, their multi- and many-objective versions have been applied to several problems to figure out their performances on problems having more than one objective. This research will focus on several heuristic algorithms to solve the chemical shift assignment problem in protein three-dimensional structure determination. The relations of the objectives of this problem were not clearly identified before. First, we investigated these objectives and defined the problem model.

Furthermore, we analyzed the constraints of the problem and their relations with each other, and their impacts on the objectives of the problem. Additionally, we studied the problem to find out possible additional objectives and constraints that are not formulated yet. Manual assignments might take care of these objectives or constraints intuitively, but they are not formulated yet in the problem definition. Afterwards, we defined our algorithms for solving this problem including conventional genetic algorithms and multi-objective optimization methods. We furthermore defined several combinations of these algorithms with each other and some other heuristic algorithms. We finally optimized the parameters of algorithms by analyzing their results on several

different experimental datasets. As a result, this thesis provides a comprehensive overview of the existing methodologies of computer science on this optimization problem.

This thesis provides a deep dive analysis on the heuristic algorithms with several aspects. The functionality, efficiency, performance, and behaviors of these methods are analyzed in detail. These analyses are examined on a real-world problem: the chemical shift assignment from protein three-dimensional structure determination area. This thesis also provides a method for the chemical shift assignment problem that can make a fully automatic assignment on the backbone and sidechain atoms without any manual interactions. This method will reduce the time and cost needed for the chemical shift assignment in NMR spectroscopy compared to the manual assignment process. Additionally, instead of a spectrometer specialist, the resonance assignment can be determined by non-specialists in the NMR area. Furthermore, the results obtained by an algorithm instead of subjective human decisions will be reproducible. As a result, it will make NMR structure determination and dynamics studies of proteins more favorite and will contribute to the three-dimensional structure determination of the macromolecules. Eventually, all parties, such as pharmaceutical companies, universities and biomedical research in general, will benefit from this thesis.

In this thesis, for the first time, multi-objective optimization techniques are applied to the NMR resonance assignment problem. Multi-objective techniques allow the user to systematically control and analyze the impact of different types of information on the assignment result. Eventually, it will bring a different point of view on the chemical shift assignment problem, where each objective and each constraint of the problem are analyzed separately from each other in order to find better solutions. This behavior will bring flexibility in adding new objectives to the problem model, which have not been evaluated in the problem model yet. Additionally, the implementation of this thesis is developed on the MOEA framework, which makes further implementation even with the new algorithms on the framework very easy.

1.3 Outline of the Thesis

In this chapter, a general introduction to the research of this thesis is presented shortly. In the second chapter, the background of this research on the computer science and natural science areas is explained. These are followed by the literature review on the

chemical shift assignment problem on the NMR spectroscopy. In the third chapter, the materials and methods of this thesis are defined in detail to provide a comprehensive overview of the methods used in this thesis. In the fourth chapter, the results of these methods on several different experimental datasets have been presented. The last chapter, 5, concludes the thesis and the research results and it also proposes future work areas for the researchers on this topic.

2. BACKGROUND AND RELATED WORK

In this chapter, we provided a comprehensive overview of the background of our thesis. Firstly, the heuristic algorithms are explained starting from the general terms, search techniques, and optimization approaches. Secondly, the Nuclear Magnetic Resonance (NMR) is explained by giving an overview of its hardware, principles, which is followed by the NMR experiment steps. Afterward, the chemical shift process and the chemical shift assignment problem are discussed in detail, followed by the protein structure explanation. Lastly, a broad literature review of the solution proposals for the chemical shift assignment problem is presented.

2.1 Heuristic Algorithms

Heuristic algorithms are used to find solutions to optimization problems that cannot be solved by conventional methods due to the vast search space of the problems. One of the heuristic algorithms and a very widely used one is evolutionary algorithms. In this thesis, we used evolutionary algorithms and their several combinations for solving the chemical shift assignment problem. This subchapter will present an overview of the optimization problems and heuristic algorithms.

2.1.1 Optimization

Optimization is a process that focuses on getting the best possible option as an outcome under predefined conditions and constraints. Optimization already occurs in daily life as a basis for our decisions, such as taking the shortest route, deciding on which car to buy or where to eat.

2.1.1.1 Problem types

There are two types of problems with respect to optimization: Decision problems and optimization problems. The decision problems try to find out a decision, whereas the optimization problems focus on maximizing or minimizing given objective(s).

Additionally, there are combinatorial problems that can be defined as either decision or optimization problems.

Decision problems

The basis of these problems is finding out whether there are one or more solutions that can satisfy all given constraints of the problem. For example, the graph coloring decision problem tries to find a graph coloring combination with less than the predefined number of colors under the condition that none of the adjacent vertices have the same color.

Optimization problems

These problems are generalized decision problems, where an optimum solution for an objective is searched. For example: Coloring a graph with a minimum number of colors, where none of the two adjacent vertices has the same color, is an optimization problem. These problems have objective functions to evaluate the success of their result. There are two types of optimization problems: Minimization problems, where the goal is to minimize the given objective function, and maximization problems that work on maximizing it. Any minimization problem can be formulated as a maximization problem and vice versa.

Combinatorial Problems

Combinatorial problems [18] can be either decision problems or optimization problems. These problems deal with a finite set of discrete elements and try to find out an assignment, grouping, or ordering of these finite numbers of elements. In case of a decision problem, they try to find a decision. In an optimization problem, they try to minimize or maximize the given objectives of the problem while satisfying its constraints. There are many application areas of combinatorial problems in computer science and other disciplines such as planning, scheduling, finding the shortest or optimum path, internet data packet routing, and map coloring.

Combinatorial problems search for solutions in a vast search space. The search space and the possible combinations are so huge that deterministic search methods cannot find solutions for these problems. Therefore, instead of systematic search techniques, heuristic approaches are applied to these problems.

There have been several kinds of research in this area to solve combinatorial optimization problems using heuristic approaches. The chemical shift assignment

problem is a combinatorial optimization problem, where an optimum mapping from expected peaks to measured peaks is searched.

2.1.1.2 Objectives and constraints

Optimization problems are defined with its objective(s) and constraint(s).

Objective

The objective is the goal of an optimization problem. It can be either minimizing or maximizing a given objective. For example, minimizing the total distance is an objective of finding the shortest path problem. The objective of an optimization problem has to be formulated so that it can be evaluated during optimization. The most fitting solution for the predefined objective will be searched according to its value in the problem.

The objective function is used in optimization problems to measure the quality of the solution candidate concerning its objective. In minimization problems, the goal is to find the solution creating the minimum value for the problem's objective function. Similarly, a solution that produces the maximum value for the objective function is searched in maximization problems.

Constraint

The constraint is a criterion that has to be satisfied in a problem. If a constraint of a problem is not fulfilled, then the solution is not acceptable. For example, one of the constraints in the traveling salesman problem is visiting a city exactly once. If one city is visited more than once in a solution, then this solution is not acceptable.

2.1.1.3 Solution evaluations

Several different terms are used in the evaluation of the solutions in optimization problems. In this subchapter, we explained the most important terms that have been used in the optimization problems.

Candidate Solution

Candidate solutions are the combinations of solution components that might produce a solution to the problem.

Solution

A solution is a candidate solution that satisfies all constraints of the problem.

Feasible (valid) solution

A candidate solution is called feasible (or valid) if it satisfies all of the given constraints of the problem.

Infeasible (invalid) solution

A candidate solution is called infeasible (invalid) if it does not satisfy one or more constraints of the problem.

2.1.1.4 Local and global optimum

Optimization problems search for the best possible option on their search space, either the minimum or the maximum point depending on their objective. If a point on the search space is better than all of its neighbors, it is called local optimum. If it is a minimization problem, then the point is called local minimum. If it is a maximization problem, it is called local maximum. If a point is better than all of the other points of the search space, then it is called global maximum. The point is called global maximum in a maximization problem and global minimum in a minimization problem. An example of the global and local optimum points is given in Figure 2.1. [19]

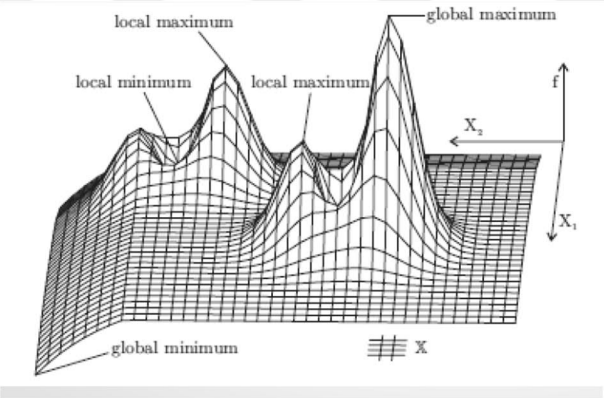

Figure 2.1 : Local optima and global optima on a search space [19].

2.1.2 Search types

The computational approaches that deal with finding an optimum solution for combinatorial problems are called search algorithms. There are several different types of search algorithms depending on some criteria.

2.1.2.1 Systematic and local search

There are two types of search algorithms with respect to its search approach: systematic search and local search.

Systematic search

Systematic search algorithms traverse the whole search space consistently by using deterministic search methods. These techniques are also called classical techniques or deterministic techniques. They check all of the solution candidates in its search space to find out the global optimum solution. Because of this behavior, they require a relatively long computational time. Systematic search methods are preferred for the problems without any time constraints. [19]

Local search

Local search algorithms do not traverse the whole search space. They start on a position on the search space and iteratively moves from one position to another by checking the neighbors at each step. These algorithms terminate once a deemed optimum solution is found or a particular time is elapsed. The local search algorithms are also called modern heuristics or stochastic techniques. One of the biggest challenges in the local search algorithms is that they might be stuck at the local optimum instead of finding the global optimum. Most local search algorithms take advantage of the random neighbor selection to deal with the local optima. These approaches are called stochastic local search algorithms. The local search algorithms are preferred in case relatively good results are needed in a short time. [19]

2.1.2.2 Perturbative and constructive search

In combinatorial problems, search space is traversed to find solutions. There are two search paradigms with respect to the search space types: perturbative and constructive search. [19]

Perturbative search

Perturbative search works on a search space that is composed of complete solution candidates. In other words, each item of the search space is a complete solution candidate. This approach receives the current solution candidate and changes its one or more components in order to create the new solution candidate. A complete solution candidate exists at the algorithm at any time of the search.

Constructive search

On the other hand, constructive search iteratively builds up a solution candidate on search space of partial candidate solutions. A solution component is defined at each step of the algorithm and a complete solution is achieved at the end of the algorithm.

2.1.2.3 Complete and incomplete methods

Complete methods guarantee to find a solution if there exists any. They can also detect that there does not exist any solution to a given problem. The systematic search techniques are complete, which means that they guarantee to find a solution if it exists in the search space. Otherwise, they provide the information that the optimum solution does not exist. However, the incomplete methods cannot ensure to find a solution and they cannot determine whether a solution does exist or not for the given problem. The local search methods are incomplete, so they cannot guarantee to find a solution for the problem on the search space, and additionally, they cannot provide information on whether a solution on the search space exists or does not exist.

2.1.3 Stochastic local search

Stochastic local search (SLS) algorithms are local search algorithms that use randomized approaches while creating or selecting a possible solution candidate in combinatorial optimization problems. The stochastic local search algorithms are one of the best approaches to solve hard combinatorial optimization problems.

The SLS algorithms use the local information, called heuristic values, at each step to decide the next solution candidate that will be assigned as the current solution of the problem. There are several types of stochastic local search algorithms. This subchapter provides an overview of these types of stochastic local search algorithms. [19]

2.1.3.1 Iterative improvement (hill-climbing)

The basic idea of this technique is to provide an iterative improvement in the solution candidate at each step of the algorithm. The method starts the search process on a random point on the search space and improves the results iteratively by checking its neighbors at each step.

Any of the neighbors that improve the solution is selected as the current solution. If there is no improving neighbor, the method stops and returns the current solution as the final solution. This approach is also known as hill climbing. As a result of accepting improving neighbors only, iterative improvement methods cannot escape from local optima. It might get stuck to the local optimum and might not reach the global optimum. [20]

2.1.3.2 Simple stochastic local search methods

As iterative improvement methods cannot escape from local optimum, simple SLS methods have been developed to deal with local optima. While the iterative improvements methods continuously improve the current solution candidate at each algorithm step, the simple stochastic local search methods allow some worsening steps on the current solution to escape from local optima. [19]

Randomized iterative improvement

In order to escape from local optima, randomized iterative improvement allows the selection of random neighbors with a predefined probability. It is different from the iterative improvement algorithm only with this probability approach. As in the iterative improvement method, it starts the search process on a random point in the search space. Afterwards, it decided firstly at each search step whether an iterative improvement step will be done, meaning that an improving solution from the neighbors of the current solution will be selected; or a random walk will be applied, meaning that one of the neighbors of the current solution candidate will be selected randomly. After a predefined number of iterations are done, or after predefined CPU time is elapsed, the algorithm terminates and returns its current solution candidate as the final solution of the algorithm.

Probabilistic iterative improvement

The probabilistic iterative improvement method is a modified version of the iterative improvement, where a worsening neighbor selection is made with a fixed probability. Like the iterative improvement algorithm, this algorithm starts the search process on a random point in the search space. At each step, this algorithm makes an iterative improvement step with some predefined probability and makes a worsening step otherwise. In the worsening steps, the algorithm selects one of the worse neighbors than the current solution candidate's objective function value. The parameter that defines the probability of selecting a worsening solution candidate is called *temperature*. The worsening steps are used to escape from local optima.

Simulated annealing

The simulated annealing algorithm [21] is a specific version of the probabilistic iterative improvement. As in the probabilistic iterative improvement method, simulated annealing starts at a random point in the search space. The difference between is on the temperature value, which defines the probability of selecting a

worsening neighbor. The temperature value is constant during the probabilistic iterative improvement algorithm. However, the temperature parameter is not constant in simulated annealing, and it decreases with time. The decrease ensures that the probability of selecting a worsening neighbor is high at the beginning of the algorithm, and it decreases while coming closer to the end of the algorithm. The worsening steps avoid getting stuck to local optima.

Tabu search

Tabu search [22] is another simple SLS algorithm. Tabu search has a different approach from the other improvement algorithms to escape from local optima: It assigns tabu attributes to recently visited solution candidates or solution components. Tabu search applies an iterative improvement algorithm at each step of the algorithm but it does not select a neighbor solution marked with tabu attributes. The tabu attributes are kept on solution candidates during a predefined number of algorithm steps, which is called *tabu tenure*. The tabu tenure parameter mainly defines the performance of the tabu search: If the tabu tenure is too low, the algorithm has problems with escaping from local optima. If the tabu tenure is too high, then the acceptable neighbor moves are very restricted, so the algorithm cannot achieve acceptable results.

2.1.3.3 Hybrid stochastic local search methods

Simple SLS methods might not find optimum results when they are run alone. In such cases, a combination of simple SLS methods might produce better results. This approach is called the hybrid SLS method. For example, randomized iterative improvement is an example of a hybrid stochastic local search algorithm that is a combination of iterative improvement and uninformed random walk [19]. There are many other hybrid stochastic local search combinations. For example, any stochastic local search algorithm can be combined with uninformed random picking, restarting the algorithm from a random point on the search space to create a hybrid stochastic local search algorithm.

2.1.3.4 Population-based stochastic local search methods

The SLS methods explained above deal with only one candidate solution at each search step of the algorithm at any time. Population-based SLS methods deal with a population of solution candidates during the search steps of the algorithm. The most

widely used population-based SLS methods are ant colony optimization and evolutionary algorithms.

Ant colony optimization

Ant colony optimization [23] is a population-based stochastic local search algorithm that is inspired by the behaviors of real ants. Ants leave a so-called pheromone on their way while moving from one point to another, and the colony gets the advantage of the pheromone on the route to find their way. The more ants select a way, the more pheromone is released on that path. As more ants use the same way, the probability of selecting that way increases due to the increase in the pheromone on that way. The pheromone stays a specific time on that way and it evaporates over time.

The ant colony optimization algorithm uses a very similar idea to natural ant behavior. A predefined number of artificial ants, agents, are created in the colony. The ant colony optimization is a constructive search methodology since, in the beginning, the ants did not have any solution candidates but build up their solution candidates by adding one solution component at each step. As in all stochastic local search methods, ants use local neighborhood information to select their next move. Local information is composed of the pheromone trail and a heuristic value. The ants have an indirect information share via pheromone trails in the ant colony optimization algorithm. Heuristic value is a piece of deductive information that shows the desirability of selecting that path, mainly the cost of adding that solution component. After adding one solution component to their solution, each ant updates the pheromone information on their route. By adding a solution component at each search step, each ant builds one solution candidate at the end of the algorithm.

Evolutionary algorithms

Evolutionary algorithms are another population-based stochastic local search algorithm. In the ant colony optimization algorithm, there is not a direct information exchange among the ants. However, there is a direct information exchange in evolutionary algorithms among the individuals of the population. The population of evolutionary algorithms is composed of a predefined number of individuals. Each individual represents a solution candidate at any time of the algorithm.

The evolutionary algorithm performs a perturbative search on the search space. In other words, existing solution candidates are changed and converted into different solution candidates during the search. At the end of the algorithm, each individual of

the population has a solution candidate, and the algorithm returns the best one as the final result.

There are many different types of evolutionary algorithms, some of which are genetic algorithms [24] and genetic programming [25]. This thesis applied evolutionary algorithms and their several combinations on the chemical shift assignment problem explained in detail in the next subchapters.

2.1.4 Evolutionary algorithms

An evolutionary algorithm is a population-based stochastic local search algorithm. Evolutionary algorithms are inspired by the theory of biological evolution of the species in nature that is formulated by Charles Darwin. The theory is also known as the survival of the fittest. [26]

2.1.4.1 Terminology

There are some terms to be explained to understand the evolutionary algorithms. In this subchapter, we introduced the most important terms to understand our methods for the chemical shift assignment problem.

Chromosome

Chromosome defines how one solution candidate is represented in an evolutionary algorithm. Depending on the problem model, a chromosome can have binary, integer, or string encodings. Figure 2.2 presents an example of a binary encoding chromosome, which represents a chromosome in the 0-1 Knapsack problem.

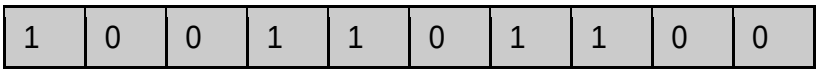

Figure 2.2 : One chromosome of 0-1 Knapsack problem with 10 items.

Each item has a fixed weight and a fixed value in the Knapsack problem. The goal is to select the items so that the maximum weight limit of the knapsack will not be exceeded, and the maximum value will be achieved from these selected items. The chromosome presents the results of the item selection: 1 means the item is selected, and 0 means it is not selected.

Gene

Each value of a chromosome is called *gene*. For example, each of the 0 or 1 values in Figure 2.2 is the gene of the chromosome.

Individual

Each solution candidate of the population is called *individual*. In evolutionary algorithms, one individual has one chromosome.

Fitness

Fitness is the objective function value of an individual. The fitness shows the degree to which the solution candidate fits the objective of the problem and how much the solution candidate is desirable to be the solution of the problem.

Parents

The individuals that are used to create a child individual are called *parents.*

Offspring

The new individuals created after some operators such as crossover and mutation are applied on the parent individuals are called *offspring*.

Population

The individuals of an evolutionary algorithm at any time are the *population* of the algorithm. Each evolutionary algorithm mainly works with a population that is composed of a predefined number of solution candidates.

Population size

The number of individuals in a population is called the *population size*.

Generations

The individuals in a population that are evaluated during iterations are called *generations* of evolutionary algorithms.

2.1.4.2 Methodology

The workflow of a basic evolutionary algorithm is given in Figure 2.3. An evolutionary algorithm starts with population initialization, where a predefined number of individuals is created by the algorithm's initialization method. These individuals are the initial population of the evolutionary algorithm. Until a termination criterium is reached, the algorithm performs the following activities on its current population: selection of parents that will go through the further operators, applying recombination operators such as crossover and mutation in genetic algorithms [24], and replacing some parent individuals in the current population with these new offspring, with the replacement method of the algorithm. After these operators are applied, the current

population is partially or totally replaced by the new individuals that are created during the reproduction phase. The details of each step of the basic flow of an evolutionary algorithm is explained below in detail.

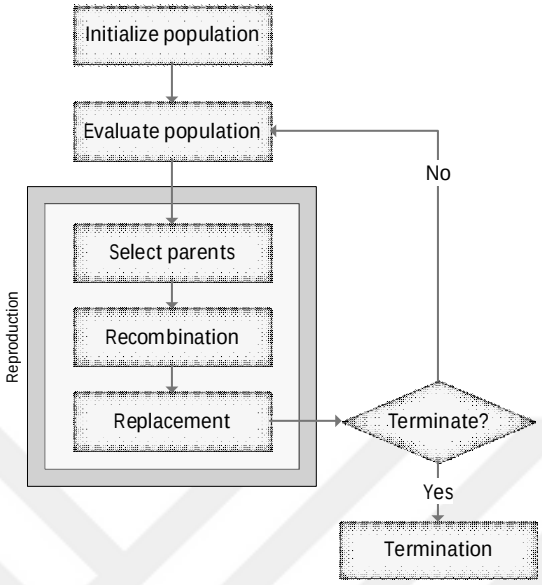

Figure 2.3 : Basic workflow of an evolutionary algorithm.

Evaluation

At the beginning of new generation creation, existing population's individuals are evaluated with respect to their fitness values. Then reproduction of the evolutionary algorithm is employed that will perform selection, recombination, and replacement over the population.

Selection

Firstly, the individuals that will go through the reproduction steps are selected as per the selection strategy of the algorithm. There are several selection methods in evolutionary algorithms, such as roulette-wheel selection and tournament selection. The individuals that have better fitness values will have more probability of being selected. The result of the selection operators will survive in the next generation, and they will be parents for the crossover and mutation operators.

Recombination

The selected parents of the population go through the recombination process in evolutionary algorithms to produce the generation's offspring. In genetic algorithms [27], there are crossover and mutation operators as the most widely used recombination operators.

Termination

The algorithm terminates once the termination criterium is reached, which is mostly performing a predefined number of iterations. There are also some other termination criteria such as reaching a fixed CPU time or having a predefined number of iterations in a row where no improvements in the solution quality are achieved. If elitism is applied, then the overall best solution candidate from any iteration is returned as the final solution. If no elitism is used, then the best solution candidate of the current generation is returned as the final solution of the algorithm.

2.1.4.3 Genetic algorithms

Genetic algorithm is one of the evolutionary algorithm approaches that is a population-based stochastic local search method. The workflow of a standard genetic algorithm [28] is provided in Figure 2.4. The method works the same as evolutionary algorithms. As reproduction operators, crossover and mutation are used in genetic algorithms.

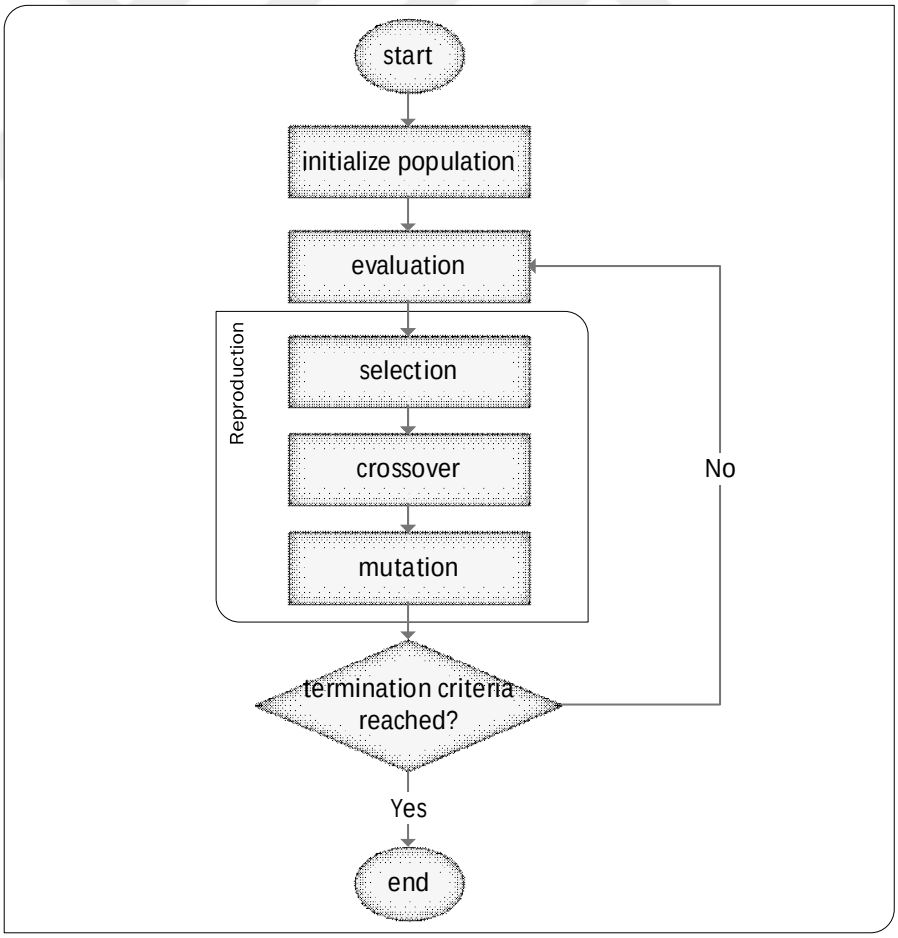

Figure 2.4 : Basic workflow of the standard genetic algorithm.

Crossover

The *crossover* operator is also known as the *recombination* operator. The selected two or more individuals will go through the crossover and exchange some of their genes. Here, the share of knowledge among the parent individuals is accomplished with the goal that the offspring will have a more fitting solution candidate than its parents with respect to the algorithm's objective. There are different types of crossover operators, such as one-point crossover, two-point crossover, and uniform crossover operators. An example representation of a one-point crossover operator is given in Figure 2.5. There are two parents selected for the crossover operation. One-point crossover cuts these parents and exchanges their genes after this crossover line. As a result, two offspring with exchanged genes are created. [29]

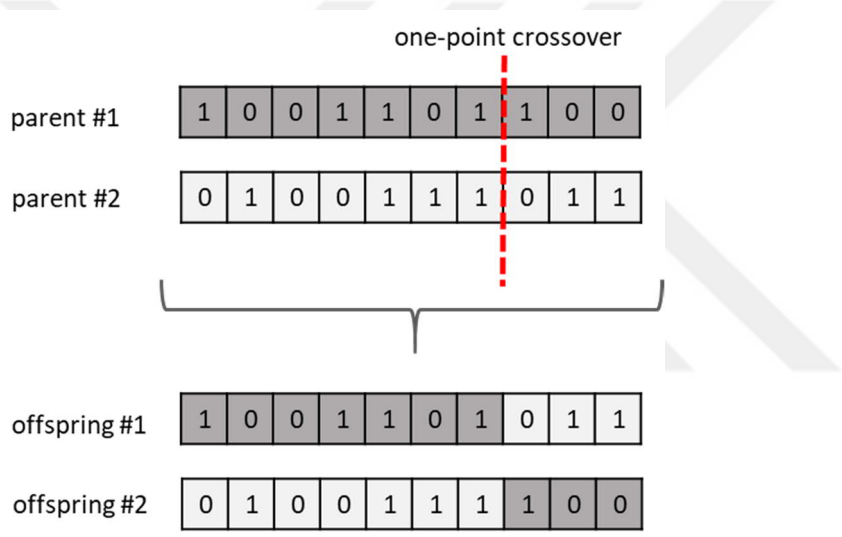

Figure 2.5 : One-point crossover operator representation on binary chromosomes.

Mutation

At the end of the reproduction phase, the mutation is applied to the individuals. Mutation performs changes on the randomly selected gene values. An example of the mutation operator on a binary chromosome is provided in Figure 2.6. The mutation changes one selected gene of the individual. Since this example is for binary chromosomes, the value of the gene is changed from 1 to 0. [29]

2.1.5 Multi-objective evolutionary algorithms

The methods mentioned so far deal with single objective algorithms, where either minimization or maximization of this given objective is targeted. However, there exists more than one objective in real-world problems, which are even conflicting with each

other. For example, we may need to find the shortest path in the traveling salesman problem, whereas the cheapest path is another interest. Some objectives might be aligned with each other, meaning improving one objective will make the other objective better as well. However, most of these objectives create a trade-off in real life. In other words, improving one objective will diminish other objectives. The evolutionary algorithms designed to deal with multi-objective optimization problems are called multi-objective evolutionary algorithms [29].

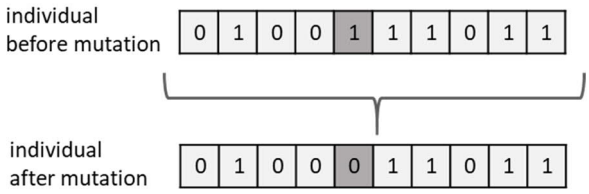

Figure 2.6 : Mutation operator representation on binary chromosomes.

2.1.5.1 Terminology

The terminology used in these methods are explained below: [30]

Pareto optimality

As explained above, multi-objective optimization problems return a set of solution candidates. This set is composed of Pareto optimum solution candidates. Pareto optimality means that there is no other solution in search space that has better results in one of the objectives while not worsening any of its other objectives.

Pareto optimal set

The Pareto optimal set is a set of solution candidates that have Pareto optimality.

Non-dominated solutions

Non-dominated solutions are the solutions that exist in the Pareto optimal set.

Pareto front

The set of the Pareto optimum vectors plotted in objective space are called the Pareto front. Figure 2.7 shows an example Pareto front on the objective space of a problem with two objectives. The curve on this graph is the Pareto front of this search space. All solution candidates on the Pareto front dominates the solution candidates under the curve as shown in the figure. It means that the solution candidate *a* is better at least on one objective than the solution candidates on the dotted area and there does not exist any solution candidate on this area that is better than *a* in any objective.

Multi-objective optimization problems

Optimization problems with two or three objectives are called multi-objective optimization problems.

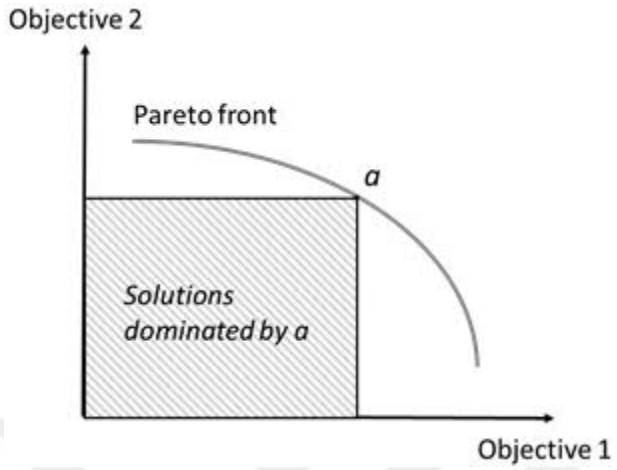

Figure 2.7 : Pareto front and dominated solutions on a two-dimensional objective space.

Many-objective optimization problems

Optimization problems with four or more objectives are called many-objective problems. Due to the fact that many-objective optimization problems deal with relatively complex search space, they are hard to handle compared to multi-objective optimization problems.

2.1.5.2 Methodology

Multi-objective optimization methods do not produce a single solution as in single optimization problems but a set of solutions representing the compromises, or trade-offs, in objective space. The multi-objective optimization problems are composed of two major parts: 1) finding a set of solution candidates 2) choosing the best solution for a particular application. In the first part of the solution, a set of good solution candidates are found out, which are vectors of solution candidates representing their trade-offs in objective space. In the second part of the problem, a decision-maker selects the best solution among these possible solutions.

Evolutionary algorithms work with a set of solution candidates, called population. As explained above, multi-objective optimization methods should return a set of solution candidates that should be evaluated afterward. Therefore, the evolutionary algorithms occur to be a suitable match for the multi-objective optimization problems due to their

population-related approach. Within a single run of an evolutionary algorithm, many Pareto optimal solution sets can be identified. However, any other traditional mathematical method would need several runs to produce a set of Pareto optimum solution candidates. Additionally, evolutionary algorithms work irrespective of the shape and continuity of the search space, whereas traditional mathematical methods might have difficulties with both. [30]

There are two generations of multi-objective evolutionary optimization algorithms: The first generation is composed of relatively simple approaches such as Aggregating functions, VEGA, MOGA, NSGA, NPGA, and NPGAII [30]. These approaches use fitness sharing (or niching) and Pareto ranking. Fitness sharing ensures that the size of neighbors is controlled with a constant value, called niche radius. The second generation comprises more complex methodologies, where a second generation is introduced to enable elitism. SPEA, SPEAII, NSGA-II, PAES, PESA, PESA II, and microGA are the second generation MOEA. The most favorable ones of these methodologies are defined in detail in the next subchapters [30].

2.1.5.3 Non-dominated sorting genetic algorithm II (NSGA2)

NSGA2 is an improved version of the Non-dominated Sorting Genetic Algorithm (NSGA). NSGA2 is computationally more efficient than its predecessor. NSGA2 uses elitism and a crowded comparison operator to keep diversity. The pseudo-code of the algorithm is available in Algorithm 1. All non-dominated individuals are classified into a layer, and these individuals are then separated from the population. Afterwards, another non-dominated group is created from the remaining individuals of the population and they are also separated from the population. This process continues until all individuals belong to a specific non-dominated layer.

A dummy fitness value is assigned to each layer, which is called a non-dominated rank. This value gets greater in lower layers, ensuring that the individuals from the first layer are more likely to reproduce. Furthermore, niching is applied by calculating crowding distance for each individual. Crowding distance represents the density of solutions surrounding a particular solution. Its insertion into the algorithms ensures diversity. Selection is based on both random rank and crowding distance called crowded comparison operator.

Algorithm 1 NSGA-II algorithm [30].

1: **procedure** NSGA-II $(N', g, f_k(X_k))$ $\forall N'$ members evolved g generations to solve $f_k(X)$
2: Initialize Population $\mathbb{P}'$
3: Generate random population – size $\mathcal{N}'$
4: Evaluate Objective Values
5: Assign Rank (level) Based on Pareto dominance – sort
6: Generate Child Population
7: Binary Tournament Selection
8: Recombination and Mutation
9: **for** $i = 1$ to g **do**
10: **for** each Parent and Child in Population **do**
11: Assign Rank (level) based on Pareto – sort
12: Generate sets of nondominated vectors along PF_{known}
13: Loop (inside) by adding solutions to next generation starting from the *first* front until $\mathcal{N}'$ individuals found determine crowding distance between points on each front
14: **end for**
15: Select points (elitist) on the lower front (with lower rank) and are outside a crowding distance
16: Create next generation
17: Binary Tournament Selection
18: Recombination and Mutation
19: **end for**
20: **end procedure**

In case two solution candidates are from different fronts, the one with lower rank is selected. If they are in the same front, then the one from a less crowded region is preferred, which ensures that the solution candidates from less crowded regions are proposed for selection by the algorithm. [30]

2.1.5.4 Strength pareto evolutionary algorithm 2 (SPEA2)

SPEA2 is an improved version of the SPEA algorithm. The pseudo-code of the SPEA2 algorithm is presented in Algorithm 2. There is two population in this algorithm: The first one is used for evolutionary optimization process and the other one is used as an archive for the best individuals. The fitness function of this method is derived by using two attributes for each individual: The first one is the number of individuals that dominate this individual, which is called the strength of the individual, and the second one is the number of individuals that are dominated by the current individual, which is the strengths of individuals that dominate this one. The fitness assignment of each individual is calculated by summing up these two attributes. [30]

Algorithm 2 SPEA2 algorithm [30].

1: **procedure** SPEA2 $(N', g, f_k(X)$
2: Initialize Population $\mathbb{P}'$
3: Create empty external set $\mathbb{E}'$
4: **for** $i = 1$ to g **do**
5: Computer fitness of each individual in $\mathbb{P}'$ and $\mathbb{E}'$
6: Copy all individual evaluating to nondominated vectors $\mathbb{P}'$ & $\mathbb{E}'$ to $\mathbb{E}'$
7: Use the truncation operator to remove elements from E when the capacity of the file has been extended
8: If the capacity of $\mathbb{E}'$ has not been exceeded then use dominated individuals in $\mathbb{P}'$ to fill $\mathbb{E}'$
9: Perform binary tournament selection with replacement to fill the mating pool
10: Apply crossover and mutation to the mating pool
11: **end for**

2.1.5.5 Non-dominated sorting genetic algorithm III (NSGA3)

This method is an improved version of the NSGA2 algorithm. The main advantage of this method is its better results on many-objective optimization problems [31]. The pseudo-code of the one-generation creation process of the algorithm is presented in Algorithm 3. There does not exist an explicit selection operator of the method. Reference directions are created on objective space, and individuals that best present these references are selected.

2.2 Nuclear Magnetic Resonance (NMR)

The three-dimensional structure of macromolecules provides essential information about the function, shape, and domain structure of the macromolecules. It has a vital role in drug design since it can also provide information about the interactions of proteins between each other and with small ligands. Nuclear Magnetic Resonance (NMR) is one of the ways to determine the three-dimensional structure of macromolecules.

Once compared to the other structure determination method, X-ray crystallography, NMR has some advantages: NMR does not need crystallization. It can produce protein

structure in a liquid environment, which is similar to the natural circumstances of the molecules. Additionally, it does not need any additional time for crystallization.

Algorithm 3 Generation *t* of NSGA-III procedure [31].

Input: H structured reference points Z^s or supplied aspiration points Z^a, parent population P_t
Output: P_{t+1}
1: $S_t = \emptyset,\ i = 1$
2: $Q_t =$ Recombination + Mutation (P_t)
3: $R_t = P_t \cup Q_t$
4: $(F_1, F_2, \dots) =$ Non-dominated-sort (R_t)
5: **repeat**
6: $S_t = S_t \cup F_i \ and \ i = i + 1$
7: **until** $|S_t| \geq N$
8: Last front to be included: $F_l = F_i$
9: **if** $|S_t| = N$ **then**
10: $P_{t+1} = S_t$, break
11: **else**
12: $P_{t+1} = \cup_{j=1}^{i-1} F_j$
13: Points to be chosen from $F_l : K = N - |P_{t+1}|$
14: Normalize objectives and create reference set $Z^r : Normalize\ (f^n, S_t, Z^r, Z^s, Z^a)$
15: Associate each member s of S_t with a reference point: $[\pi(s), d(s)] = Associate(S_t, Z^r)$ % $\pi(s)$: $closest\ reference\ point$, d: $distance\ between\ s\ and\ \pi(s)$
16: Computer niche count of reference point $j\ \epsilon\ Z^r$: $\rho_j = \sum_{s \varepsilon S_t / F_l}((\pi(s) = j)?1:0)$
17: Choose K members one at a time from F_l to construct P_{t+1}: $Niching\ (K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$
18: **end if**

However, NMR needs very advanced computational techniques to produce protein structures. NMR does not automatically produce the three-dimensional structure of proteins, but it produces signals on its spectrum. These output signals need to be processed by computational methods to produce the three- dimensional structure of the macromolecule that is being investigated in the NMR spectrum. This processing is time-consuming and makes NMR spectroscopy hard to handle. For that reason, there are some computational techniques developed to deal with this complexity of NMR spectroscopy. In this chapter, we will provide a detailed background about NMR spectroscopy and its processing steps. Afterwards, we will focus on the chemical shift assignment problem of NMR spectroscopy.

2.2.1 NMR hardware

NMR spectroscopy is based on the magnetic properties of certain atoms' nucleus. NMR spectrometer example is given in Figure 2.8. An NMR spectrometer is composed of a magnet that can produce a strong magnetic field, a sample tube, a probe, a radio frequency transmitter, a radio frequency receiver, an ADC (analog-to-digital converter), and a computer. This hardware is shown in Figure 2.9. In NMR spectroscopy, the molecule sample is placed in the tube. The magnet is composed of a closed-loop "solenoid" of superconducting wire that is placed in liquid helium.

Figure 2.8 : 950 MHz Liquid State NMR Spectrometer in BMRZ, Goethe University, Frankfurt am Main, Germany [32].

A large current that flows through the solenoid creates a continuous and strong magnetic field without additional power. The probe is indeed a coil of wire that goes to the tube and surrounds it. The probe receives the radio frequency signals from the tube and transmits the radio frequency signals to the tube. While the sample is under the strong magnetic field, the computer triggers the transmitter to send a short duration pulse of radio frequency through the probe to the sample. Once the pulse is sent, the probe coil receives a weak signal called free induction decay (FID). This FID signal is amplified into an audio frequency signal. ADC receives this audio signal and converts it into a digital FID signal. Many pulses are applied to the sample and after each pulse,

this digital FID is transferred into the computer. The computer collects the digital FID signals after each pulse. Then the computer applies Fourier transform to these signals and creates the resulting spectrum. [33]

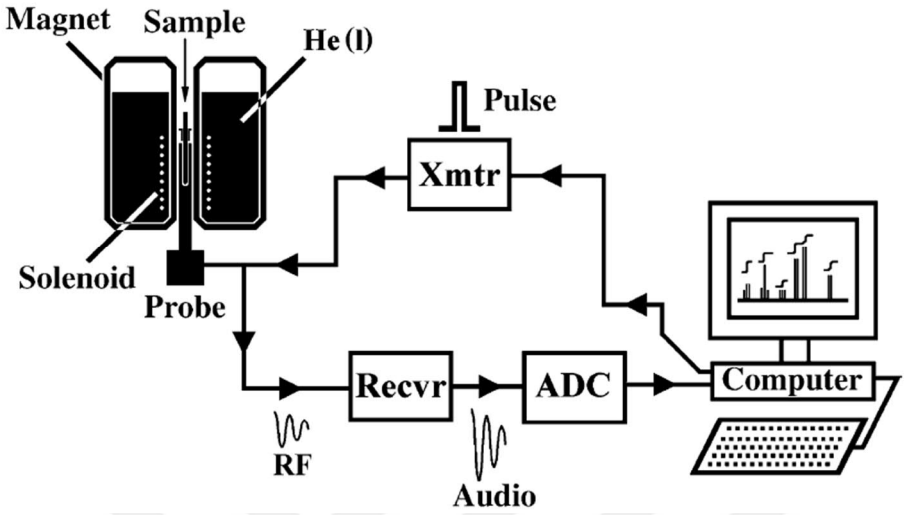

Figure 2.9 : NMR spectrometer hardware [33].

2.2.2 NMR principles

Certain atoms behave as their nucleus is spinning. This movement creates a magnetic field around their nucleus. Because of their charge and spin, their nucleus behaves like a magnet. This behavior occurs in some nuclei with an odd number of protons or an odd number of neutrons. The nuclei that can be observed on NMR spectroscopy are hydrogen ($_1$H), carbon (stable isotope $_{13}$C), nitrogen (stable isotope $_{15}$N), fluorine (stable isotope $_{19}$F), phosphorus (stable isotope $_{31}$P), and iron (stable isotope $_{57}$Fe). A strong magnetic field is applied to the sample in the tube in NMR.

When the magnetic field is applied to these spinning nuclei, they can have two possible orientations aligned with the direction of the magnetic field. Some of these nuclei can position with the direction of the field, whereas some of them can position against the direction of the magnetic field. This orientation comes from the quantum theory that a spinning particle can have two states of energy: lower energy state and higher energy state, shown in Figure 2.10 [33]. The energy state where the nucleus is aligned with the direction of the magnetic field is called the lower energy state "α" and the energy state where the nucleus is aligned against the magnetic field is called the higher energy state "β". The energy difference between these two states is proportional to the strength of the external magnetic field. [33]

The electrons around nuclei have a shielding effect on nuclei and they reduce the impact of the magnetic field on nuclei. If there is no electron around a nucleus, this nucleus is completely aligned with the direction of the external magnetic field. As a result, it needs higher energy to come to a higher energy state. This nucleus needing higher energy is called “deshielded”. On the other hand, if there are many electrons around a nucleus, this nucleus is not that much impacted by the external magnetic field and it does not need that much energy to come to the higher energy state. This nucleus is called “shielded”.

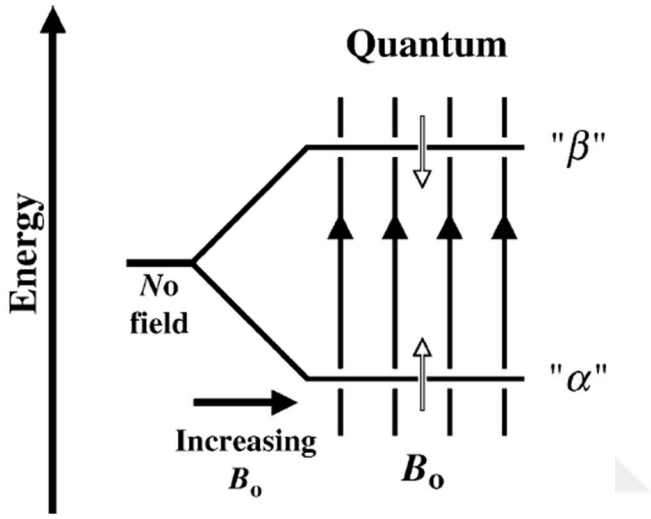

Figure 2.10 : Two quantum energy states: lower energy state “α” and higher energy state “β” [33].

NMR spectrometer measures the energy needed for a nucleus to come from the lower energy state to the higher energy state. This energy provides information about the environment of the nucleus: If a nucleus is surrounded by electrons, the energy to bring it from the lower energy state to the higher energy state is lower than the energy needed for a nucleus that has a fewer number of electrons around it. NMR spectrometer applies a short-duration pulse to the sample via its transmitter. These pulses bring nuclei from the lower energy state to the higher energy state. Directly after the pulse is stopped, these nuclei release the energy that is received as a weak signal by the probe. This weak signal is called free induction decay (FID). The probe sends this signal to the receiver that amplifies it to an audio frequency signal. ADC converts the audio frequency signal into a digital FID signal. After the Fourier transform, this signal is converted from the time domain into the frequency domain, which is the NMR spectrum. The transition between the energy levels shown in Figure 2.11 creates a line

on the NMR spectrum. A line on the NMR spectrum is called peak. According to the equation in the figure, such a transition in the energy levels is caused by a photon of light, whose frequency is *v*. The *hv* is equal to the difference between the energy levels. The *h* value here is Planck's constant that is a universal constant. [34]

The number of atoms that correlate together and give rise to a peak on the NMR spectrum is the dimension of that spectrum. There are different types of NMR spectra depending on their hardware, such as COSY (correlation spectroscopy), NOESY (nuclear Overhauser effect spectroscopy), and HNCA spectra. Depending on the NMR spectrum type, the peaks on the spectrum show different correlations between different atoms in the sample. Some of the NMR spectrometers give rise to a peak on either correlating atoms that are connected by one, two or three chemical bonds. In contrast, some NMR spectrometers give rise to a peak for the atoms that are close together in space (e.g. with distance < 5 Å = 5 x 10^{-10} m). For example, the COSY spectrum is a two-dimensional spectrum where hydrogen atoms separated from each other by less than or equal to three covalent bonds are expected to give rise to a peak on the spectrum.

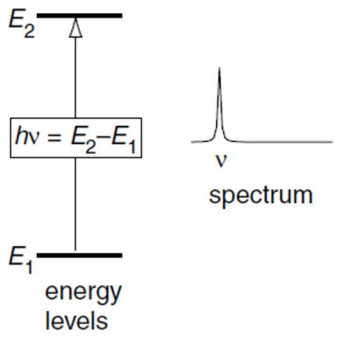

Figure 2.11 : The energy difference between the two energy states is observed as a line on the NMR spectrum [34].

2.2.3 NMR experiment steps

Nuclear magnetic resonance (NMR) spectra are n-dimensional (n = 2, 3, 4) data matrices. Each dimension is a frequency axis representing the resonance frequency of a nuclear spin in a strong, constant magnetic field. Traditionally, frequencies are specified as "chemical shifts", measured in parts per million, ppm. The NMR spectra contain signals, the "measured peaks" (or, equivalently, "observed peaks" or "experimental peaks"). Each peak has n chemical shift coordinates, corresponding to the resonance frequencies of n nuclear spins or atoms, which are correlated by the

peak. The resonance frequencies are very different for different types of atoms, e.g., in proteins hydrogen (^{1}H), carbon (stable isotope ^{13}C), and nitrogen (stable isotope ^{15}N). However, they differ slightly (by a few ppm) between different atoms of the same chemical element in a protein because of their slightly different environment. If two peaks involve the same atom, they will share the same chemical shift coordinate in one of their dimensions, i.e., peaks involving the same atom are aligned in the NMR spectra.

NMR spectroscopy is based on determining the magnetic properties of atom nuclei under a strong magnetic field. In an NMR experiment, a strong magnetic field is applied to the molecule in the experiment tube. Depending on the behavior of the nuclei in this environment, the atom-atom distances of the molecule are tried to be assigned by using computational techniques. The steps of a standard NMR experiment are explained in detail below [35].

Sample Preparation: This first step of an NMR experiment is to prepare the sample. The concentration, ionic strength, PH value, and temperature are important criteria for the sample preparation, especially to achieve good results from the NMR experiments.

Data Collection: The second step is the data collection from the NMR spectrum. In this step, firstly, field/frequency lock is applied to the sample. Even if the magnetic field produced by the NMR magnet is fixed, there might be some changes in the field. It is ensured with this locking functionality that the sample is examined under a stable magnetic field during the calculation. Secondly, the probe has to be tuned to the correct impedance, and the resonance frequency since replacing a sample tube into the receive/transmit coils affects the tuned circuit. In order to create a homogenous magnetic field for NMR experiments, additional room-temperature electromagnetic coils, shim-coils are placed around the sample tube because the superconducting magnet of the spectrometer cannot provide the optimum stable magnetic field alone. This process is called shimming.

In NMR experiments, multiple pulses are applied to the biological sample. Calibration of these pulses with the correct flip angle is accomplished next. After these steps are completed, acquisition parameters of the NMR spectrum are calculated. These parameters include the transmitter offset, spectral width and

correlation times. Eventually, after all of these preparations are completed and the attributes are fixed, the protein sample is exposed to a strong and stable magnetic field and a sequence of radiofrequency pulses. NMR experiment is composed of several scans that are combined to produce the complete experiment result. In each scan different pulse sequence is applied to the same sample. The overall result of these scans produces experiment-specific signals demonstrating the covalent and spatial connectivity of the atoms of the sample in the tube.

Data Processing: The results of the NMR experiments are time-domain signal that has been examined in the scans mentioned above and then they are converted into digital format. These time-domain data is converted into the frequency domain by using Fourier transformation.

Peak picking: Signals originated from different atoms may overlap on the NMR spectrum. It is hard to detect such signals on the overlapped regions. There might also be noise and artifacts on the NMR spectrum, which should be filtered out from the NMR results. In this step of the NMR spectroscopy, the signal in the frequency domain is filtered out from artifacts and noise in the experiment environment in order to observe the actual signals of the samples' atoms. The results of this process are inputs to the next steps of the assignment, so the correct peak picking is a crucial step for the overall success of structure determination. Several programs, such as CAPP [36], AUTOPSY [37], XEASY [38], and CYPICK [39] have been developed for the peak picking step, which provides sustainable results.

Chemical Shift Assignment: Each atom of the protein has a specific chemical shift value originated from the chemical environment of its nucleus. Even if the positions of the observed peaks are known, it is unknown which chemical shift is originated from which atom of the sample. The chemical shift assignment step maps the measured peaks to the expected peaks. The scope of this thesis is based on this problem, and its details are explained in the following subchapter.

NOE (Nuclear Overhauser Effect) assignment: The chemical shift assignments of the atoms are used in this step to determine the atom-atom distances of the sample's atoms in the three-dimensional structure. The distance restraints are derived from the volumes of the peaks. ^{1}H atoms that are separated from each other less than 5 Å result a signal on the NMR spectrum, which is called peak. The

distance between these atoms is inverse proportional to the peak volume between these atoms. These distance restraints are calculated in this step on NOESY (Nuclear Overhauser Effect Spectroscopy) spectra using the results of the chemical shift assignment, which indicates which peak belongs to which atom pairs. Some methods are working fine for the NOE assignment, such as PASD [40], ASDP [41] and CANDID [42]. One of the sample results of this step is depicted in Figure 2.12.

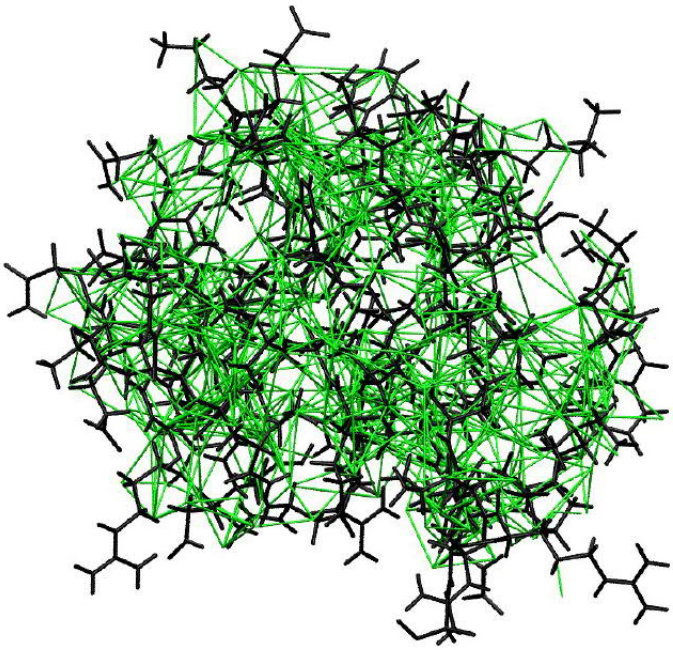

Figure 2.12 : Distance restraints that connect the hydrogen atoms are in greed and the protein atoms are in black. This sample is obtained from $_{13}C$ and $_{15}N$ NOESY spectra [43].

Structure Calculation: Three-dimensional structure of the protein depends on the atom-atom distance constraints obtained in the previous NOE assignment step. There have been many pieces of research on this area and many programs such as CYANA [44], Xplor-NIH [45], and CNS [46], whereas CYANA is the most widely used one. CYANA uses the atom-atom distance and the amino acid sequence of the protein to calculate the arrangement of the atoms in space. Once the three-dimensional structure is calculated, iterative correction activities are employed. The structure information is used in the NOE assignment, peak picking and chemical shift assignments to improve the structure results.

Structure Refinement: After several iterations, the result of the NMR experiment is achieved: three-dimensional structure of the sample in solution.

The steps of a standard NMR experiment with the CYANA framework are given in Figure 2.13. As shown in the figure, the sample preparation and data collection steps are done before the CYANA framework. The CYANA framework has the FLYA steps for the peak picking and the chemical shift assignment steps of the experiment. This thesis will only work on the chemical shift assignment step of the whole process, marked with red, as EMY (Emel Maden Yılmaz) in the figure.

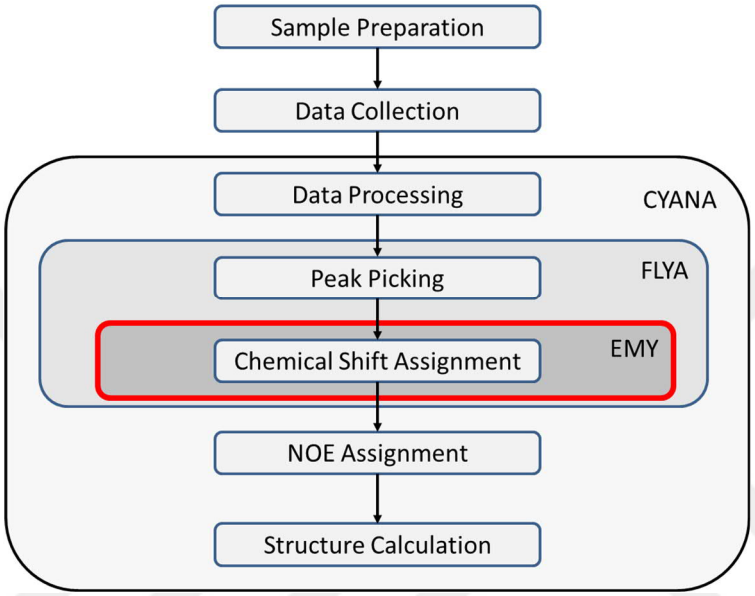

Figure 2.13 : NMR experiment steps [35].

2.2.4 Chemical shift

NMR spectrum could be plotted as a function of frequency that is the output of the calculations mentioned above. However, it would not be convenient to use these numbers since they might be very long digit values, and they are not standard and dependent on the magnetic field of the NMR spectrum. So, without mentioning the magnetic field value of the NMR spectrum, they would not make any sense. So, instead of the frequency function, the chemical shift scale is used in the NMR spectrum, where the frequencies are shown with respect to a reference compound. This reference compound is mostly the organic compound TMS (tetramethylsilane) in the NMR spectrum. The peak coming from TMS on NMR experiments is accepted as the reference line for the spectrum and the other values on the spectrum are shown relative to the reference compound TMS. This approach is shown in Figure 2.14.

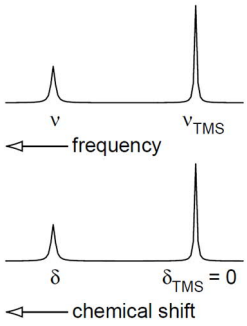

Figure 2.14 : NMR spectrum as a function of frequency is the one above. The chemical shift scale is used in general as an NMR spectrum [34].

The relative values for each compound to the reference compound are calculated as in equation 2.1. The ν value here shows the frequency of the compound, and the ν_{TMS} value is the frequency of the reference compound, TMS. The result, δ, shows the chemical shift of the compound.

$$\delta = \frac{\nu - \nu_{TMS}}{\nu_{TMS}} \tag{2.1}$$

Both the nominator and denominator show the frequency, so they have the same unit, Herz. After the division, the result shows a unitless ratio and does not depend on the magnetic field of the NMR spectrum. Since the chemical shift is very small in almost all of the cases in the NMR spectrum, it is multiplied by 10^6. The unitless result is the delta value, δ, named as parts per million, ppm. The calculation of chemical shift in ppm is shown in equation 2.2. This representation in NMR spectroscopy is called the delta scale. [34].

$$\delta_{ppm} = 10^6 \times \frac{\nu - \nu_{TMS}}{\nu_{TMS}} \tag{2.2}$$

2.2.5 Chemical shift assignment problem

The chemical shift assignment problem is related to the assignment of two peak sets in the NMR spectroscopy: expected peaks and measured peaks.

Expected peaks: For a given NMR experiment type, and a given protein, one can predict which n-tuple of atoms a peak is expected to appear in the spectrum. The expected peaks are the sets of atoms that are expected to give rise to a peak on that NMR experiment. Therefore, we can generate for each NMR experiment a list of "expected peaks". The expected peaks are defined by using the amino acid sequence

of the molecule, the covalent structure of amino acids from the protein data bank library, and the magnetization patterns of the NMR experiment [15]. For example, hydrogen atoms that are separated from each other by less than or equal to three covalent bonds are expected to give rise to a peak on a COSY spectrum. H-X-X-H represents the pattern of this connectivity. The expected peaks are not only determined by the patterns but also the intra-residual distances of the atoms. As patterns can determine through-bond expected peaks, through-space expected peaks are difficult to define. Existing expected peak list creation methodologies produce good results, so these will be used to produce the expected peak list [15] as an input to the chemical shift assignment problem in this thesis. The atoms that are covered by each expected peak are known precisely; however, we cannot a priori know where the peak will appear in the spectrum because we do not know the chemical shifts of the atoms yet.

Measured peaks: Measured peaks are the observed peaks on NMR spectra. In other words, measured peaks are the output of the NMR spectroscopy that shows where exactly on the spectrum the peaks, or signals, occurred. An example of measured peaks in a two-dimensional NMR spectrum is given in Figure 2.15.

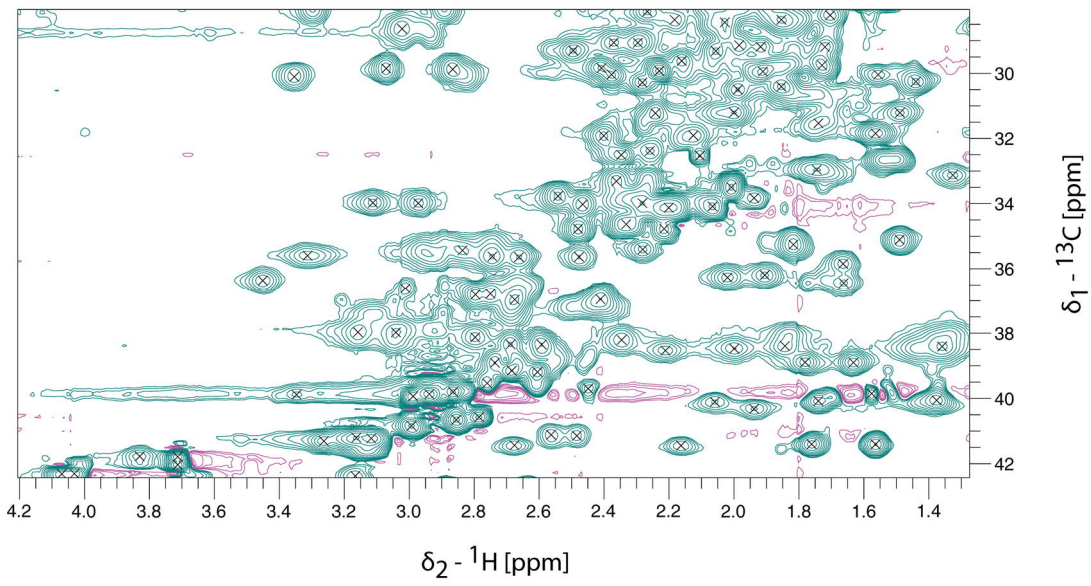

Figure 2.15 : Two-dimensional NMR spectrum measured peaks, where the picked peaks are marked with x [47].

In NMR spectra, there might be artifacts and noise, whereas signals from different atoms may overlap. So, the peak picking step is employed that removes these kinds of signals and picks only the peaks that are relevant for the NMR structure calculation process. The remaining peaks on the NMR spectrum are taken to the structure

calculation step as measured peaks. The specific positions of the measured peaks are known on the NMR spectrum, i.e., the chemical shift frequency for each dimension of each of the measured peaks is known. However, the atoms that are causing these peaks are unknown.

The chemical shift assignment problem is represented in Figure 2.16. The expected peaks are shown on the right side of the figure, and their atoms are known. However, their positions on the spectrum are not known precisely. Measured peaks of the spectrum are shown on the left side of the figure. Here, the position of each measured peak is known. However, the atoms giving rise to these peaks are not known. The chemical shift assignment problem is making the assignment from the expected peaks to the measured peaks to define the chemical shift frequency values of the atoms in the sample.

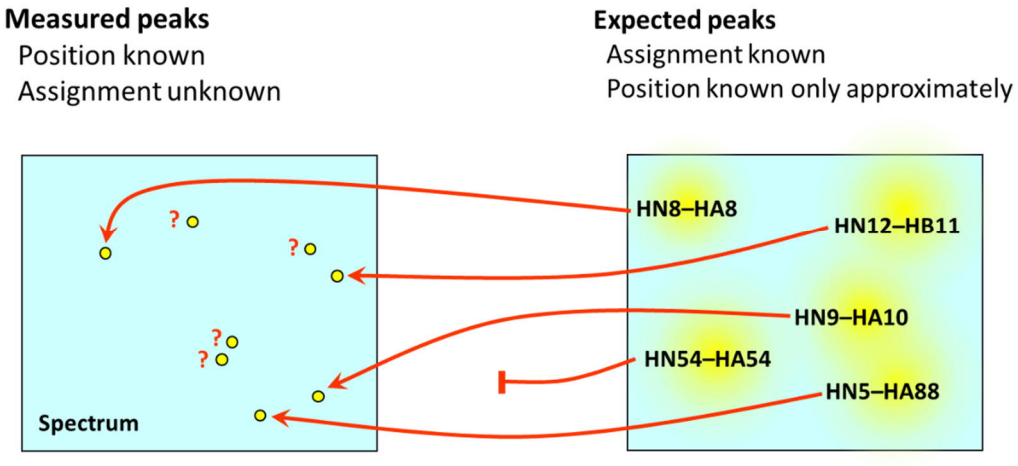

Figure 2.16 : The chemical shift assignment problem finds mapping from expected peaks to measured peaks [44].

The expected peak carries the names (identities) of the n atoms, and the position of the measured peak provides their chemical shift frequencies, where n is the dimension of the NMR spectrum. Each assignment from the expected peaks to the measured peaks yields the chemical shift frequency assignments of the n atoms involved.

The chemical shift assignment is a prerequisite for all NMR studies of proteins because they allow defining the chemical shift frequencies of the atoms in a protein. On the other hand, the chemical shift assignment problem is cumbersome and requires sophisticated computational methods to find a solution. Chemical shift assignment provides essential information to the three-dimensional structure calculation programs for further processing. Even if the structure calculation methods perform very well,

they cannot create a correct three-dimensional structure with wrong chemical shift assignments. As a result, chemical shift assignments are very crucial information for the complete NMR experiment.

2.2.6 Protein structure

The chemical shift assignment problem and the whole protein structure determination process are strongly related to the protein structure and its formation from residues. In order to understand this process, a brief introduction about the protein structure is presented in this subchapter.

There are four basic levels for presenting protein structure: Primary, secondary, tertiary, and quaternary structures. Primary structure defines the amino acid sequences and the composition of protein from its amino acids. Secondary structure studies the local interactions between some stretches of protein. Tertiary and quaternary structures define the three-dimensional structures and three-dimensional interactions of proteins, respectively. In the chemical shift assignment problem, only the primary structure of proteins is studied, so this chapter will focus on the primary structure of proteins. Proteins are composed of amino acids connected one by one. The primary structure of a protein depicts the atoms in the protein and the bonds between these atoms.

The general formula of amino acid in the primary structure is given in Figure 2.17. As shown in the figure, one amino acid contains four different building blocks: Amino Group (NH_3^+), Hydrogen Group (H), Carboxyl Group (COO^-), and Radical Group (R). The carbon atom in the middle of the structure is called a central tetrahedral carbon atom, also known as alpha carbon (C^α) and all of the amino acid groups are bonded to the alpha carbon. The amino, hydrogen, and carboxyl groups of all amino acids are the same. However, the radical groups differ from each other.

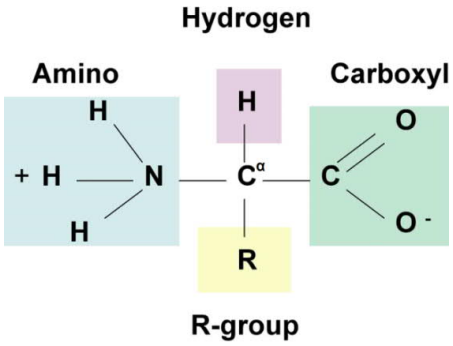

Figure 2.17 : Amino acid general formula [48].

Amino acids create a chain and connect to form protein. The naming of the peptides comes from the number of amino acids connected. Two amino acids connected to each are called a *dipeptide*, whereas more than two bonded amino acids construct a *polypeptide*. The formation of a dipeptide is displayed in Figure 2.18. As seen in the figure, the oxygen atom in the carboxyl group of the first amino acid bonds to the two hydrogen atoms in the amino groups of the following amino acid. After this bonding, one H_2O molecule will be released, and the carboxyl group of the first amino acid is bonded to the amino group of the second one. This bond connecting two amino acids is called a *peptide bond.*

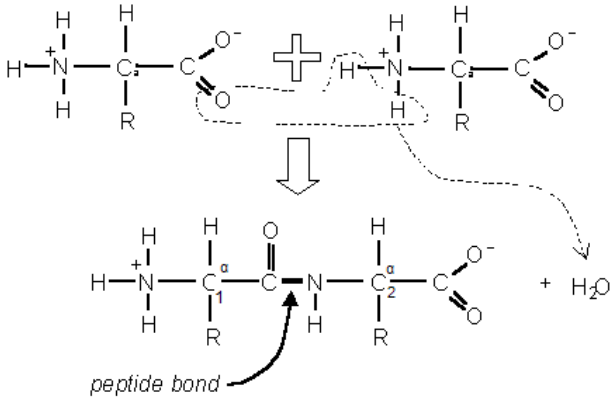

Figure 2.18 : Formation of a dipeptide [48].

If more than two amino acids bond to each other, the same process will be followed: One H_2O molecule will be released. A peptide bond will bond the carboxyl group of one amino acid to the amino group of the other one. This repeating formula of the polypeptide formation is given in Figure 2.19. Once more than one amino acid bonds to each other, the core part of the amino acid remains in the amino acid after the release of the H_2O molecule is called *residue*. The parts between the parentheses in the figure are the residue of amino acid. [48]

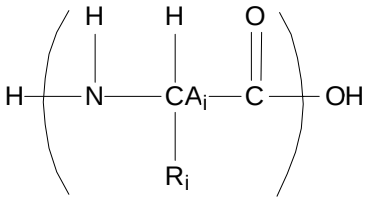

Figure 2.19 : Polypeptide formation formula [48].

2.3 Literature Review

There have been various studies on the chemical shift assignment problem, and they use different methodologies of computer science, including methods derived from artificial intelligence [9], peak-particle dynamics simulation [49], probabilistic interaction network of evidence algorithm [50], and simulated annealing [51]. For small and medium-sized proteins only, there are some methodologies with exhaustive search algorithms [52,53]. Heuristic algorithms have been adopted in some of the existing approaches [13,14,17]. In some of these researches, the evolutionary algorithms and local optimization algorithms are employed together and a hybrid model is developed [15,16]. Some of these researches did not provide a full assignment of the whole protein but only assigned the atoms on the protein's backbone and did not make any assignment on the sidechain atoms. [7,8,9,51]. Of these, only the program AutoAssign [9] has been used extensively in a structural genomics project [54] and reported in more than 200 PDB depositions [55]. However, both the sidechain and backbone atoms' chemical shift frequency assignments are required for a complete three-dimensional structure of a protein. Some of these methods provide a full assignment for both backbone and sidechain atoms; however, they [12-14,52,53] have a long computation time and cannot be applied to the large size of proteins. Five algorithms apply to the automated assignment of all (backbone and side-chain) resonances exclusively from peak lists [15,49,50,56-59]. However, Guerry and Herrmann showed in [55] that until 2011 only two of the pure peak list-based algorithms for complete resonance assignment, PINE [50] and GARANT [15], have been used in actual protein structure determinations deposited in the Protein Data Bank (PDB). This chapter presents an overview of these approaches along with their drawbacks and strengths.

The research [6] has developed a program PASTA (Protein ASsignment by Threshold Accepting), which formulated the assignment problem as a combinatorial optimization problem and tried to solve the problem with a combinatorial optimization strategy called threshold accepting, which is a superior approach to the simulated annealing. The handicap of this program is that only the backbone atoms are kept in the scope, and sidechain atoms are exempted. The sidechain atoms are excluded since their assignment was not suitable with this method.

The research [7] used Bayes' theorem to formulate the scoring functions of the problem's objectives. However, this approach could also work on the backbone atoms and could not assign the sidechain atoms of proteins. The research [8] performed a simulated annealing approach to solve the chemical shift assignment problem. This approach did concentrate on the assignments of the backbone atoms only and addressed the assignment of the sidechain atoms in further work.

The paper [60] has developed a program, MARS, for the automatic assignment of the backbone atoms. MARS uses exhaustive search to find out all possible assignments and then search for the best possible option to find out the optimum solution. As already noted, MARS does not work on the sidechain atoms.

The research [10] has created a solution to the assignment problem with an exhaustive combinatorial search algorithm. However, this method can make the assignment only on the backbone atoms. Additionally, it can work only for 80 residues or less.

Another exhaustive search method is developed in the paper [11]. Here a new exhaustive search algorithm TATAPRO (Tracked AuTomated Assignments in Proteins), is developed. However, this approach has a significant time complexity and cannot make the assignment of the whole side-chain atoms.

The research CAMRA (Computer-Aided Magnetic Resonance Assignment) [12] has developed a method for the assignment problem by matching the predicted chemical shifts to the observed spin systems. This approach has its handicap that it can work only on small- and medium-sized proteins and assigns only the backbone atoms.

Some research tried to solve this problem with heuristic best-first search algorithms: [13,14]. These methods are applicable only on small proteins though.

The paper [17] has developed a heuristic algorithm to solve the problem; however, it also performs only on the auto-assignment of the backbone atoms.

AUTOASSIGN [9] is an extensively used algorithm for solving the reduced problem of protein backbone resonance assignment using methods from artificial intelligence. It combines symbolic constraint satisfaction methods with a domain-specific knowledge base to exploit the logical structure of the sequential assignment problem, the specific features of the various NMR experiments, and the expected chemical shift frequencies of different amino acids. This paper showed that AUTOASSIGN could

produce good results only for the backbone atom assignment of small and medium-sized proteins.

DYNASSIGN [49] is another algorithm for the automated assignment of NMR chemical shift resonances. Peak particles represent expected cross-peaks in multidimensional NMR spectra, and assignment restraints are translated into a potential energy function. Molecular dynamics simulation techniques are used to calculate a trajectory of the system of peak-particles subjected to the potential function to find energetically optimal configurations that correspond to correct assignments. Peak-particle dynamics-based simulated annealing was combined with the Hungarian algorithm [61] for local optimization. This approach showed convergence problems for any but the smallest proteins or peptides.

One notable algorithm for general automated resonance assignment is PINE [50]. PINE uses estimates of evidence derived from empirical distributions of previously observed data, along with consistency measures, to drive a fictitious system M with Hamiltonian H to a quasi-stationary state that produces probabilistic label assignments for relevant subsets of the data [50]. PINE was used for converting peak lists extracted from various NMR experiments into assignments associated with probabilities for their correctness. The PINE-NMR server (http://pine.nmrfam.wisc.edu) accepts the sequence of the protein plus user-specified combinations of data corresponding to an extensive list of NMR experiments as input. It provides a probabilistic assignment of NMR chemical shifts to the sequence-specific backbone and aliphatic side-chain atoms as output. Using the same input data, a comparison of PINE with FLYA has clearly shown superior performance of FLYA [16].

The first primary automatic algorithm is GARANT [15]. GARANT algorithm represents the resonance assignment problem as an optimal match of two graphs, which are the observed peaks on the NMR spectrum and the expected peaks. The expected peak list is calculated depending on the inner structure of the protein. For example, all proton pairs separated by 2-3 covalent bonds produce a peak on a COSY spectrum. Depending on these predefined rules and patterns, expected peaks of the specific spectrum are calculated by the algorithm. Observed peaks are already available from the NMR experiment. GARANT uses the concept of mutual information for the assignment process [62]. The mutual information between the expected peaks and observed peaks is handled as a score of the quality of the complete

assignment. So, the assignment that results from the maximum value for the mutual information is a correct one. Evolutionary optimization was combined with local search to find an automatic way for the resonance frequency assignments. In each generation of the evolutionary algorithm, local optimization is performed on the selected parent solutions. The algorithm was, in principle, generally applicable but required a lot of computation time and was cumbersome to adapt to new types of NMR spectra. Especially choosing optimal weights for different terms of the mutual information was difficult.

Then the FLYA algorithm was developed [16]. This algorithm represents the problem as a graph, where the nodes are the union of all of the atoms and all of the expected peaks. The edges of the graph represent the expected peaks. In other words, an edge shows that there is an expected peak between these adjacent atoms. The FLYA algorithm generates a network of expected peaks from the protein sequence and the magnetization transfer pathways of a set of NMR experiments. It then computes a mapping from this network to the measured peaks, which implies an assignment of the measured chemical shifts to atoms. FLYA tries to find an optimum mapping from the expected peaks to the measured peaks to solve the problem. As in GARANT, an evolutionary algorithm is combined with a local optimization routine. FLYA program is the default algorithm in the CYANA framework for solving the chemical shift assignment problem. The proteins existing in the Protein Data Bank (PDB) [5] can be converted into the CYANA library format by using the CYLIB algorithm [63] and can be used in the framework without any manual interactions. Two types of scoring schemes are defined in FLYA [64-66]: The first one is the global scoring, which calculates the score of the complete assignment solution candidate, and the second one is the local scoring scheme that evaluates the score of the single atoms in an assignment. The evolutionary algorithm uses the global scoring scheme, whereas the local optimization routine uses the local scoring scheme. The global score of a solution candidate is derived by using four attributes of an assignment: the chemical shift value of an assigned atom, the variety of the chemical frequencies mapped to the same atom, the ambiguity of the assignment, and the number of assigned peaks. The optimization algorithm in FLYA is a reimplementation of the procedure that had been developed for GARANT. It is combined with a more flexible network representation and a new scoring scheme for assignments. The FLYA algorithm has been implemented in

FORTRAN, and several specialized applications have been published [43,64-68]. FLYA makes an assignment on both the backbone and the sidechain atoms. The FLYA algorithm showed the most promising results so far among these algorithms. [16]

3. NEW METHOD ON CHEMICAL SHIFT ASSIGNMENT

In this thesis, we analyze the chemical shift assignment process of the protein structure calculation process in NMR experiments. The chemical shift assignment process assigns the expected peaks of the NMR experiment to the measured chemical shift frequency values on the NMR spectrum. The chemical shift frequency values for the atoms in the sample will be determined as a result of this assignment. The chemical shift assignment provides very crucial information for the protein structure calculation, and it cannot be possible to create the correct protein three-dimensional structure with the wrong chemical shift assignment. Therefore, the chemical shift assignment step is critical for the complete success of the protein structure calculation process. On the other hand, the chemical shift assignment is a very cumbersome process and needs very qualified computational methods to succeed.

In this thesis, we accept that the rest of the steps of the protein structure determination are working perfectly, such as peak picking, NOE assignment, and structure calculation. We will investigate only the chemical shift assignment step of the protein structure determination in this study. We took the chemical shift assignment problem as a standalone problem and provided our solution as a standalone solution to this problem.

3.1 Algorithms

The chemical shift assignment searches for solutions in a huge search space, where searching the possible combinations of solution candidates very hard and mostly not possible. Therefore, instead of systematic search approaches, we applied heuristic methods as a solution for this problem. Among the heuristic algorithms, we selected the evolutionary algorithms for this problem. Since the evolutionary algorithms already work with a set of solution candidates, called population, they appeared to be a suitable match for this problem. We also separated the objectives of the problem from each other and observed the results on a multi-objective optimization problem structure.

We implemented our solution on the MOEA framework [69], which is an open-source Java library of multi-objective optimization algorithms. We designed this problem model on the MOEA library with an object-oriented programming approach in java language.

We applied the Non-dominated Sorting Genetic Algorithm II (NSGA2), Non-dominated Sorting Genetic Algorithm III (NSGA3), and Genetic Algorithm (GA) of the MOEA framework in this thesis. We also developed a hill-climbing algorithm and combined it with the NSGA2, NSGA3, and GA algorithms to form a hybrid solution to achieve better results. Additionally, we developed a constructive heuristic algorithm for further optimization in our method.

3.1.1 NSGA2 algorithm

The NSGA2 algorithm [70] is a multi-objective evolutionary optimization algorithm that uses elitism and crowded comparison operators to keep the diversity in the search process. The algorithm is summarized in Algorithm 1. All of the individuals in the population are classified to a particular non-dominated layer in this algorithm. In order to classify all individuals, first all non-dominated individuals are classified into a layer, and these individuals are then separated from the population. Then, from the remaining individuals in the population, another non-dominated group is generated. This process continues until all individuals belong to a specific non-dominated layer. Each of these non-dominated layers gets a non-dominated rank, which is a dummy fitness value. The dummy fitness values in the lower layers are greater to guarantee that these individuals from the first layers have a better probability of reproducing.

Additionally, NSGA2 applies niching by using the crowding distance for the individuals, ensuring that diversity is also considered in the algorithm. The crowding distance gives information about the density of the other solution candidates around the specific individual. Selection is based on both random rank and crowding distance called crowded comparison operator. If two solution candidates from the same front are being compared, then the one from the less crowded region is selected. If they are from different fronts, then the one with lower rank is selected. This approach ensures that in case of the selections from the same front, the solution candidates from the less crowded regions are being selected, which will improve the diversity in the population.

Crowding distance represents the density of solutions surrounding a particular solution. Its insertion into the algorithms ensures diversity. Selection is based on both random rank and crowding distance called crowded comparison operator. In case two solution candidates are from different fronts, the one with lower rank is selected.

3.1.2 NSGA3 algorithm

The NSGA3 [71] algorithm is another multi-objective optimization algorithm, and its pseudo-code of the one-generation creation process of the algorithm is presented in Algorithm 3. The NSGA3 algorithm is originated from the NSGA2 algorithm, and it is specified for producing better results on the many-objective optimization problems, where there are four or more numbers of objectives in the optimization problem model. This method does not have an explicit selection operator. The reference directions are defined on the objective space of the algorithm. Them the individuals that best represent these reference directions are selected.

3.1.3 GA algorithm

The Genetic Algorithm [28] in MOEA is the standard genetic algorithm with a single objective optimization approach and elitism. We have modeled four different objectives in our problem model. These objective values are combined in one score function with the same weight in the Genetic Algorithm (GA), and one single objective value is achieved. The basic workflow of the GA algorithm was given in Figure 2.4.

3.1.4 Hill climbing and hybrid algorithms

The hill-climbing algorithm is the iterative improvement method in stochastic local search that iteratively seeks improvement in each step. We implemented a hill-climbing algorithm [19] from scratch for the chemical shift assignment problem.

Additionally, we combined the hill-climbing algorithm with the NSGA2, NSGA3, and GA algorithms to escape from local optima. As a result, we implemented a hybrid algorithm in the MOEA framework that runs in each of its iterations, one iteration of the NSGA2, NSGA3 or GA algorithms and one iteration of the hill-climbing algorithm.

3.2 Problem Model

Defining and finding the optimal correspondence between expected and measured peaks in a set of NMR spectra for a protein is not trivial. Its main reason is the fact that measured NMR spectra are not perfect: In general, they do not contain all peaks that one would expect to see, but, in addition, they also contain noise and artifact peaks, which do not correspond to any expected peak. Furthermore, it is difficult to predict the expected peaks for some NMR experiments theoretically. On the other hand, the chemical shift assignment step provides the essential information for the protein structure calculation programs, so it is an essential milestone in the structure calculation. In order to have a correct structure determination, the chemical shift assignment problem should be modeled successfully.

The chemical shift assignment problem is formulated so far in most of the researches as a combinatorial optimization problem, where finding an optimum mapping from the expected peak list into the measured peak list is the target. The representation of the assignment depicting the inputs of the assignment is given in Figure 3.1.

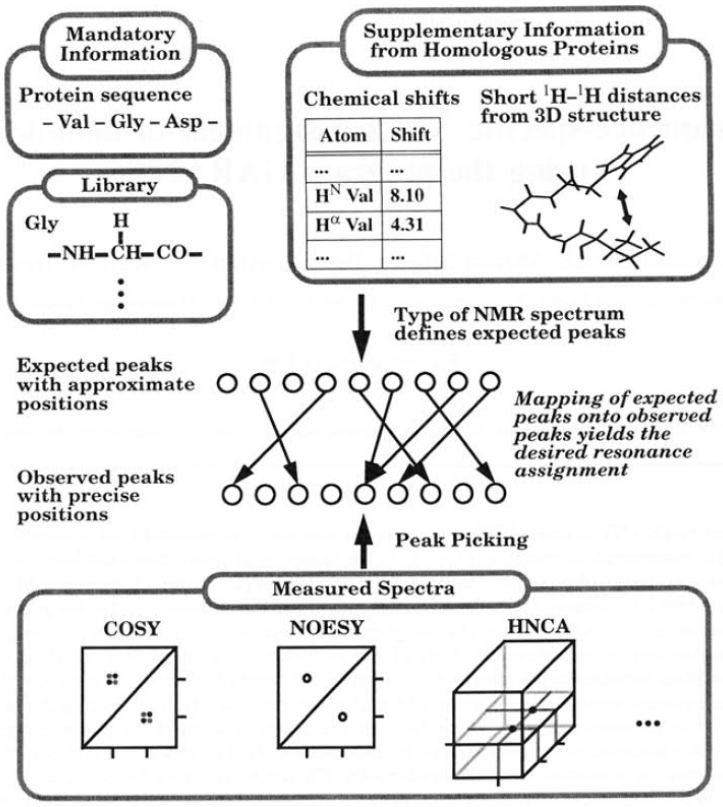

Figure 3.1 : The formulation of the chemical shift assignment problem [15].

This figure illustrates the chemical shift assignment problem by explaining the creation of the expected peak list and measured peak list. The problem is mapping the expected peaks to the observed, i.e., measured peaks so that desired chemical shift frequencies will be determined. First, the observed peak list of the sample under the NMR experiment in the defined spectra is needed, illustrated in the lower part of the figure. Each measured peak list includes the type of the spectrum and the positions of the peaks occurring on this specific spectrum. Peak picking is done on these measured peaks to exclude the artifact and noise from the spectra. The spectra on this example are COSY, NOESY, and HNCA. These lists provide precise positions, i.e., chemical shift frequencies, for the atoms in each measured peak. On the upper part of the figure, the expected peak list creation is explained. There is some information that needs to be provided for the expected peaks creation: The sequence of the amino acids in the protein, the structure of the residue in the protein data bank and supplementary information from homologous proteins such as standard chemical shift values of the atoms and standard atom-atom distances of the corresponding residues, which are obtained from general protein data bank libraries. The list of expected peaks for each spectrum is generated with this information.

Depending on the dimension of the spectrum, each peak associates with some number of atoms, in other words, the number of the axes to plot the data in space. For example, the NOESY spectrum is two-dimensional, meaning two atoms can produce a peak, and the results on the NOESY spectrum are depicted with two frequency axes. Similarly, the HNCA spectrum is three-dimensional, so it has three atoms behind each expected peak and three chemical shift frequency values behind each measured peak. On the other hand, one atom may appear in several peaks. All of the expected peaks that one atom can give rise in various experiments are called adjacent peaks of the atoms. The chemical shift assignment maps the expected peaks to the measured peaks, as illustrated in the middle of the figure.

3.3 Objectives and Constraints

The chemical shift assignment problem is an optimization problem where objectives must be optimized and constraints must be fulfilled. In this subchapter, the objectives and constraints of the chemical shift assignment problem are explained.

3.3.1 Objectives

The objectives of the chemical shift assignment problem are given in Figure 3.2.

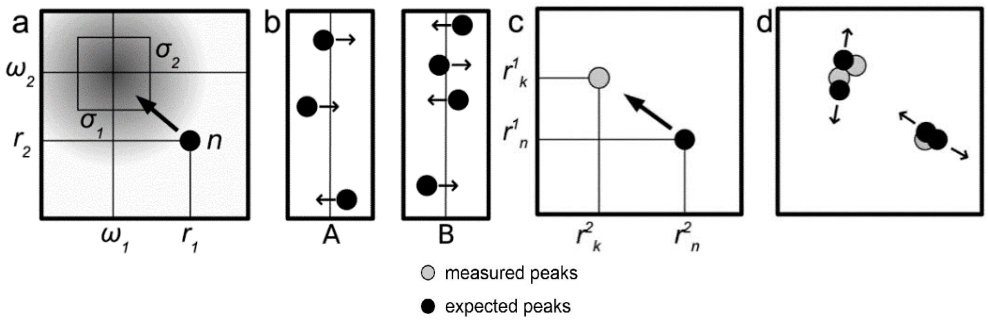

Figure 3.2 : The objectives of the chemical shift assignment problem: a) Shift normality b) Alignment c) Completeness d) Low degeneracy [72].

3.3.1.1 Shift normality

It is required that the assigned chemical shift values to the atoms are consistent with their general chemical shift statistics, which are taken from chemical libraries such as Biological Magnetic Resonance Databank (BMRB) for those atoms. The goal is to keep the chemical shift assignments of the atoms as close as possible to their general chemical shift statistics values.

3.3.1.2 Alignment

It is possible that the atoms in a spectrum will give rises to more than one peak. Each peak assignment will assign a chemical shift value to the specific atom. These chemical shift assignments coming from different measured peaks to a single atom should be aligned with each other as much as possible.

3.3.1.3 Completeness

Another objective of the problem is to assign as many peaks as possible, which is called completeness. Some expected peaks and observed peaks may be set as unassigned in the solution though.

3.3.1.4 Low degeneracy

The last objective of the problem is low degeneracy, which means that the number of degenerate peaks is small. Degenerate peaks are the expected peaks that are assigned to the same observed peaks. With an increased number of degenerate peaks, the number of the measured peaks covered by the assignment might decrease as well,

which is the completeness objective. The number of degenerate expected peaks assigned to the same observed peak is called ambiguity of the assignment.

3.3.2 Constraints

The following constraints exist in the chemical shift assignment problem:

An expected peak can be assigned to one measured peak only. In other words, it is not possible to map one expected peak more than one observed peak. However, several expected peaks can be mapped to the same measured peak. It is also possible that some of the measured peaks and expected peaks are not mapped at all. However, this situation might conflict with the completeness objective. If none of the adjacent expected peaks of an atom is mapped to a measured peak, then the chemical frequency of this atom remains undefined.

Expected peaks can be mapped to measured peaks of the same spectrum only. One expected peak of a spectrum cannot be assigned to a measured peak of another spectrum.

The variance of the chemical shifts assigned to an atom cannot exceed the given tolerance representing the accuracy of the measurement.

3.3.3 Evaluation

The goal for the shift normality, alignment, and completeness objectives is to maximize their values, whereas we need to minimize the degeneracy objective. The MOEA framework works only with minimization problems, so all of the maximization objectives' values (shift normality, alignment, and completeness) are multiplied by *-1* during objective value calculation to convert them into minimization objectives. For the first time, the objectives of the chemical shift assignment problem are handled separately in this thesis. Before these objectives were combined and one score function was used to find the optimum assignment. We also applied the standard genetic algorithm to the problem, where we combined all of these objectives in one score function.

The problem model implicitly covers the constraints. As per our definition for a solution candidate, one expected peak could be assigned to a maximum of one measured peak, which covers the first constraint. Similarly, our method allows mapping an expected peak only to a measured peak of the same spectrum, which

covers the second constraint. Additionally, the third constraint is covered by the alignment objective. The third constraint requires that the variance of the chemical shift assignment of an atom cannot exceed its given tolerance. The alignment objective seeks for the assignment where peaks assigned to the same atom are aligned.

3.4 Score Function

The score function of an algorithm plays an essential role in the success of the overall method. However, it is not trivial to define and find the optimal correspondence between expected and observed peaks in a set of NMR spectra for a protein. There might be some missing measured peaks in the spectrum, so there can exist some expected peaks, for which a corresponding measured peak does not exist at all in the spectrum. Additionally, there can be artifacts in the spectrum so some of the measured peaks do not correspond to any expected peak. Furthermore, it is difficult to predict the expected peaks for some NMR experiments theoretically. Therefore, the notion of "optimal correspondence" should be expressed in a scoring function that is defined in a robust way such that it tolerates a significant amount of artifacts and missing observed peaks. For instance, it is well possible in practice that only about 50% of the expected peaks can be observed and that the list of measured peaks contains as many artifact peaks as real ones. If defined properly, the maximal score value corresponds to the best, i.e., most correct, resonance assignment.

In order to define the scoring function of the problem, the characteristics of a correct assignment are defined as the objective values of the problem: shift normality, alignment, completeness, and low degeneracy. Shift normality aims to keep the chemical shift assignments of the atoms as close as their statistics values from chemical shift libraries. The alignment objective keeps that the entire chemical shift assignments for a single atom in the example are aligned with each other as much as possible. The completeness objective's goal is to assign as many peaks as possible. The last objective is the low degeneracy, which means that the number of degenerate peaks is small. Degenerate peaks are the expected peaks that are assigned to the same observed peaks.

We used the global score function of FLYA that is given in equation 3.1 [16]. This function assigns one single global score value for an assignment by evaluating four objectives of the assignment. The numerator goes through all of the assigned atoms

and sums up the score coming from every single assigned atom, and the denominator is used to normalize the *G* values.

$$G = \frac{\sum_{a \in A} [W_1(a)Q_1(a) + \sum_{n \in N'a} W_2(a,n)Q_2(a,n)/b(n)]}{\sum_{a \in A_0} [W_1(a) + \sum_{n \in N_a} W_2(a,n)]} \quad (3.1)$$

A_0 in the denominator of equation 3.1 refers to the set of all atoms of the expected peak list. N_a represents the set of expected peaks of atom *a*. The numerator of this equation calculates the assignment's score by evaluating the objective value results of the assigned atoms. *A* in the numerator is a subset of A_0 and it denotes the set of atoms of the assigned expected peak list. The global score, *G*, of the assignment is calculated by summing up the contribution coming from each assigned atom, *a*, in the numerator. N'_a shows the set of expected peaks of the atom *a* that is assigned to a measured peak. The value *b(n)* is equal to the number of expected peaks that are mapped to the same measured peak as the expected peak, so it is the ambiguity of the assignment. $Q_1(a)$ denotes the shift normality, meaning the alignment of the assigned chemical shift values of the atoms to their general chemical shift statistics. Equivalently, $Q_2(a,n)$ calculates the alignment of the chemical frequency of atom *a*, to the other chemical frequency values for the same atom defined in other mapped expected peaks of the spectrum. An aggregating function is used in this equation to produce a single global score value out of these different objectives. The weight values $W_1(a)$ and $W_2(a,n)$ are used to combine the objective value scores. In the calculations of FLYA, these values are used as $W_1(a)= 4$ and $W_2(a,n)=1$.

The four objectives of the problem are covered by this score function. We separated this equation into four different score functions in our algorithm. Each of these score functions represents one objective of the problem: Our first objective is the degeneracy of the atoms, which is represented by *b(n)*. Our second objective is the shift normality value, which is represented by $Q_1(a)$. Our third objective is the score the chemical shift references from different peaks, which is shown by $Q_2(a,n)$. The fourth objective is the completeness objective that is the number of assigned atoms, which is implicitly covered by the FLYA global score function. In our model, we added a fourth objective to explicitly evaluate the total number of assigned atoms, *d(n)*, in an assignment. We

tried to minimize the degeneracy objective *b(n)* and tried to maximize the other objectives, shift normality $Q_1(a)$, alignment $Q_2(a,n)$, and completeness *d(n)*.

3.5 Problem Representation

We modelled the chemical shift assignment problem as a combinatorial optimization problem, where the expected peaks are mapped to the measured peaks.

3.5.1 Representation on multiple spectra

We worked with more than one spectrum in our experiments. A chemical shift problem representation of more than one spectrum is shown in Figure 3.3. Each spectrum has its own expected peaks and measured peaks, so each spectrum is handled separately. However, the atoms in the experiment are the same. As a result, different spectra can assign different chemical shift frequency values for the same atoms of the experiment.

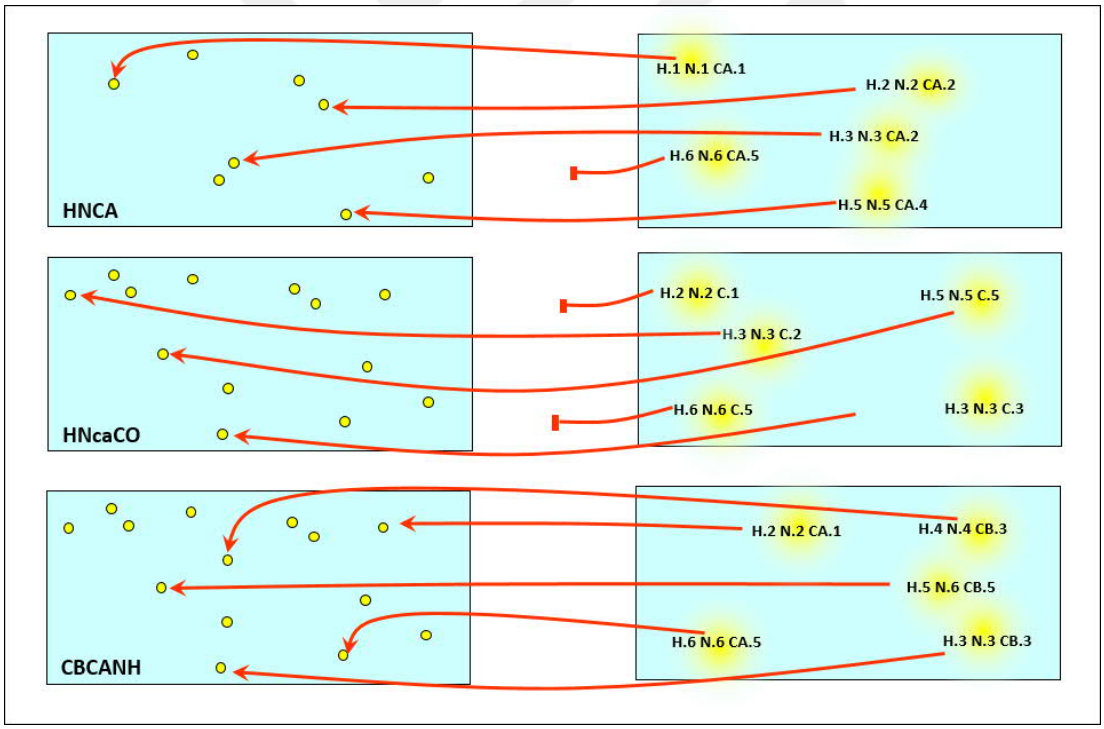

Figure 3.3 : The assignment in three spectra, HNCA, HNcaCO, and CBCANH.

This behavior is explained in Figure 3.4, which shows the overall structure of the atoms and the peaks of an NMR experiment before the assignment is done. The upper part of the figure shows the residue chain of the protein. The atoms that build each residue are also shown in the figure. There are three spectra in this experiment and expected peaks for each spectrum are marked: The atoms that build the expected peaks of the HNCA

spectrum are marked with red, the ones of the C13HSQC spectrum are marked with green and the atoms of the expected peaks of the CcoNH spectrum are marked with grey on the figure. As seen in the figure, several atoms of the protein give rise to many different expected peaks that even belong to different spectra.

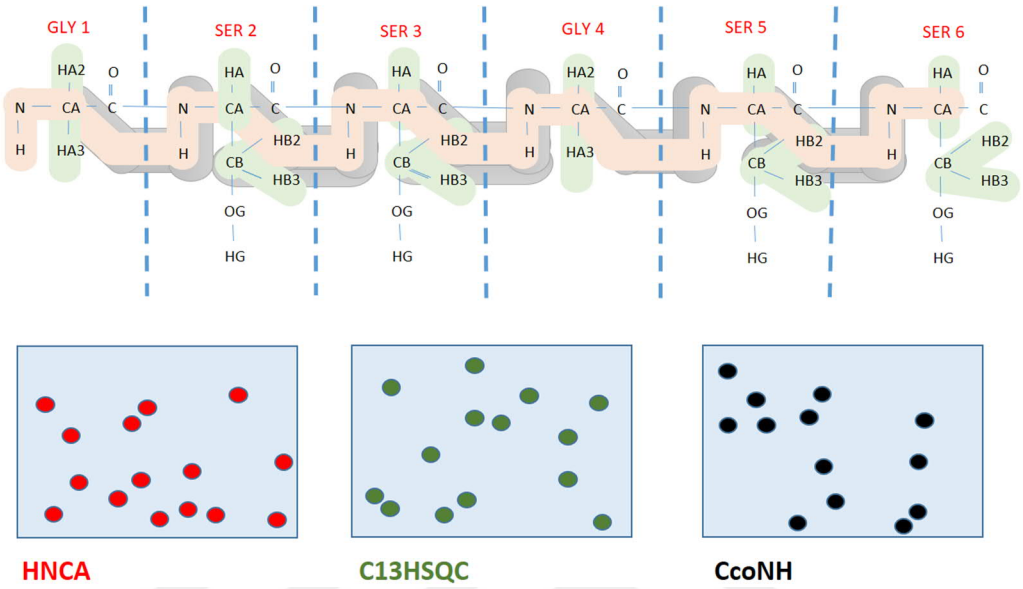

Figure 3.4 : Multiple spectrum atoms, expected peaks and measured peaks before the assignment is done.

The representation of the problem model becomes as shown in Figure 3.5 after the assignment is done. The expected peaks are shown on the upper side of the figure, where we know precisely which atoms are expected to give peaks on each spectrum. However, their location is only approximately available. Similarly, we have the measured peaks of each spectrum, which are shown at the bottom of the figure.

On the lower side of Figure 3.5, the specific locations of these peaks on the spectrum are known. However, the atoms giving rise to these peaks are unknown. We can easily assign the chemical shift values to the atoms once the assignment is completed because we know precisely which atoms belong to which expected peaks and the exact chemical shift values for a measured peak. As shown in the figure the expected peaks are assigned to the measured peaks of the same spectrum only. Some of the atoms are assigned to more than one measured peak, even in different spectra. As a result, these atoms will get multiple chemical shift frequencies coming from different measured peaks of different spectra. The average of all assigned frequencies is the final chemical shift frequency of an atom.

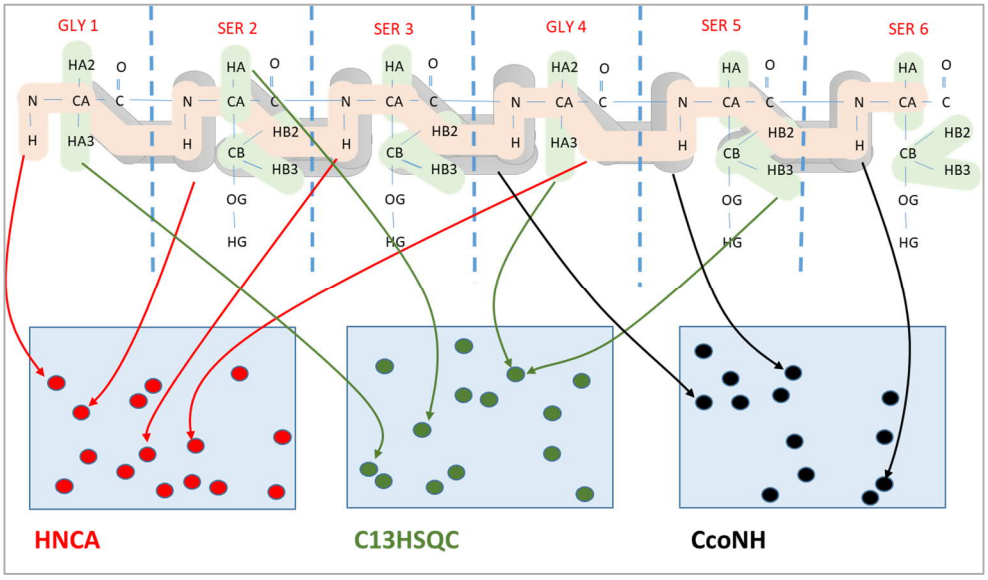

Figure 3.5 : The chemical shift assignment problem representation of the expected peaks onto the measured peaks on three different spectra.

3.5.2 Chromosome structure

Evolutionary algorithms work with a population that is composed of several solution candidates. In our study, each solution candidate represents the expected peaks of the experiment, which is the chromosome of the algorithm. The values in the chromosome indicate the measured peaks that are assigned to that specific expected peak. Figure 3.6 shows the chromosome structure where there is only one spectrum in the experiment. The solution candidate represents a chromosome of our problem model, where each gene shows the measured peak that is assigned to that single expected peak. If the value is -1, then that expected peak is not assigned to any measured peak.

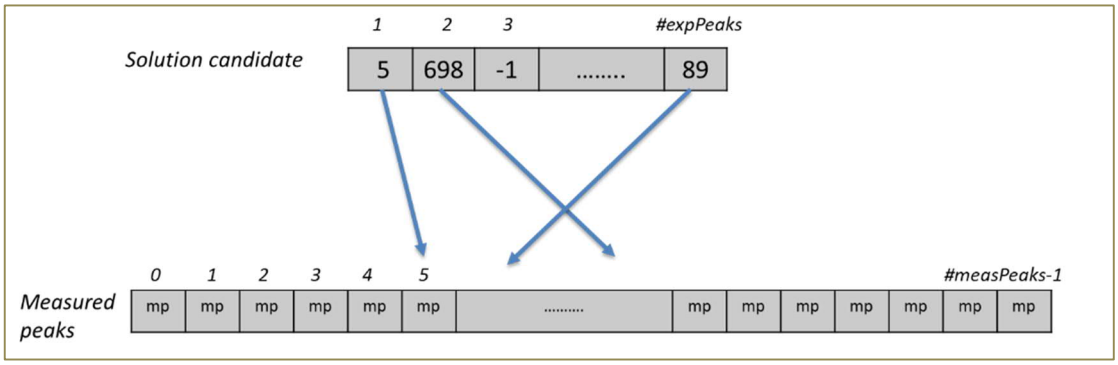

Figure 3.6 : Solution candidate shows the chromosome structure of one spectrum.

Figure 3.6 shows the representation of a chromosome with only one spectrum. If there is more than one spectrum, then the solution candidate is expanded for the expected peaks of the other spectra. An example of our representation for three spectra is given

in Figure 3.7. All spectra assignments are added after each other, and one chromosome is created.

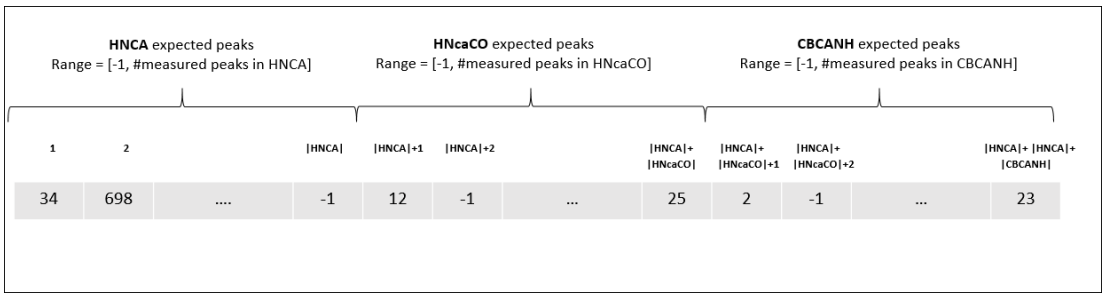

Figure 3.7 : The chromosome of the evolutionary algorithm for three spectra.

Another example with fifteen spectra is given in Figure 3.8. As seen in the figure, each expected peak of each spectrum is added one after another to create the solution candidate, which is the chromosome of our evolutionary algorithm. Each expected peak of a spectrum can be assigned only to the measured peaks of the same spectrum. For this reason, each expected peak from each spectrum is represented by one gene on the chromosome of a solution candidate. Therefore, it is not possible to assign an expected peak to more than one measured peak. This is the first constraint of our problem model. As a result, the first constraint of our problem model is fulfilled by the chromosome representation automatically. Similarly, since the assignment is done only on the same spectrum of expected and measured peaks, the second constraint is also covered naturally by the problem model.

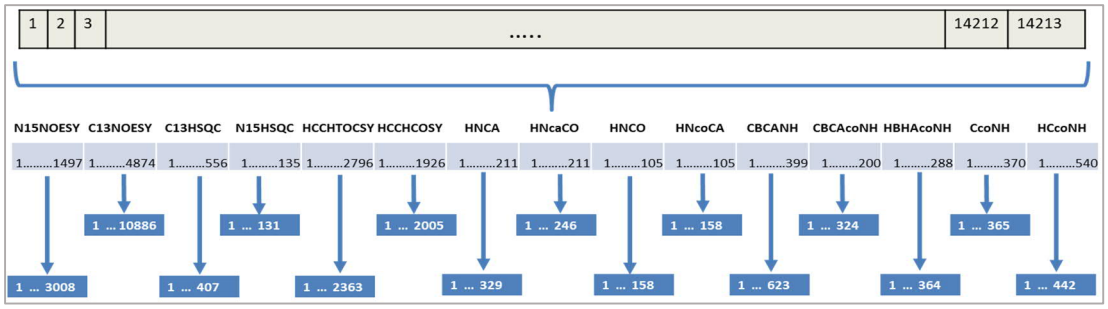

Figure 3.8 : Chemical shift assignment representation on fifteen spectra.

3.5.3 Chemical shift frequency calculation

The chromosome of the algorithm represents the expected peaks of experiments. One assignment maps one expected peak to a measured peak of that spectrum. However, the objective values are calculated from the chemical shift values of each atom instead of the expected peak mappings. So, a connection from the expected peak assignment

to chemical shift values of atoms was established by using the fact that each expected peak is composed of several atoms depending on the spectrum definition. An atom representation is built and all expected peaks from different spectra are assigned to the atoms, as shown in Figure 3.9. This representation is beneficial, especially for dynamically expanding the assigned peaks of each atom and eventually calculating their final chemical shift frequency, which is the average of all assigned chemical shift values.

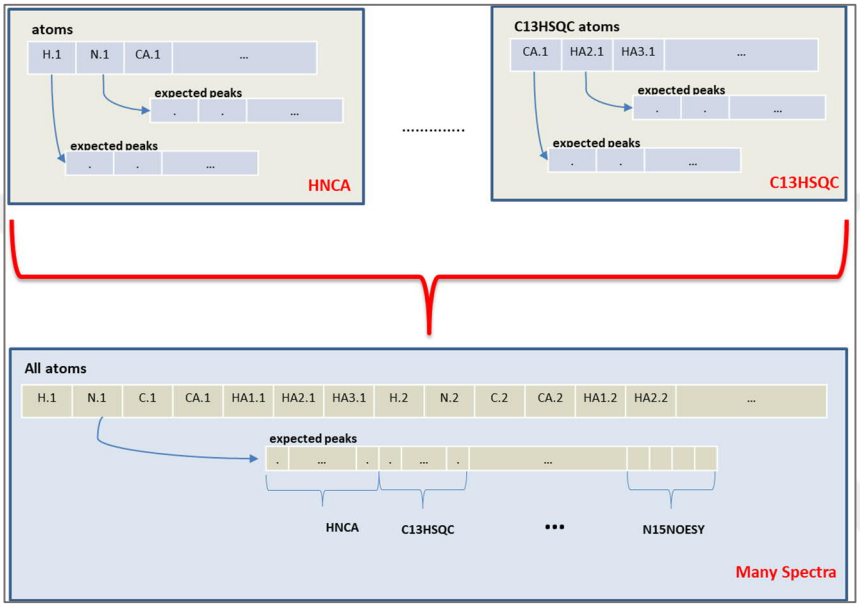

Figure 3.9 : Atoms of an expected peak. The above part shows spectrum-specific atoms, atoms along with their expected peaks are shown at the bottom.

3.6 Input and Output Files

We implemented our program as a standalone program that receives the necessary information via input files and delivers its results via output files.

3.6.1 Input files

The input files of our program are defined in the subchapters below.

3.6.1.1 Sequence file

One of the input files of our algorithm is the *sequence file*. The sequence of the amino acids that bond to each other to create the protein of the experiment is provided in the d*emo.seq* file. Here each amino acid is represented by its three-letter code. An example of this file, where only six amino acids are displayed, is given in Figure 3.10.

3.6.1.2 CYANA library file

Another input file of our method is the CYANA *library file*, *cyana.lib*. This file contains the atoms, their connectivity, and bonds for each residue, along with the connectivity patterns for each spectrum that defines the pattern of the atoms that are expected to give rise to a peak in the NMR experiment. An abstract from the CYANA library file is given in Figure 3.11. The CYANA library file contains four blocks: *ATOMTYPES*, *RESIDUE*, *CSTABLE*, and *SPECTRUM*.

Figure 3.10 : Input file for the sequence of amino acids of the protein, *demo.seq*.

The first block *ATOMTYPES* declares all of the atoms that are used in the residue library. This block starts with the header, including the *ATOMTYPES* keyword and the total number of atoms of this declaration. The following lines define each atom in detail. This order value is followed by the atom type name, repulsive core radius of these types of atoms, a value for hydrogen bond capabilities of the atom and the order number of the chemical element.

Then the second block *RESIDUE* defines each residue of the library one after another. The first one is *ALA* here; afterward, the remaining residues are explained. Each residue description block starts with the *RESIDUE* and three-letter name of the residue followed by the total number of the rotatable dihedral angles of this residue and the total number of the atoms in the residue. The last two numbers show the first and last atom indices on the atom definitions which belong to the residue. The first and last atom indices are needed because they can be different from the first and last atoms of the atom declarations of the residue. After all, the first atoms might be the ones from

the previous residue and similarly, the last atoms might be the ones from the next residue. This line of the residue block is followed by two sub-blocks: the dihedral angles and the atom definitions of the residue. The lines for the dihedral angles show the name of the dihedral angle. These four atoms define the dihedral angle and the number of the last atom that a rotation of this dihedral angle will impact. For the backbone dihedral angles, this number is set to 0. Then the atom descriptions of the residue are given. The first number is the atom order in the residue that is followed by the atom name and its type. Then the three-dimensional reference coordinates of the atom in the residue are given. Then the atom indices of the covalent bond are given.

The third block is the *CSTABLE* that provides the general statistics chemical shift frequency values of the atoms in each residue. These values will be used to calculate the objective values for each atom.

The last block of the CYANA library file is the spectrum definitions. Only the HNCA spectrum is displayed as an example in this figure. These spectrum definitions are used to create the expected peaks. The first line in the file, starting with `SPECTRUM`, specifies the name of the spectrum, `HNCA` here, and the label of the atoms that exist in each dimension of the spectrum. HNCA spectrum is used in the baseline example, which is three-dimensional, composed of the `'HN'`, `'N:'`, and `'C'` atoms. The lines below show the communication patterns that give rise to a peak in this spectrum. In the HNCA spectrum, there are two communication patterns, as seen in the figure. The pattern description lines start with the probability value, which is followed by the atom labels that give rise to that peak. For example, the first HNCA spectrum connectivity pattern in the figure below starts with `0.980`, which is the probability that this peak will occur. Then the atoms giving rise to this peak are defined as `HN:H_AMI N:N_AM* C:C_ALI C_BYL.`

There are connectivity patterns for each spectrum that defines the pattern of atoms that might give rise to a peak on the spectrum with some probability. Atom combinations that match with these predefined patterns are expected to give rise to peaks on the spectrum, so they are called *expected peaks*. The CYANA program already calculates these expected peaks so perfectly, that the expected peaks are taken directly from CYANA into our algorithm.

```
ATOMTYPES    20       0.10
   1 PSEUD    -10.00     0     0     0.0000
   2 H_ALI      1.00     0     1     1.0080
   3 H_AMI      0.95     1     1     1.0080
   4 H_ARO      1.00     0     1     1.0080
   5 H_SUL      1.00     1     1     1.0080
   6 H_OXY      1.00     1     1     1.0080
   7 C_ALI      1.60     0     6    12.0100
   8 C_BYL      1.50     0     6    12.0100
   9 C_ARO      1.60     0     6    12.0100
  10 C_VIN      1.60     0     6    12.0100
  11 N_AMI      1.45     0     7    14.0100
  12 N_AMO      1.50    -1     7    14.0100
  13 O_BYL      1.30    -1     8    16.0000
  14 O_HYD      1.30    -1     8    16.0000
  15 O_EST      1.30    -1     8    16.0000
  16 S_OXY      1.80     0    16    32.0600
  17 S_RED      1.80     0    16    32.0600
  18 P_ALI      1.80     0    15    30.9700
  19 METAL      1.80     0    50    60.0000
  20 DUMMY    -10.00     0   999     1.0000

RESIDUE   ALA      4   14    3   13
 1 OMEGA    0    0    0.0000    2    1    3    4    0
 2 PHI      0    0    0.0000    1    3    5   12    0
 3 CHI1     0    0    0.0000    3    5    8    9   11
 4 PSI      0    0    0.0000    3    5   12   14    0
 1 C   C_BYL    0    0.0000    0.0000    0.0000    0.0000  2   3   0    0  0
 2 O   O_BYL    0    0.0000   -0.6700    0.0000   -1.0322  1   0   0    0  0
 3 N   N_AMI    0    0.0000    1.3290    0.0000    0.0000  1   4   5    0  0
 4 H   H_AMI    0    0.0000    1.8069   -0.0007    0.8553  3   0   0    0  0
 5 CA  C_ALI    0    0.0000    2.0929   -0.0011   -1.2414  3   6   8   12  0
 6 HA  H_ALI    0    0.0000    2.7479   -0.8605   -1.2295  5   0   0    0  0
 7 QB  PSEUD    0    0.0000    0.9385   -0.1668   -2.7179  0   0   0    0  0
 8 CB  C_ALI    0    0.0000    1.1594   -0.1351   -2.4354  5   9  10   11  0
 9 HB1 H_ALI    0    0.0000    0.1776    0.2261   -2.1671  8   0   0    0  7
10 HB2 H_ALI    0    0.0000    1.5440    0.4465   -3.2601  8   0   0    0  7
11 HB3 H_ALI    0    0.0000    1.0940   -1.1731   -2.7265  8   0   0    0  7
12 C   C_BYL    0    0.0000    2.9378    1.2621   -1.3672  5  13  14    0  0
13 O   O_BYL    0    0.0000    2.4397    2.3166   -1.7634 12   0   0    0  0
14 N   N_AMI    0    0.0000    4.2169    1.1488   -1.0273 12   0   0    0  0
...

CSTABLE     410
   1 ALA   N       21679  123.24    3.56   77.10  142.81
   2 ALA   H       23992    8.20    0.60    3.53   11.48
   3 ALA   CA      19703   53.16    1.97   44.22   65.52
   4 ALA   HA      18269    4.26    0.44    1.24    6.51
   5 ALA   QB      17243    1.36    0.25   -0.83    3.12
   6 ALA   CB      18409   18.99    1.81    9.79   28.40
   7 ALA   C       14308  177.77    2.15  164.48  187.20
   8 ARG   N       14311  120.81    3.71  102.94  137.60
   9 ARG   H       16408    8.24    0.61    3.64   12.69
  10 ARG   CA      13154   56.78    2.31   43.27   67.98
...
 408 URA   H5        331    5.45    0.28    4.82    6.29
 409 URA   C6        214  141.02    1.99  134.31  145.70
 410 URA   H6        328    7.76    0.18    7.12    8.46
...

SPECTRUM HNCA HN N C
 0.980 HN:H_AMI N:N_AM* C:C_ALI C_BYL
 0.800 HN:H_AMI N:N_AMI (C_ALI) C_BYL C:C_ALI
...
```

Figure 3.11 : CYANA library file, *cyana.lib***.**

3.6.1.3 Expected peaks file

An example from the HNCA spectrum is given in Figure 3.12. First, the file gives the information about the dimension of the spectrum that shows how many atoms give rise to a peak on that spectrum. Then the names of the atoms are presented in each line. Afterwards, each expected peak of this spectrum is given in the next lines. Each expected peak line starts with the index of that expected peak on this spectrum. The last three values for each line show the atoms that give rise to a peak on that spectrum. The atom names and residue numbers are merged with the "." dot together for defining unique atom names. The additional data between the expected peak index and the atom names are not used for the expected peak values.

```
# Number of dimensions 3
#FORMAT xeasy3D
#INAME 1 HN
#INAME 2 N
#INAME 3 C
#SPECTRUM HNCA HN N C
 1    8.257  115.146   62.251 1 U   9.800E-01  0.000E+00 e 0 H.1    N.1   CA.1
 2    8.317  122.049   62.251 1 U   8.000E-01  0.000E+00 e 0 H.2    N.2   CA.1
 3    8.317  122.049   57.962 1 U   9.800E-01  0.000E+00 e 0 H.2    N.2   CA.2
 4    7.876  117.975   57.962 1 U   8.000E-01  0.000E+00 e 0 H.3    N.3   CA.2
 5    7.876  117.975   63.654 1 U   9.800E-01  0.000E+00 e 0 H.3    N.3   CA.3
 6    7.675  117.122   63.654 1 U   8.000E-01  0.000E+00 e 0 H.4    N.4   CA.3
 7    7.675  117.122   55.611 1 U   9.800E-01  0.000E+00 e 0 H.4    N.4   CA.4
 8    7.512  120.777   55.611 1 U   8.000E-01  0.000E+00 e 0 H.5    N.5   CA.4
 9    7.512  120.777   45.595 1 U   9.800E-01  0.000E+00 e 0 H.5    N.5   CA.5
10    7.626  118.524   45.595 1 U   8.000E-01  0.000E+00 e 0 H.6    N.6   CA.5
11    7.626  118.524   55.742 1 U   9.800E-01  0.000E+00 e 0 H.6    N.6   CA.6
12    8.311  119.788   55.742 1 U   8.000E-01  0.000E+00 e 0 H.7    N.7   CA.6
13    8.311  119.788   56.031 1 U   9.800E-01  0.000E+00 e 0 H.7    N.7   CA.7
14    7.341  121.030   56.031 1 U   8.000E-01  0.000E+00 e 0 H.8    N.8   CA.7
...
53    9.103  124.952   55.539 1 U   9.800E-01  0.000E+00 e 0 H.28   N.28  CA.28
54    8.459  109.463   55.539 1 U   8.000E-01  0.000E+00 e 0 H.29   N.29  CA.28
55    8.459  109.463   54.603 1 U   9.800E-01  0.000E+00 e 0 H.29   N.29  CA.29
56    7.412  117.998   54.603 1 U   8.000E-01  0.000E+00 e 0 H.30   N.30  CA.29
57    7.412  117.998   53.111 1 U   9.800E-01  0.000E+00 e 0 H.30   N.30  CA.30
```

Figure 3.12 : Example expected peaks for the HNCA spectrum, *HNCA_exp. peaks*.

The expected peaks are created in CYANA by using the information from the CYANA library file, *cyana.lib*, and the amino acids of the protein, *demo.seq*. Depending on the amino acid structure descriptions from the CYANA library file and the amino acids and their sequences from the sequence file, the primary structure of the whole protein is created. There are already patterns defined in the CYANA library file for the expected peaks for each spectrum. The HNCA spectrum's expected peak patterns are given in Figure 3.13. The first column shows the probability of this expected peak to occur that is followed by the atoms that will give to that expected peak. Afterwards, the connectivity patterns of the respective spectrum are searched in this structure. The atoms that match the patterns of the respective spectrum are defined to be expected

atoms of the experiment, in other words, these atoms are expected to give rise to a peak on that spectrum. The results of this calculation are saved into the expected peaks file of CYANA.

```
SPECTRUM HNCA HN N C
0.980 HN:H_AMI N:N_AM* C:C_ALI C_BYL
0.800 HN:H_AMI N:N_AMI (C_ALI) C_BYL C:C_ALI
```

Figure 3.13 : Expected peak patterns in the HNCA spectrum.

An example of the expected peak creation process for the HNCA spectrum is given in Figure 3.14. Firstly, the residues from the input sequence file, *demo.seq*, are chained to each in the same order in the sequence file. Their chemical structure is taken from the *cyana.lib* file. As shown in Figure 3.13, there are two connectivity patterns for the HNCA spectrum that are expected to give rise to peaks. Depending on these patterns, the expected peaks are defined on the structure and the results are saved in the expected peaks file. The resulting expected peaks are found as shown in Figure 3.14. Here the blue expected peaks have the probability of 0.98 and the orange ones have the probability of 0.80.

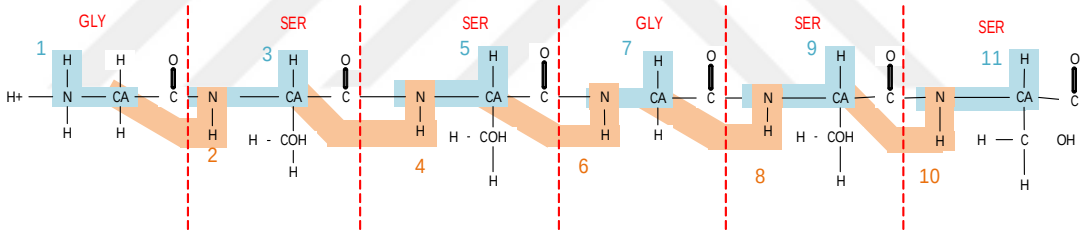

Figure 3.14 : Expected peaks example for HNCA spectrum.

3.6.1.4 Measured peaks file

Our algorithm needs as input the *measured peaks* file for each spectrum that includes the type of the spectrum and the positions of the peaks occurring on this specific spectrum. An example of measured peaks file from the HNCA spectrum is given in Figure 3.15. This file is composed of 698 measured peaks. The first line shows the number of dimensions of the spectrum, the second line shows the format of the file, whereas all of the measured peak lists are in the XEASY3D format. Then the atoms that give rise to a peak in this spectrum are listed: HN, C and N in this file, which is followed by the spectrum name: HNCA in this file. Afterwards, the measured peaks are listed. Not the whole list of the measured peaks is given here to shorten the file. The first number in one measured peak line shows the identification number of that

peak. The next values are the chemical shift in ppm unit for HN, C and N atoms respectively. The other values show the peak volume and intensity values of the peaks that are not used by our algorithm.

```
# Number of dimensions 3
#FORMAT xeasy3D
#INAME 1 HN
#INAME 2 C
#INAME 3 N
#SPECTRUM HNCA HN C N
     1     8.710   62.103  129.036 1 U   3.171E+00  0.000E+00 e 0 0 0 0
     2     8.710   62.103  129.036 1 U   3.171E+00  0.000E+00 e 0 0 0 0
     3     8.625   58.421  129.024 1 U   3.789E+00  0.000E+00 e 0 0 0 0
     4     8.625   54.452  129.028 1 U   3.305E+00  0.000E+00 e 0 0 0 0
     5     8.593   62.098  128.230 1 U   3.866E+00  0.000E+00 e 0 0 0 0
     6     8.593   55.616  128.225 1 U   3.336E+00  0.000E+00 e 0 0 0 0
     7     8.525   57.827  128.160 1 U   3.467E+00  0.000E+00 e 0 0 0 0
     8     8.525   54.436  128.153 1 U   3.900E+00  0.000E+00 e 0 0 0 0
     9     8.820   54.389  127.201 1 U   3.741E+00  0.000E+00 e 0 0 0 0
     10    8.820   45.490  127.234 1 U   5.086E+00  0.000E+00 e 0 0 0 0
     11    8.220   55.912  127.161 1 U   3.511E+00  0.000E+00 e 0 0 0 0
     12    8.219   54.807  127.160 1 U   3.148E+00  0.000E+00 e 0 0 0 0
...
    691    8.564   55.789  116.565 1 U   2.868E+00  0.000E+00 e 0 0 0 0
    692    8.564   55.789  116.565 1 U   2.868E+00  0.000E+00 e 0 0 0 0
    693    9.393   54.665  116.003 1 U   3.434E+00  0.000E+00 e 0 0 0 0
    694    8.260   62.078  115.151 1 U   3.448E+00  0.000E+00 e 0 0 0 0
    695    8.462   55.593  109.484 1 U   3.757E+00  0.000E+00 e 0 0 0 0
    696    8.461   54.599  109.480 1 U   3.161E+00  0.000E+00 e 0 0 0 0
    697    8.571   55.056  102.105 1 U   3.873E+00  0.000E+00 e 0 0 0 0
    698    8.572   45.390  102.105 1 U   3.201E+00  0.000E+00 e 0 0 0 0
```

Figure 3.15 : Input file measured peaks example for HNCA spectrum, *HNCA.peaks*.

3.6.2 Output files

The output files of our program are defined in the subchapters below.

3.6.2.1 Chemical shift frequency values file

The output of our algorithm is the chemical shift values assigned to atoms of the experiment. These assignment results are written into the output file with *prot* extension. An example chemical shift frequency values file is given in Figure 3.16. Each line shows the chemical shift values for one atom. The first number of one line shows the identical number of the atom. The second column is the final chemical shift value that is assigned to that atom. The atom name and residue number of that atom are the last two columns. The residue number is the order of that residue in the sequence input file. This file is the only output of our program and it shows the chemical shift values that are assigned to the atoms of the protein.

As already explained in the problem representation subchapter, there are many different chemical shift frequency values assigned by different spectrum or even in the same spectrum but from different expected peaks to the same atoms. For that reason, at the end of our program, we calculated the average value of all of these assignments to assign the eventual final chemical shift frequency for the atoms of the experiment. This file shows these final values that are the average of all relevant assignments for that atom of our program.

The CYANA framework has further methods, such as the consolidation part, to compare these chemical shift assignment results against their correct values and to make an analysis about the success of the assignment. The output files that are explained in the next subchapters are the output files of this consolidation script of CYANA.

```
1           55.712      0.000       CA          1
2           4.459       0.000       HA          1
3           30.106      0.000       CB          1
4           2.038       0.000       HB2         1
5           2.038       0.000       HB3         1
6           30.068      0.000       CG          1
7           2.37        0.000       HG2         1
8           2.03        0.000       HG3         1
9           43.357      0.000       CD          1
10          2.742       0.000       HD2         1
11          2.906       0.000       HD3         1
12          54.364      0.000       CA          2
13          4.05        0.000       HA          2
14          29.978      0.000       CB          2
15          1.98        0.000       HB2         2
16          1.995       0.000       HB3         2
...
358         176.761     0.000       C           12
359         175.372     0.000       C           13
360         176.292     0.000       C           14
361         176.302     0.000       C           15
362         175.307     0.000       C           16
363         173.719     0.000       C           17
364         175.124     0.000       C           18
365         176.925     0.000       C           19
366         175.702     0.000       C           20
367         174.763     0.000       C           21
368         174.286     0.000       C           22
369         176.044     0.000       C           23
370         176.925     0.000       C           24
371         174.251     0.000       C           25
372         173.969     0.000       C           26
373         178.753     0.000       C           27
374         174.433     0.000       C           28
375         175.805     0.000       C           29
376         175.327     0.000       C           30
```

Figure 3.16 : Output file chemical shift frequency values, *a01_5.prot*.

3.6.2.2 Consolidation file

CYANA framework has a consolidation script to compare the chemical shift assignment values against their correct values and provide an analysis of this comparison. The consolidation script receives the chemical shift values assigned to atoms, the chemical shift frequency values file, which is the only output of our program. The script also receives the correct chemical shift values for the atoms and compares the results against the real values.

The consolidation file is in the *.tab* extension. An example of the consolidation file is given in Figure 3.17. The first part of this file shows the chemical shift assignment results for each atom. If the assignment is self-consistent, meaning that the same assignment is done on at least 80% of the assignments, then the assignment is marked as strong. Otherwise, the assignment is marked as weak. The correct assignments are counted as "identical" and they are marked with the "=" symbol on the atom lines. If an assignment is wrong, then it is counted as "different" and marked with the "!" symbol on the atom lines.

```
Total number of shift values: 7440
Cutoff for extent           : 16.00

Atom  Residue      Ref   Shift     Dev  Extent  inside   inref
N     ARG    1 124.244 123.294   0.950    20.0    50.0     5.0  !
H     ARG    1   9.471   8.597   0.874    20.0    50.0     5.0  !
CA    ARG    1  53.259  60.002  -6.743    20.0    50.0     5.0  !
HA    ARG    1   5.108   3.474   1.634    20.0    50.0     5.0  !
CB    ARG    1  33.653  29.952   3.701    20.0    48.8     5.0  !
HB2   ARG    1   2.353   1.669   0.684    20.0    49.4     0.0  !
HB3   ARG    1   1.201   1.669  -0.468    20.0    49.2     0.0  !
CG    ARG    1  26.284  27.342  -1.058    20.0    73.5     5.0  !
HG2   ARG    1   1.564   1.561   0.003    20.0    35.3    35.0  =
HG3   ARG    1   1.308   1.583  -0.275    20.0    31.2    20.0  !
CD    ARG    1  43.914  43.424   0.490    20.0    80.1    40.0  strong!
HD2   ARG    1   2.465   2.904  -0.439    20.0    40.2     0.0  !
HD3   ARG    1   3.175   3.367  -0.192    20.0    44.9    20.0  !
...
HG    LEU   30   0.492   1.444  -0.952    20.0    30.5     0.0  !
QD1   LEU   30   1.393   0.730   0.663    20.0    27.9     0.0  !
QD2   LEU   30   0.627   0.724  -0.097    20.0    44.1     5.0  !
CD1   LEU   30  25.727  23.373   2.354    20.0    53.0     0.0  !
CD2   LEU   30  25.669  25.484   0.185    20.0    53.1    40.0  =
C     LEU   30 175.504 175.929  -0.425    20.0    90.3    15.0  strong!

Shifts            Consolidated   Total

Shifts            Strong   Total
Assigned       :     161     372
Identical      :     115     172
Different      :      45     191
Non-neighboring:      36     154
Additional     :       1       9
Reference only :               2
```

Figure 3.17 : The details of the chemical shift assignment, *consolidate.tab*.

At the end of the file, the sum of these values is given. The first line "*Assigned*" shows the strong and total number of atoms that are assigned. The identical line shows the correct atom counts. Similarly, the "*Different*" line shows the atom counts for the wrong assignments. The "*Reference only*" line shows the atom counts that have reference values but no assignment from the program.

3.6.2.3 Graphical representation file

CYANA consolidation script has a graphical representation of the results in the *.pdf* extension file. An example output file is given in Figure 3.18. Each assignment of the atoms are represented by the following colors:

Green: Assignment agrees with the real values.

Red: Assignment does not agree with the real values.

Blue: Assignment is done but there does not exist a real value.

Black: Assignment is not done but real value exists.

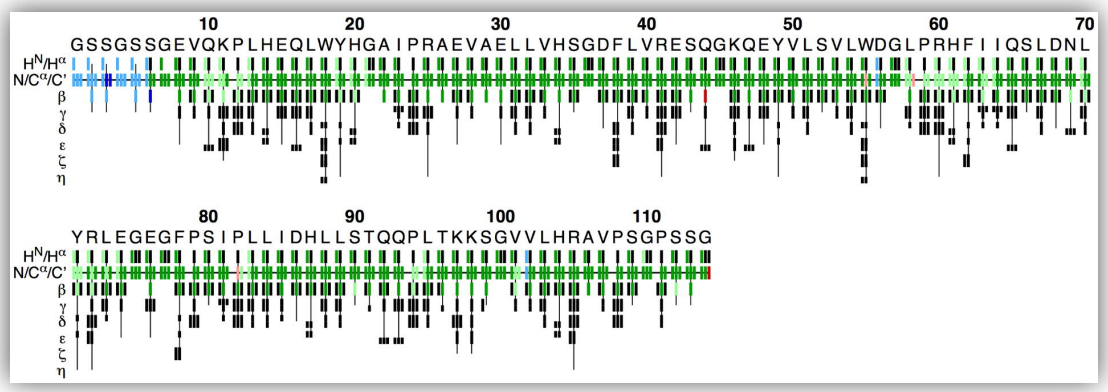

Figure 3.18 : Graphical representation of the assignment results, *flya.pdf*.

3.7 NMR Specific Behaviour

In this thesis, we used the most commonly used sixteen different NMR spectra: HNCA, N15NOESY, C13HSQC, C13HSQC_LK, C13NOESY, CBCAcoNH, CBCANH, CcoNH, CcoNH_LK, HBHAcoNH, HCCHCOSY, HCCHTOCSY, HCcoNH, HNcaCO, HNCO, and HNcoCA. Each spectrum has its own patterns for expected peaks; in other words, a different set of atoms contribute to the chemical shift assignment calculations in each spectrum. Additionally, there are some rules in the NMR experiments that were not defined anywhere but needed to be implemented to

make correct calculations. For that reason, a baseline of the NMR experiment is implemented and tested for each spectrum separately. In order to ensure that the calculations are correct, the G global value score is calculated for each assignment and they are compared against the ones in CYANA. In case of any difference, the differences are analyzed and the logic in our thesis is expanded with the new rules that are obtained on these tests.

3.7.1 Atom order

The expected peaks of each experiment are created by the CYANA framework. The order of the atoms is defined in each spectrum differently. The atom order of the peaks files might be different each time so the atom order logic is created dynamically and the order of the atoms is read from the `SPECTRUM` line of each peak file. Additionally, the order of the atoms in the expected peak list may differ from the one in the measured peak list of the same spectrum. For example, the atoms of the N15NOESY spectrum in the expected peak file are different from the ones in the measured peak file.

Figure 3.19 shows the measured peaks of the N15NOESY spectrum and Figure 3.20 shows the expected peaks of the same spectrum. The measured peak's atoms have the order `HN H N`, whereas the atoms in the expected peaks have the order `H HN N`. The order of the atoms are defined as the program starts and they have not been searched actively during the calculations to improve the performance.

```
# Number of dimensions 3
#FORMAT xeasy3D
#INAME 1 H
#INAME 2 HN
#INAME 3 N
#SPECTRUM N15NOESY H HN N:
 1    7.295    7.295  108.829 1 U   9.000E-01  0.000E+00 e 0     0     0     0
 2    3.440    7.295  108.829 1 U   9.000E-01  0.000E+00 e 0     0     0     0
 3    4.077    7.295  108.829 1 U   9.000E-01  0.000E+00 e 0     0     0     0
 4    7.424    7.295  108.829 1 U   7.000E-01  0.000E+00 e 0     0     0     0
 5    7.295    7.424  113.943 1 U   7.000E-01  0.000E+00 e 0     0     0     0
…
45    3.276    8.112  110.034 1 U   7.000E-01  0.000E+00 e 0     0     0     0
46    3.845    8.112  110.034 1 U   7.000E-01  0.000E+00 e 0     0     0     0
47    7.239    8.112  110.034 1 U   5.000E-01  0.000E+00 e 0     0     0     0
48    8.112    8.112  110.034 1 U   9.000E-01  0.000E+00 e 0     0     0     0
49    4.071    8.112  110.034 1 U   9.000E-01  0.000E+00 e 0     0     0     0
50    3.000    8.112  110.034 1 U   9.000E-01  0.000E+00 e 0     0     0     0
51    3.738    8.112  110.034 1 U   8.000E-01  0.000E+00 e 0     0     0     0
52    2.709    8.112  110.034 1 U   6.000E-01  0.000E+00 e 0     0     0     0
```

Figure 3.19 : N15NOESY.peaks measured peak file of N15NOESY spectrum.

```
# Number of dimensions 3
#FORMAT xeasy3D
#INAME 1 H
#INAME 2 HN
#INAME 3 N
#SPECTRUM N15NOESY H HN N:
 1    7.295    7.295  108.829 1 U   9.000E-01  0.000E+00 e 0 H.1       H.1  N.1
 2    3.440    7.295  108.829 1 U   9.000E-01  0.000E+00 e 0 HA2.1     H.1  N.1
 3    4.077    7.295  108.829 1 U   9.000E-01  0.000E+00 e 0 HA3.1     H.1  N.1
 4    7.424    7.295  108.829 1 U   7.000E-01  0.000E+00 e 0 H.2       H.1  N.1
 5    7.295    7.424  113.943 1 U   7.000E-01  0.000E+00 e 0 H.1       H.2  N.2
…
45    3.276    8.112  110.034 1 U   7.000E-01  0.000E+00 e 0 HB2.5     H.6  N.6
46    3.845    8.112  110.034 1 U   7.000E-01  0.000E+00 e 0 HB3.5     H.6  N.6
47    7.239    8.112  110.034 1 U   5.000E-01  0.000E+00 e 0 HG.5      H.6  N.6
48    8.112    8.112  110.034 1 U   9.000E-01  0.000E+00 e 0 H.6       H.6  N.6
49    4.071    8.112  110.034 1 U   9.000E-01  0.000E+00 e 0 HA.6      H.6  N.6
50    3.000    8.112  110.034 1 U   9.000E-01  0.000E+00 e 0 HB2.6     H.6  N.6
51    3.738    8.112  110.034 1 U   8.000E-01  0.000E+00 e 0 HB3.6     H.6  N.6
52    2.709    8.112  110.034 1 U   6.000E-01  0.000E+00 e 0 HG.6      H.6  N.6
```

Figure 3.20 : N15NOESY_exp.peaks expected peak file of N15NOESY spectrum.

3.7.2 Multiple occurrences of atoms

During the calculations, we assumed that each atom of a peak is different from the other, meaning one single atom can contribute only once to a single peak. However, it was observed in some spectra that one single atom had contributed twice to the same peak. For example, the first peak of the N15NOESY expected peaks, Figure 3.20, is composed of the atoms `H.1 H.1 N.1`. As a result, one atom contributed twice to the same peak. The solution is extended so that it can handle this behavior.

3.7.3 Shift reference values from pseudo atoms

There are shift reference values for each atom in the CSTABLE part of the CYANA library file, cyana.lib. An extract of this CYANA library file is given in Figure 3.11, and the third block of this file is the CSTABLE. Another extract from the CSTABLE is given in Figure 3.21. The column shown in the first rectangle shows the general chemical shift frequency statistics for that atom in that residue. The second one is the deviation in general from this value. Both values, the general statics, and the deviation values are used during the score calculations. These values are fetched from the library file before the program starts and they are used in the score function calculations.

During the score value comparisons, we realized that there are some differences in the reference chemical shift frequencies of some atoms that our algorithm has against the ones in CYANA. We have investigated these atoms and figured out that the atom HA2 has the reference from the QA atom from the CYANA library file.

```
CSTABLE      410
   1 ALA    N        21679   123.24    3.56   77.10  142.81
   2 ALA    H        23992     8.20    0.60    3.53   11.48
   3 ALA    CA       19703    53.16    1.97   44.22   65.52
   4 ALA    HA       18269     4.26    0.44    1.24    6.51
   5 ALA    QB       17243     1.36    0.25   -0.83    3.12
   6 ALA    CB       18409    18.99    1.81    9.79   28.40
   7 ALA    C        14308   177.77    2.15  164.48  187.20
   8 ARG    N        14311   120.81    3.71  102.94  137.60
   9 ARG    H        16408     8.24    0.61    3.64   12.69
  10 ARG    CA       13154    56.78    2.31   43.27   67.98
  11 ARG    HA       12795     4.30    0.46    1.34    6.10

...

 407 URA    C5         208   102.82    1.81   96.42  107.20
 408 URA    H5         331     5.45    0.28    4.82    6.29
 409 URA    C6         214   141.02    1.99  134.31  145.70
 410 URA    H6         328     7.76    0.18    7.12    8.46
```

Figure 3.21 : CSTABLE entries are showing the general chemical shift statistics and the deviation in rectangles.

Afterwards, we examined all of the atoms having differences from CYANA reference values in order to find out the rule. The QA atoms are pseudo atoms of the macromolecule, which means these atoms do not exist usually; they have been added artificially to the structure for simplifications in the calculation by representing more than one hydrogen atom and called pseudo atoms. After investigating the differences, the following logic of the reference values is figured out and implemented: If there is the same atom in the CSTABLE part, its reference value is taken to the calculations. For example, 123.24 and 3.56 values are taken as reference values, and the deviation for N atom in ALA residue due to the first line of the N in Figure 3.21. If the atom name does not exist in the CSTABLE, then a different logic is followed. The approach is summarized below, and its workflow is given in Figure 3.22.

If the atom name exists in the CSTABLE, take the reference shift value from the CSTABLE entry. For example, if HA123'' exists in the CSTABLE, then take its value from that line.

If it does not exist: Then replace the first character of the atom name by "Q" and if the new atom name exists in the CSTABLE, take the reference value of that atom. For example, If HA123'' does not exist in the CSTABLE, replace the first character by Q: The new atom name becomes QA123''. Then check whether this atom name exists in the CSTABLE.

If it does not exist: Then check the last character of the atom. If it is " then replace it by ' and check whether this new atom name exists in the CSTABLE. For example, HA123'' is changed into QA123'.

If it does not exist: Then remove the last numbers at the end of the atom name one by one and check whether the new atom name exists in the CSTABLE. For example, QA123' atom name is converted into QA123. If it does not exist, then QA12 is checked. In case QA12 does not exist in the CSTABLE, then QA1 is checked. With the first hit, its reference value from CSTABLE is assigned to the atom. If none of them exists, the atom will have no reference value during the score calculations.

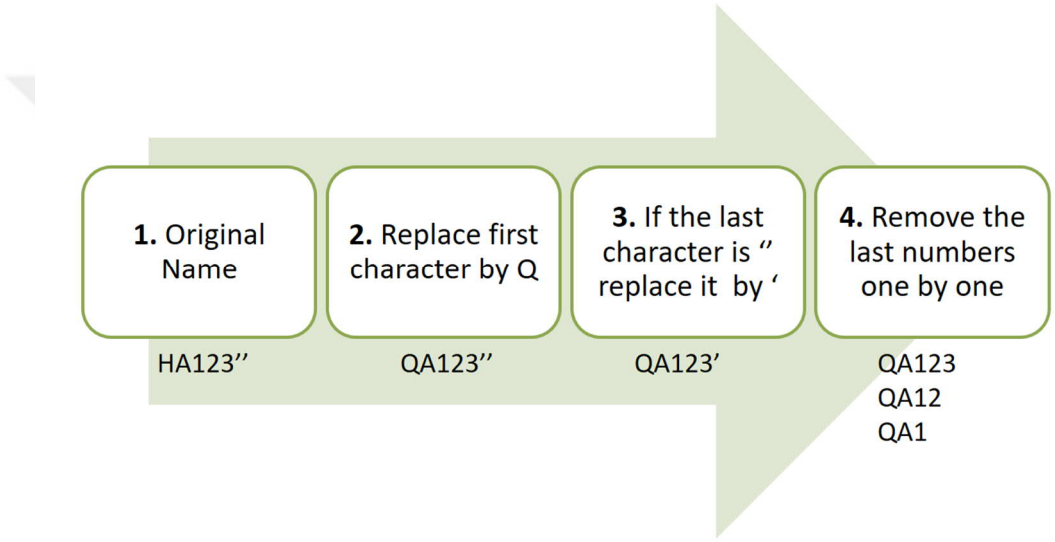

Figure 3.22 : Four steps exist in the process of finding shift reference values in the CYANA library.

3.8 Problem Challenges

3.8.1 NMR data challenges

Defining a model for the chemical shift assignment problem and finding the optimal mapping between expected and measured peaks in a set of NMR spectra for a macromolecule is not trivial. The main reason for this is that measured NMR spectra are not perfect: in general, they do not contain all peaks that one would expect to see; besides, they also contain noise and artifact peaks, which do not correspond to an expected peak. Furthermore, it is not easy to forecast the expected peaks for some of the NMR experiments. Therefore, the scoring function that can successfully model the notion of "optimal correspondence" between the expected peaks and measured peaks

and tolerate the absence of measured peaks and artifacts is crucial in the success of the NMR experiments. In practice, half of the expected peaks may not be observed on an NMR spectrum, whereas half of the observed peaks on that spectrum may belong to artifact peaks. So, it is imperative to define a proper score function where the optimum score corresponds to the best chemical shift assignment.

3.8.2 Search space challenges

Another big challenge in our problem is its vast search space. As per the definition of the chemical shift assignment problem, each expected peak can be mapped to each measured peak of the same spectrum. This behavior creates a huge number of possibilities and many solution candidates, even for small datasets. For example, Figure 3.23 represents a chemical shift assignment problem on a dataset with fifteen spectra. Here in the example, the first spectrum is N15NOESY with *1.497* expected peaks and *3.008* measured peaks. Only this spectrum creates 3008^{1497} solution candidates. The number of solution candidates from other spectra are calculated as shown in Figure 3.23. The whole search space is composed of the combinations of these solutions candidates, which is a vast number.

3.9 Improvements on the Algorithms

We tested our algorithms on different datasets and made several improvements to these algorithms accordingly. These improvements are explained in this subchapter.

3.9.1 Crossover and mutation rates

The crossover rate is the probability of applying a crossover operator to the individuals in an iteration of our evolutionary optimization methods. Similarly, the mutation rate is the probability of applying mutation in an iteration. In order to optimize these rates, we applied different rates, including static and dynamic values, for each parameter in our studies. We applied the following static crossover rates in our experiments: 0.001, 0.01, 0.1, and 1.0. Additionally, we created a dynamic crossover rate, 1/L, where L represents the length of the chromosome. After the experiments, we observed that a fixed crossover rate of 1.0 and mutation rate of 1/L produced the best results.

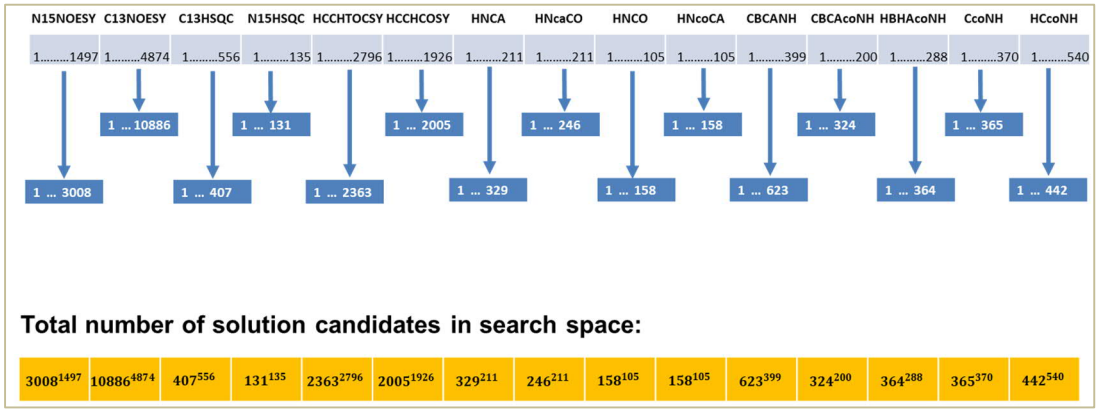

Figure 3.23 : Assignment on several spectra. The above part shows the number of the expected peaks and measured peaks for each spectrum. The part below shows the total number of solution candidates for the whole problem.

Similarly, we applied our algorithms with different mutation rates. Firstly, we applied dynamic mutation rate 1/L, where L represents the length of the individual. Then, fixed mutation rates are also applied to the algorithms, which are 0.001, 0.01, 0.1, and 1.0. In the end, we observed that the fixed mutation rate of 1.0 provided the best results in our experiments.

3.9.2 Population initialization

We produced the initial population of our algorithms by using a fitness function to start with a population that has better fitness than a randomly created one. We applied very similar logic to the local optimization and search space reduction of FLYA [16]. We reduced the search space during the initial population creation only. Afterwards, any offspring in the population is allowed during the evolutionary algorithm iterations, and the search space reduction logic is not applied anymore. We applied a constructive method while creating the initialization logic of the population. The pseudo-code of the initialization algorithm is presented in Algorithm 4. In the initialization algorithm, the search space in each dimension of expected peaks is created with the statistical frequency range for the atom of the expected peak on that dimension initially.

Additionally, the range values are updated so that the atom-specific tolerance values are added to the upper boundaries and are subtracted from the lower boundary values of the atoms. The tolerance values were defined as 0.03 ppm for ^{1}H, 0.4 ppm for ^{13}C, and 0.4 ppm for ^{15}N atoms. In the beginning, one expected peak among the non-assigned expected peaks is selected randomly. The measured peaks are checked one by one, and the first measured peak found in the restricted search area is assigned to

the expected peak. The chemical shift frequencies of the measured peak are added to the measured peak frequency lists of the adjacent atoms. The search spaces of these atoms are updated with the average values of their measured peak frequency lists and the tolerance values.

This procedure assures that the next expected peak to be mapped leads to a list entry that is consistent with the previous ones. This procedure goes on until all expected peaks are evaluated, and possible assignments are done. Once the assignments are completed, the individual is added to the initial population, and the following individual is created by following the same logic. The same logic is applied until all individuals of the population are created. In our experiments, we had fifty individuals in each population. This approach ensures that the completeness objective is fulfilled as much as possible.

Algorithm 4 Constructive Initialization approach

Input: *Expected peaks and measured peaks*
Output: *Initial population*
1: *Select randomly one expected peak to be assigned*
2: *Check possible measured peak candidates*
3: *Get the first fitting measured peak from the search space*
4: *Update search space with the new assignment*
5: **if** *there is still expected peak to be evaluated for assignment*
6: *Go back to step* #1 to evaluate a new expected peak
7: **end if**
8: *Add the individual to the initial population*
9: **if** *the population is not complete* **then**
10: *Go back to step* #1 *to create a new individual*
11: **end if**

3.9.3 Population diversity

Diversity is a crucial property in a population in order to escape from local optima. For that reason, we created some of these individuals randomly in different proportions, starting from 0%, by increasing 10% at each step, until the whole population is created randomly.

We compared the results and saw that the best results were achieved with the initial population, where all individuals are created by following the constructive logic explained above and none of them is created randomly.

3.9.4 Population size

Afterwards, the number of individuals in one population is evaluated. Several different individuals in one population are evaluated. Until now, fifty individuals were used in one population. Here, a population with hundred individuals is used. However, its result did not differ on the datasets from the original population's number, which was fifty. So still fifty individuals are used for further tests.

3.9.5 Repetition parameter

We run our algorithms several times after each other and consolidated the results all together to achieve the eventual results. We used a new parameter, the repetition parameter, in our algorithms. The repetition parameter defines how many individual executions of one algorithm are performed using different random numbers. The overall result of one run of the algorithm is composed of collecting the results of all repetitions, which is obtained by chemical shift consolidation [16]. With an increase in the repetition parameter, the results were improved however the runtime of the algorithm increased in the meantime. For that reason, the repetition parameter is set to 10 in many of our tests.

3.9.6 Hill-climbing algorithm

Hill climbing algorithms makes iterative improvement at each step by trying to find a neighbor solution candidate that improves the current solution. Therefore, they search for the local optimum in their neighborhood. In our method, we developed a hill-climbing algorithm [19] from scratch in the MOEA framework. Our hill-climbing algorithm tries to improve the current solution by checking the assignment value of each expected peak and by trying to find a better measured peak to be assigned for each expected peak in that neighborhood. In order to evaluate the fitness of the solution candidates, we used the global score function equation 3.1, which is the aggregation of all objectives.

3.9.7 Hybrid algorithms

We combined the hill-climbing algorithm with the NSGA2, NSGA3, and GA algorithms to determine the local optima. For this issue, a hybrid algorithm is implemented in the MOEA framework that combines the selected multi-objective algorithm with the hill-climbing algorithm. After the initial population is created, it

calls one iteration from the NSGA2, NSGA3, or GA algorithms, and afterward, it calls one iteration from the hill-climbing algorithm. We tested several combinations of these algorithms on several different datasets and evaluated their results. In the end, we saw that adding the hill-climbing algorithm did not improve our results. Therefore, this part is removed from the algorithm.

4. RESULTS AND DISCUSSIONS

We implemented our program as a standalone algorithm and evaluated its results by comparing them against two of the state-of-the-art programs of protein structure determination: FLYA [16] and PINE [50]. The FLYA program already exists as the default solution for the chemical shift assignment problem in the CYANA [73,74] framework. The program CYANA is a fully automatic solution to calculate the three-dimensional structure of proteins. CYANA includes all steps of NMR experiments, such as data processing, peak picking, chemical shift assignment, NOE assignment, and structure calculation. The program PINE makes an automatic assignment of the chemical shift values by using evidence estimates from empirical distributions and consistency measures [75]. We used the I-Pine webserver from http://i-pine.nmrfam.wisc.edu/. We performed our experiments with several different datasets and compared our results with the chemical shift frequency values that already have a whole structure determination prepared before by the conventional methods. We expected these chemical shift values to be correct and evaluated the success of our algorithms depending on this comparison. We used these manually assigned chemical shift values only for comparison. We run our tests on an NMR dataset of one large protein: the 114-residue Src homology 2 domain from the human feline sarcoma oncogene Fes (Fes SH2) [76-78]. We applied our algorithm, FLYA, and PINE to several different datasets obtained from the Fes SH2 protein. Furthermore, we performed our experiments on the fragments obtained from the Sso7d protein from the hyperthermophilic ar-chaeabacterium Sulfolobus solfataricus [79]. In this chapter, we present the results of these comparisons.

4.1 Experimental Design

In these experiments, FLYA is run with its default values, where 20 independent runs proceed. The results are the consolidation of these runs. PINE is run with its default values via its webserver. The algorithm's run time is evaluated as the time between the

input file submission and result arrival. All the remaining tests were run on the same personal computer on one core of an Intel Core i7-3720QM 2.60 GHz CPU.

Our algorithms are run five times independently, and their results are consolidated for the final chemical shift assignments. The results below show the mean values of these separate five runs for each algorithm. We have 50 individuals in each population of our algorithms. The constructive initialization approach is used in the initial population creation phase of our algorithms. The crossover rate and mutation rate are set to 1.0.

Additionally, the repetition parameter is set to 10 in our algorithms. The iteration count is set to 10.000. In some of the experiments below, we changed these default parameters for testing. Unless it is mentioned separately, these default values are used in the tests. We used the same peak lists such as expected peaks, measured peaks and sequence files, and spectra lists for all comparisons between the algorithms.

We already have the chemical shift values assigned by conventional methods for the datasets of these experiments. We assumed that these results are correct and compared our results against these values. These manually determined chemical shift assignment values were used only to evaluate the quality of the automated assignment.

The chemical shift assignments are grouped as *correct*, *wrong*, and *unassigned* relative to the chemical shift values from the reference lists obtained by conventional techniques. The *correct* chemical shift values show the number of atoms whose deviations from their reference chemical shift values are in their tolerance values, which are 0.03 ppm for ^{1}H, 0.4 ppm for ^{13}C, and 0.4 ppm for ^{15}N. If they do not lay on this range, then they are marked as *wrong*. The *unassigned* atom counts show the total number of atoms that were not assigned in those runs concerning the atoms in the reference lists. The atoms that do not exist in the reference list of the experiments were ignored.

4.2 Results on the Fes SH2 Protein

In these experiments, we ran our algorithms with their default parameters. Our algorithms are Non-dominated Sorting Genetic Algorithm II (NSGA2), Non-dominated Sorting Genetic Algorithm III (NSGA3), and standard Genetic Algorithm (GA). We performed these tests on the seven different fragments from the Fes SH2

protein [76-78]. We performed the same experiments also with the two state-of-the-art programs FLYA and PINE, with their default values for comparison.

4.2.1 Dataset details

There are 114 residues in the Src homology 2 domain from the human feline sarcoma oncogene Fes (Fes SH2) protein. We divided this protein into smaller fragments in order to evaluate the behavior of our algorithms in different datasets. As shown in Figure 4.1, we created seven different datasets by extracting thirty different amino acids from this example. We excluded the starting and ending amino acids from this dataset creation process since these amino acids include unstructured regions and expression tags.

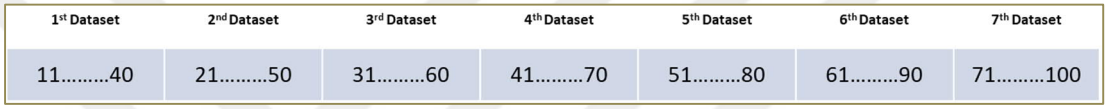

Figure 4.1 : The fragments from the 114-residue Src homology 2 domain from the human feline Fes SH2.

The details of these dataset fragments are given in Table 4.1. Since the whole protein is divided into the same-sized fragments, each dataset contains 30 residues.

Table 4.1 : Experimental dataset details.

Fragment (residues)	Expected peaks	Measured peaks	Residues	Atoms	Spectra
1 (11-40)	791	655	30	356	11
2 (21-50)	817	663	30	338	11
3 (31-60)	835	648	30	362	11
4 (41-70)	878	660	30	365	11
5 (51-80)	816	641	30	356	11
6 (61-90)	832	661	30	362	11
7 (71-100)	824	635	30	441	11

In Table 4.1, the numbers in the parenthesis of the *Fragment* column show the first and last residue number of each fragment. The *Expected peaks* column shows the number of expected peaks that are used in each fragment. These expected peaks are created by CYANA and given as input to our algorithms. The *Measured peaks* column shows the total number of measured peaks used in the experiments. The number of residues in each fragment is given in the *Residues* column. Since the whole dataset is divided into the same size, each fragment includes the same number of residues, 30. The last column, *Spectra*, shows the number of spectra in each fragment. We used the same

spectrum set in each experiment: C13HSQC, CBCANH, CBCAcoNH, CcoNH, HBHAcoNH, HCcoNH, HNCA, HNcaCO, HNCO, HNcoCA, and N15HSQC.

4.2.2 Empirical results

Table 4.2 presents a summary of these experiments on the data fragments of the Fes SH2 protein. The numbers in the table are the total number of correct, wrong, and unassigned atoms for each experiment. The numbers in the parenthesis show the percentages of the correct, wrong, and unassigned atoms relative to the total number of chemical shift values in the reference lists that were assigned by conventional techniques. The results where our results are better than the PINE results are marked in bold. The run time is in hours.

Our algorithms GA, NSGA2, and NSGA3 assigned 63–71%, 62–74% and 7–9% of the atoms correctly in all fragments, respectively, and the other atoms were assigned incorrectly. PINE provided 65–73% correct atom assignments, 15–20% of the atoms were assigned incorrectly, and 11–16% of the atoms remained unassigned. FLYA yielded 74–85% correct atom assignments, whereas 7–11% of the atoms were assigned incorrectly, and 6–19% remained unassigned, in other words, without strong assignments [16]. The runtimes of the algorithms were 0.4–0.6 h for NSGA2, 0.5–0.7 h for NSGA3, 0.4–0.6 h for GA, 0.9–1.1 h for FLYA on the personal computer and 0.1 h for PINE on the webserver.

In the first fragment, our algorithm GA could assign more atoms correctly than PINE. All of our algorithms completed the assignments faster than FLYA and could assign all of the atoms in this fragment. In the second fragment, the correct assignments of our algorithm NSGA2 are more than PINE. On the third fragment, our algorithm NSGA2 could assign more number of correct atoms as PINE did. On the fourth fragment, our algorithm NSGA2 yielded more correct assignments than PINE did and our algorithms could find the results faster than FLYA did. Our algorithms NSGA2 and GA could find comparable results to PINE on the fifth fragment with a better runtime than FLYA on this dataset. On the sixth fragment, our algorithms could not assign as many correct atoms as FLYA and PINE did. Our algorithm GA could provide better results than PINE did on the seventh fragment. Additionally, the runtime of our algorithms is better than FLYA on this fragment.

Table 4.2 : Chemical shift assignment results of our algorithms NSGA2, NSGA3 and GA, and the state-of-the-art programs FLYA and PINE on the seven fragments of the Fes SH2 protein.

Fragment	Class	NSGA2	NSGA3	GA	FLYA	PINE
1st	Correct	238 (67%)	25 (7%)	**245 (69%)**	274 (77%)	245 (69%)
	Wrong	118 (33%)	331 (93%)	111 (31%)	38 (11%)	54 (15%)
	Unassigned	0	0	0	44 (12%)	57 (16%)
	Run time	**0.4**	**0.5**	**0.4**	1.0	0.1
2nd	Correct	**249 (74%)**	27 (8%)	228 (67%)	287 (85%)	246 (73%)
	Wrong	89 (26%)	311 (92%)	110 (33%)	30 (9%)	51 (15%)
	Unassigned	0	0	0	21 (6%)	41 (12%)
	Run time	**0.6**	**0.7**	**0.6**	0.9	0.1
3rd	Correct	**240 (67%)**	31 (9%)	225 (63%)	277 (78%)	233 (65%)
	Wrong	116 (33%)	325 (91%)	131 (37%)	27 (8%)	72 (20%)
	Unassigned	0	0	0	52 (15%)	51 (14%)
	Run time	**0.5**	**0.6**	**0.4**	1.0	0.1
4th	Correct	**252 (69%)**	31 (8%)	237 (65%)	287 (79%)	240 (66%)
	Wrong	113 (31%)	334 (92%)	128 (35%)	26 (7%)	69 (19%)
	Unassigned	0	0	0	52 (14%)	56 (15%)
	Run time	**0.5**	**0.6**	**0.5**	1.0	0.1
5th	Correct	238 (67%)	31 (9%)	227 (64%)	285 (80%)	249 (70%)
	Wrong	118 (33%)	325 (91%)	129 (36%)	30 (8%)	53 (15%)
	Unassigned	0	0	0	41 (12%)	54 (15%)
	Run time	**0.5**	**0.5**	**0.4**	1.0	0.1
6th	Correct	225 (62%)	25 (7%)	233 (64%)	287 (79%)	253 (70%)
	Wrong	137 (38%)	337 (93%)	129 (36%)	26 (7%)	62 (17%)
	Unassigned	0	0	0	49 (14%)	47 (13%)
	Run time	**0.5**	**0.6**	**0.5**	1.0	0.1
7th	Correct	240 (67%)	31 (9%)	**252 (71%)**	265 (74%)	246 (69%)
	Wrong	117 (33%)	326 (91%)	105 (29%)	24 (7%)	71 (20%)
	Unassigned	0	0	0	68 (19%)	40 (11%)
	Run time	**0.5**	**0.6**	**0.5**	1.1	0.1

4.2.3 Graphical representation

The graphical representation of the results of our GA algorithm on the 4th fragment is given in Figure 4.2. Each atom is represented by a rectangle here, and the colors represent the correctness of a single atom's assignment. Green means that the chemical shift assigned is marked as correct. Correct means that the difference between the

assigned chemical shift frequency value and the reference value is smaller than the tolerance values of the atoms, which are 0.03 ppm for ^{1}H, 0.4 ppm for ^{13}C, and 0.4 ppm for ^{15}N. Red means that the assigned chemical shift value is not on that range, so they are marked as wrong. Blue means that the algorithm could assign the atom however there does not exist any reference value for that atom to compare. Black means that there is a reference chemical shift value for that atom. However, the algorithm did not assign it. In FLYA, the dark-colored ones mean that 80% of the twenty runs yielded the same result. The H^N/H^α row shows the atoms H^N and H^α atoms of the residues.

Similarly, the $N/C^\alpha/C'$ shows the results for the N, C^α, and C' atoms. The lines β, γ, δ, ε, ζ, and η show the assignments of the atoms in the sidechain. The GA could make 65% correct assignments on this dataset. As shown in the figure, the wrong ones mostly lay on the sidechain of the molecule. The results of FLYA and PINE can be found in Figure A.1 in Appendix A. FLYA could assign 79% of the atoms correctly. PINE could assign 66% of the atoms correctly on this dataset.

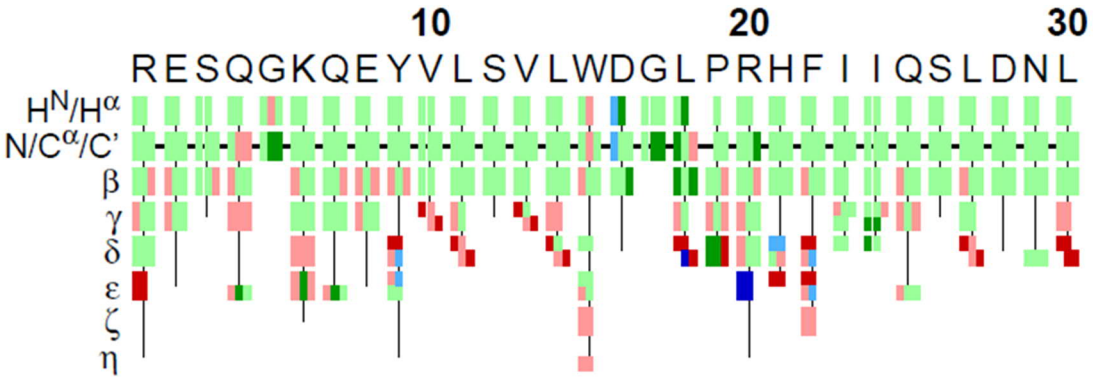

Figure 4.2 : The graphical representation of assignments done by the GA.

4.2.4 Repetition parameter tuning

The repetition parameter is set to 10 by default in our tests. We analyzed the effect of the repetition parameter on our experiments. We set this number to 50 for the fourth fragment of the Fes SH2 protein and summarized these results in Table 4.3. In these experiments, FLYA is run with its default values, where 20 independent runs proceed. The results are the consolidation of these runs. PINE is run with its default values via its webserver. The algorithm's run time is evaluated as the time between the input file

submission and result arrival. Mean values from 5 independent runs of the algorithms are given as the final results of the NSGA2, NSGA3 and GA algorithms. We applied the constructive initialization approach in each algorithm. Each population in our algorithms has 50 individuals. The crossover and mutation rate are set to 1.0. For the repetition comparison, all of the other parameters are the same. The only difference is the repetition parameter, which is set to 10 in the first block and 50 in the second block.

Table 4.3 : Chemical shift assignment results on the 4th fragment of the Fes SH2 protein.

	FLYA	PINE	10 Repetition			50 Repetition		
			NSGA2	NSGA3	GA	NSGA2	NSGA3	GA
Correct	287 (79%)	240 (66%)	252 (69%)	31 (8%)	237 (65%)	256 (70%)	42 (12%)	259 (71%)
Wrong	26 (7%)	69 (19%)	113 (31%)	334 (92%)	128 (35%)	109 (30%)	323 (88%)	106 (29%)
Unassigned	52 (14%)	56 (15%)	0	0	0	0	0	0
Run time	1.0	0.1	0.5	0.6	0.5	7.2	7.9	6.4

Table 4.3 shows the total number of correct, wrong, and unassigned atoms for all experiments. Our algorithms NSGA2, NSGA3, and GA, assigned 70%, 12% and 71% of the atoms correctly with 50 repetitions, which corresponds to an increase by 1%, 4% and 6% for NSGA2, NSGA3 and GA, respectively. The runtime of all our algorithms increased with 50 repetitions compared to 10 repetition results.

4.2.5 Iteration count tuning

We applied different iteration counts in our algorithms. The default iteration count is set to 10,000 in our experiments. In order to evaluate its impact on the results, we set this parameter to 100,000 and 1,000,000 in our NSGA2, NSGA3, and GA algorithms. The results of GA with these iterations counts are given in Table 4.4. The results of all our algorithms, NSGA2, NSGA3, and GA, are given in Table B.1 in Appendix B.

The tables show the total number of correct, wrong, and unassigned atoms and run time in hours. The numbers in the parenthesis show the percentage of these atoms among the total atom count. As shown in the results, increasing this iteration count did not significantly improve the results and the runtime of each algorithm is increased with the increased number of iterations.

Table 4.4 : Chemical shift assignment results on the 2nd and 4th fragments of the Fes SH2 protein with different iteration numbers of GA algorithm.

Fragment	Class	FLYA	PINE	GA Iterations		
				10,000	100,000	1,000,000
2	Correct	287 (85%)	246 (73%)	228 (67%)	216 (64%)	221 (65%)
	Wrong	30 (9%)	51 (15%)	110 (33%)	122 (36%)	117 (35%)
	Unassigned	21 (6%)	41 (12%)	0	0	0
	Run time (h)	0.9	0.1	0.6	1.1	6.1
4	Correct	287 (79%)	240 (66%)	237 (65%)	237 (65%)	233 (64%)
	Wrong	26 (7%)	69 (19%)	128 (35%)	128 (35%)	132 (36%)
	Unassigned	52 (14%)	56 (15%)	0	0	0
	Run time (h)	1.0	0.1	0.5	1.2	6.4

4.2.6 Evaluation

In summary, the results vary depending on the dataset. Our NSGA2 or GA algorithms could find more correct atoms than PINE in some fragments, except the fifth and sixth fragments. Additionally, our algorithms had better performance than FLYA concerning the runtime in all fragments. Our algorithms could assign all of the atoms and fulfilled the completeness objective in all of the fragments. However, 6–19% of the atoms remained unassigned on FLYA, and PINE could not assign 11–16% of atoms. On the other hand, fulfilling the completeness objective increased the number of wrong atom assignments. Our NSGA3 algorithm could not perform as well as our other algorithms in all of the fragments.

We performed further analysis on the results we achieved from these experiments. First of all, we analyzed the objectives and their trends to determine the reason behind the outperformance of the NSGA3 algorithm. Afterwards, we checked the overlapping atoms that were assigned correctly by the algorithms. These results showed that our algorithms could assign many atoms successfully that could not be assigned by other methods. Furthermore, we introduced a hierarchical model that will leverage the success of all these algorithms. These analysis details and their results are presented in subchapter 4.4.

4.3 Results on the Sso7d Protein

We performed further tests on the Sso7d protein from the hyperthermophilic archaeabacterium Sulfolobus solfataricus, which was previously assigned by conventional methods [79]. In these tests, we run our algorithms with their default

parameters. This subchapter presents the comparison of our algorithms, NSGA2, NSGA3, and GA, and with the two state-of-the-art programs FLYA and PINE.

4.3.1 Dataset details

There are 65 residues in the Sso7d protein. We divided this protein into two fragments in order to evaluate the behavior of our algorithms in different datasets. As shown in Figure 4.3, we created two different datasets by extracting thirty different amino acids from this protein. We excluded the starting and ending amino acids from this dataset creation process. In addition to these fragments, we applied the algorithms on the whole protein containing 65 residues, shown as the 3rd dataset in the experiment results below.

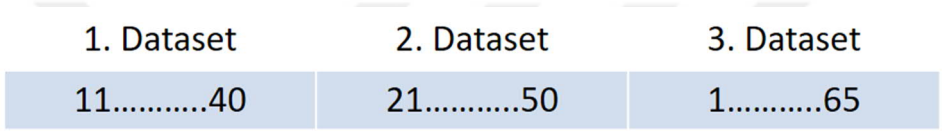

Figure 4.3 : The residues on the datasets from the 65-residue Sso7d protein.

The details of these datasets are given in Table 4.5.

Table 4.5 : Experimental dataset details.

Dataset (residues)	Expected peaks	Measured peaks	Residues	Atoms	Spectra
1 (11-40)	2351	1119	30	361	9
2 (21-50)	2316	1203	30	355	9
3 (1- 65)	5394	3527	65	721	9

The *Dataset* column of Table 4.5 shows the first and last residue of each dataset. The residues that exist in that dataset are given in the parenthesis of the same column. The *Expected peaks* column shows the number of expected peaks that are used in each dataset. These expected peaks are created by CYANA and given as input to our algorithms. The *Measured peaks* column shows the total number of measured peaks used in the experiments. The number of residues in each dataset is given in the *Residues* column. The last column, *Spectra*, shows the number of spectra in each dataset. In each of our experiments, we used the same spectrum set: N15NOESY, C13HSQC, C13NOESY, CBCANH, HNcaCO, CBCAcoNH, CcoNH, HCcoNH and HNCO.

4.3.2 Empirical results

Table 4.6 presents a summary of these experiments on the data datasets of the Sso7d protein. Our algorithms NSGA2, NSGA3, and GA, and the state-of-the-art algorithms, FLYA and PINE, are run with their default parameters as explained in subchapter 4.1. The numbers in the table are the total number of correct, wrong and unassigned atoms for each experiment. The numbers in the parenthesis show the percentages of the correct, wrong and unassigned atoms relative to the total number of chemical shift values in the reference lists that were assigned by conventional techniques. The results where our results are better than the FLYA or the PINE results are marked in bold. The run time is in hours.

Table 4.6 : Chemical shift assignment results of our algorithms NSGA2, NSGA3 and GA, and the state-of-the-art programs FLYA and PINE on the two datasets of the Sso7d protein.

Dataset	Class	NSGA2	NSGA3	GA	FLYA	PINE
1st	Correct	**317 (88%)**	30 (8%)	**314 (87%)**	335 (93%)	263 (73%)
	Wrong	**44 (12%)**	331 (92%)	**47 (13%)**	12 (3%)	55 (15%)
	Unassigned	0	0	0	14 (4%)	43 (12%)
	Run time	2.8	2.4	2.0	1.6	0.1
2nd	Correct	**319 (90%)**	30 (8%)	**307 (86%)**	323 (91%)	268 (75%)
	Wrong	**36 (10%)**	325 (92%)	**48 (14%)**	19 (5%)	43 (12%)
	Unassigned	0	0	0	13 (4%)	44 (12%)
	Run time	2.0	2.9	2.0	1.9	0.1
3rd	Correct	456 (63%)	37 (9%)	381 (53%)	588 (82%)	607 (84%)
	Wrong	265 (37%)	684 (91%)	340 (47%)	42 (6%)	67 (9%)
	Unassigned	0	0	0	91 (13%)	47 (7%)
	Run time	6.8	6.9	4.8	3.3	0.1

In the first dataset, our algorithms NSGA2, NSGA3 and GA yielded 88%, 8% and 87% of correctly assigned atoms. They could assign all of the atoms of this dataset. The state-of-the-art algorithms PINE and FLYA assigned 73% and 93% of the atoms correctly. PINE and FLYA made 15% and 3% incorrect assignments and 12% and 4% of the atoms remained unassigned, respectively. The runtime was 2 hours for GA, 2.8 hours for NSGA2 and 2.4 hours for NSGA3. The runtime was about 1.6 hours for FLYA and 0.1 hours for PINE.

In the second fragment, our NSGA2, NSGA3 and GA algorithms provided 90%, 8% and 86% correct assignments. The remaining atoms were assigned incorrectly. PINE could make 75% correct and 12% incorrect assignments, whereas 12% of the atoms remained unassigned. FLYA made 91% correct assignments and 5% incorrect assignments on this dataset. FLYA could not assign 4% of the atoms. The runtime was 2.0 hours for our algorithms NSGA2 and GA and 2.9 hours NSGA3. The runtime was 1.9 hours for FLYA and 0.1 hours for PINE on this dataset.

The third dataset of this example shows the results on the whole protein. These results show that our algorithms NSGA2, NSGA3 and GA could make 63%, 9% and 53% correct assignments respectively on this dataset, and the remaining atoms were assigned incorrectly. PINE and FLYA could assign 84% and 82% atoms correctly, respectively. PINE and FLYA did 9% and 6% wrong assignments. Our algorithms could assign all of the atoms in this dataset, whereas FLYA and PINE could not assign 13% and 7% of the atoms on this dataset, respectively.

4.3.3 Graphical representation

The graphical representation of the results of our NSGA2 algorithm on the 1st dataset is given in Figure 4.4. Each atom is represented by a rectangle here, and the colors represent the correctness of a single atom's assignment. Green means that the chemical shift assigned is marked as correct. Correct means that the difference between the assigned chemical shift frequency value and the reference value is smaller than the tolerance values of the atoms, which are 0.03 ppm for ^{1}H, 0.4 ppm for ^{13}C, and 0.4 ppm for ^{15}N. Red means that the assigned chemical shift value is not on that range, so they are marked as wrong. Blue means that the algorithm could assign the atom however, there does not exist any reference value for that atom to compare. Black means that there is a reference chemical shift value for that atom, however; the algorithm did not assign it. In FLYA, the dark-colored ones mean that 80% of the twenty runs yielded the same result. The H^N/H^α row shows the atoms H^N and H^α atoms of the residues.

Similarly, the $N/C^\alpha/C'$ shows the results for the N, C^α and C' atoms. The lines β, γ, δ, ε, ζ, and η show the assignments of the atoms in the sidechain. Our algorithms NSGA2 and GA could make 88% and 87% correct atom assignments, respectively on this

dataset. PINE assigned 73% and FLYA 93% of the atoms correctly on this dataset. As shown in the figure, the wrong ones mostly lay on the sidechain of the molecule on our NSGA2 algorithm. The graphical representation of FLYA and PINE can be found in Figure C.1 in Appendix C.

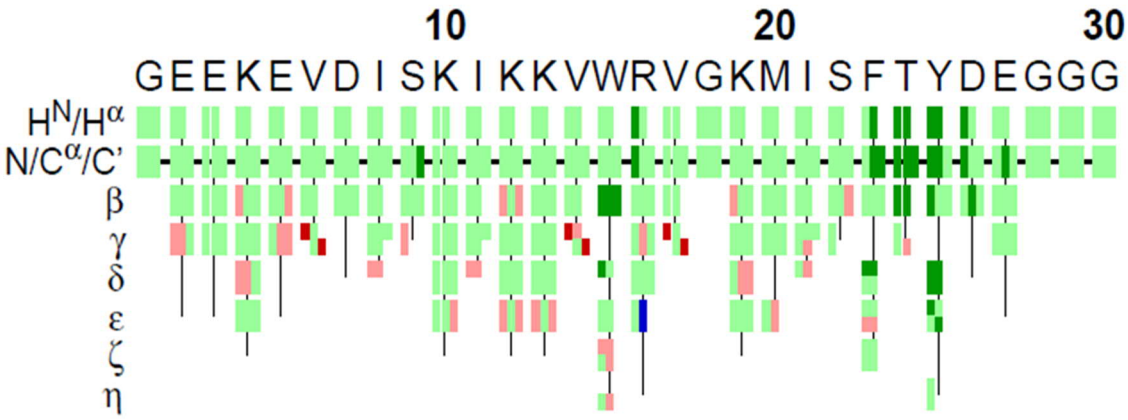

Figure 4.4 : The graphical representation of assignments done by the GA.

4.3.4 Evaluation

Our algorithms NSGA2 and GA achieved better results than PINE in all fragments of the Sso7d protein. Our NSGA2 and GA algorithms assigned around 20% more correct atoms on average than PINE in these fragments. Additionally, our algorithms NSGA2 and GA assigned less number of wrong atoms than PINE on these fragments. Furthermore, all of our algorithms could assign all of the atoms in all datasets, whereas PINE and FLYA could not assign 12% and 4% of the atoms, respectively.

We observed from comparing the results from the 1st and 2nd dataset with the 3rd dataset results that our algorithms produced significant improvements when a protein dataset is split into smaller fragments. On this protein, NSGA2 results improved from 67% correct assignments, the result on the 3rd dataset, to 89%, which is the average of the 1st and 2nd fragments. Similarly, the GA correct assignment results improved from 53% to 87%. At the same time, the ratio of the incorrect assignments decreased as well with this approach. The incorrect assignments of the NSGA2 algorithm decreased from 33%, the result on the 3rd dataset, to 11%, the average of the 1st and 2nd fragments. Similarly, the wrong assignment ratio of the GA algorithm decreased from 47%, the result on the 3rd dataset, to 13%, the average of the incorrect assignments on the 1st and 2nd fragments. This approach applies to NMR experiments with the segmental isotope

labelling [80-82] that makes it possible to measure NMR spectra containing only peaks from a given part of a protein. As a result, splitting a large protein into smaller fragments with the segmental isotope labelling improves the results of our algorithms dramatically. With this approach, the percentage of the correct assignments is increased 28% on average, and also the percentage of the incorrect assignments is decreased around 28% on average on this dataset.

We performed further exploration on these results. First, we analyzed the objectives and their trends. Additionally, we checked the overlapping atoms of these algorithms to find out the behavior of each algorithm in the search space. We observed from these results that our algorithms could make successful assignments on many atoms that the other methods could not assign. Eventually, we employed a hierarchical model in order to improve the results. These analysis results are given in the following subchapter.

4.4 Further Analysis Results

We performed further analysis as per the results we achieved in our empirical experiments. We analyzed the trends of the objectives, the overlapping atoms, and the influence of a hierarchical model on the results. This subchapter explains these analyses and their results.

4.4.1 Objective analysis

NSGA3 algorithm could not provide such good results as NSGA2 and GA algorithms. In order to find out the reason behind this behavior, we compared the values of each objective as defined in this subchapter. There are four objectives in our problem model: Shift normality, alignment, completeness and low degeneracy.

The MOEA framework can work with minimization problems only and it cannot solve maximization problems. Therefore, we converted our maximization objectives into minimization objectives. For that, we multiplied the values of the shift normality, alignment and completeness objectives by -1 so that they will be optimized by minimizing them. The degeneracy objective is already to be minimized so we did not make any change on its value. We analyzed the bilateral relationship of these objectives in the 4^{th} fragment of the Fes SH2 protein. This chapter will provide an overview of this analysis.

4.4.1.1 Results

We checked the values of each objective after each iteration and compared their values against each other to find out whether these objectives really create a trade-off or whether they are being optimized together. In these comparisons, all of these objectives are compared with their reference values during iterations. Since all of the objectives are modelled as minimization objectives, the overall goal of our method is to minimize all of these objectives. In order to make their bilateral trends, we normalized the values of these objectives for this comparison so that their maximum value is *one* and their minimum value is *zero*.

The overall results of this comparison are given in Table 4.7. Each row shows the value of the objective values after each evaluation. The following subchapters will discuss the trends of these objectives.

Table 4.7 : Objective values after each evaluation.

Evaluation Number	Shift Normality	Alignment	Degeneracy	Completeness
1	1.000	1.000	0.941	0.995
2	0.965	0.992	0.961	0.997
3	0.956	0.991	0.970	0.997
4	0.951	0.987	0.978	0.998
5	0.962	0.980	0.971	0.997
6	0.957	0.981	0.971	0.997
7	0.953	0.983	0.980	0.997
8	0.945	0.983	0.983	0.997
9	0.930	0.982	0.997	0.998
10	0.924	0.984	0.996	0.998
11	0.928	0.985	0.982	0.999
12	0.925	0.987	0.980	0.998
13	0.920	0.979	0.985	0.997
14	0.915	0.980	0.981	0.997
15	0.926	0.984	0.982	0.998
16	0.928	0.979	0.982	0.998
17	0.933	0.983	0.976	0.999
18	0.942	0.981	0.972	0.998
19	0.926	0.979	0.973	0.998

Shift normality and alignment

The behavior of shift normality and alignment objectives are given in Figure 4.5 below. As shown in the figure, these objectives are aligned with each other, and they do not create a trade-off. While the value of the shift normality objective is decreasing,

the value of the alignment objective is decreasing as well. Similarly, their values are increasing together.

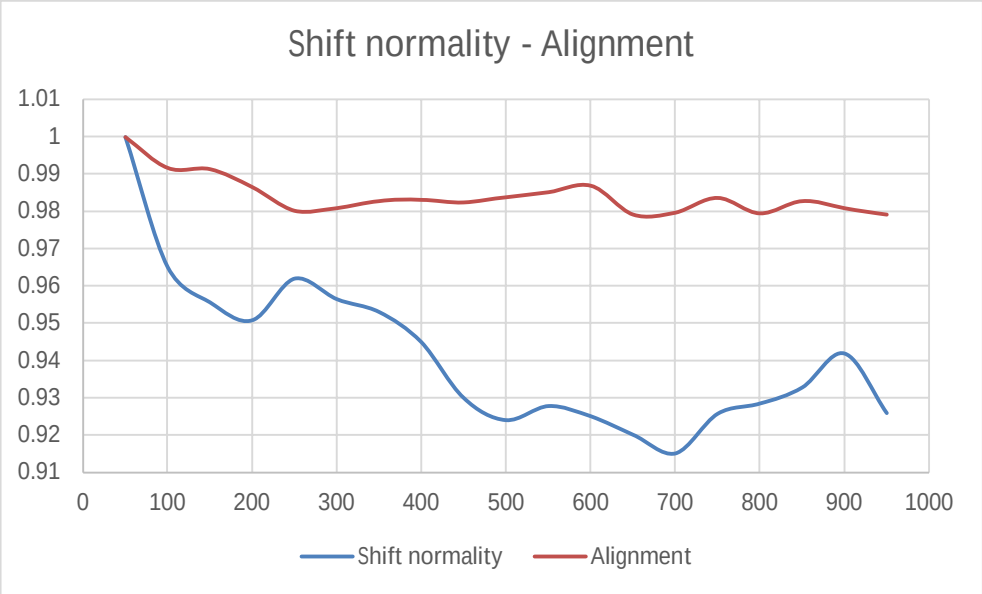

Figure 4.5 : The trend of the objectives “shift normality” and “alignment”.

Shift normality and degeneracy

The behavior of shift normality and degeneracy objectives are given in Figure 4.6. These objectives create a trade-off: While the shift normality value is minimized, the degeneracy value is maximized. At the same time, the shift normality objective is maximized while the degeneracy value is being minimized.

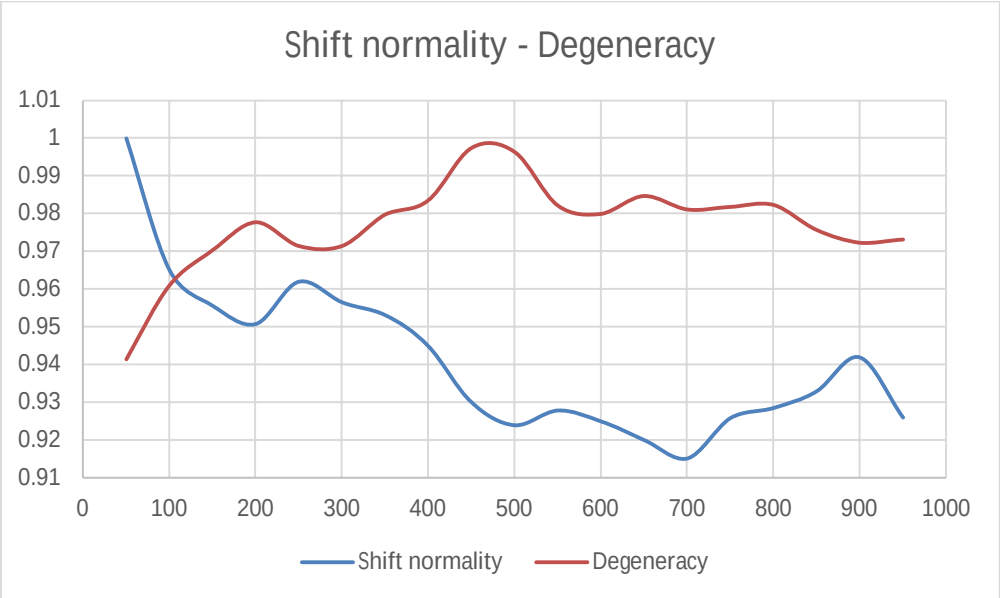

Figure 4.6 : The trend of the objectives “shift normality” and “degeneracy”.

Shift normality and completeness

The objectives shift normality and completeness create a trade-off, as shown in Figure 4.7. While the shift normality value is increasing, the completeness value decreases. Similarly, the completeness value increases with a decrease in the shift normality value.

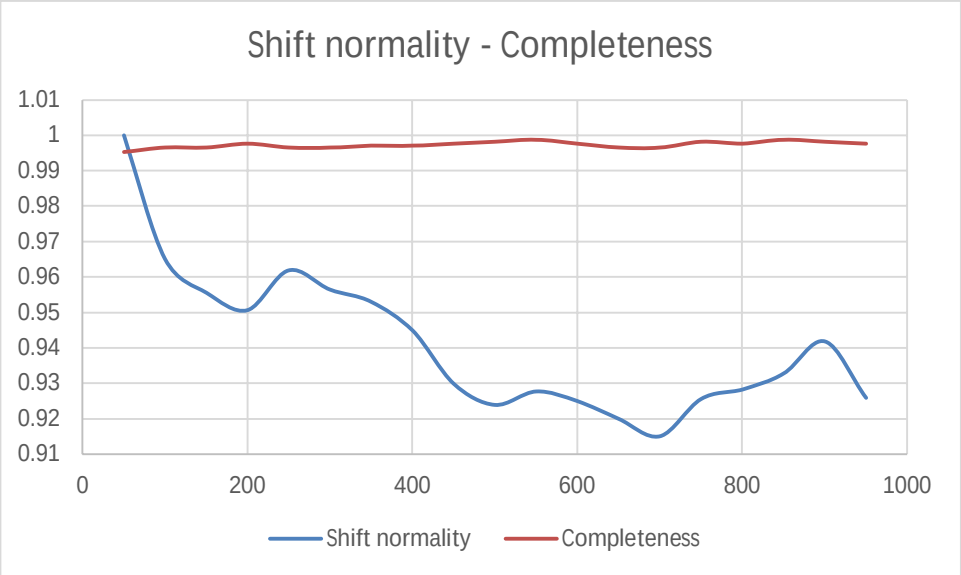

Figure 4.7 : The trend of the objectives “shift normality” and “completeness”.

Alignment and degeneracy

The alignment and degeneracy objectives create a trade-off, as shown in Figure 4.8. While the alignment objective is being minimized, the degeneracy is being maximized. Similarly, the alignment objective is being maximized while the degeneracy is being minimized.

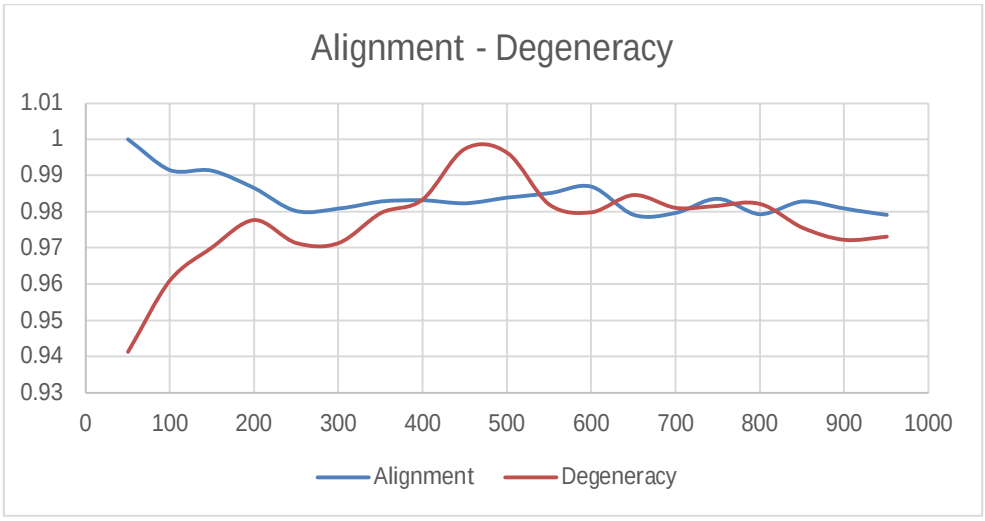

Figure 4.8 : The trend of the objectives “alignment” and “degeneracy”.

Alignment and completeness

The trend of the alignment and completeness is given in Figure 4.9. These objectives create a trade-off as shown in the figure since the completeness is being minimized while the alignment is maximized, and vice versa.

Figure 4.9 : The trend of the objectives “alignment” and “completeness”.

Degeneracy and completeness

The degeneracy and completeness objectives are aligned with each other, as shown in Figure 4.10. As a result, they do not create a trade-off and the degeneracy objective can be optimized while optimizing the completeness objective.

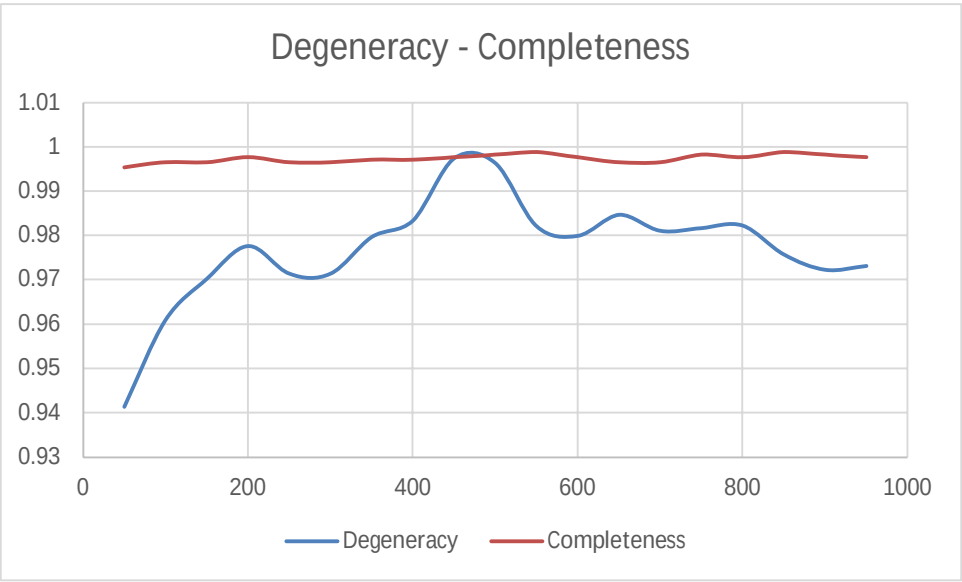

Figure 4.10 : The trend of the objectives “degeneracy” and “completeness”.

Shift normality, alignment, degeneracy, and completeness in first 1,000 iterations

The results of the four objectives, shift normality, alignment, degeneracy and completeness, are plotted in Figure 4.11 for the first 1,000 iterations.

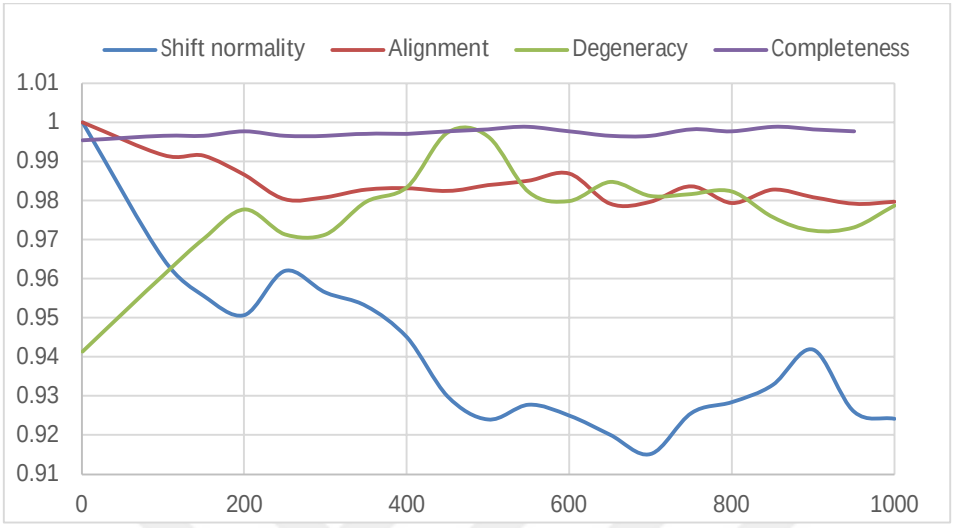

Figure 4.11 : The trend of the objectives “shift normality”, “alignment”, “degeneracy”, and “completeness” in the first 1.000 iterations.

Shift normality, alignment, degeneracy, and completeness in 10,000 iterations

The results of the four objectives, shift normality, alignment, degeneracy, and completeness, are plotted in Figure 4.12 for the 10,000 iterations. This figure shows that the correlations between the objectives blurred in the later iterations of the algorithm.

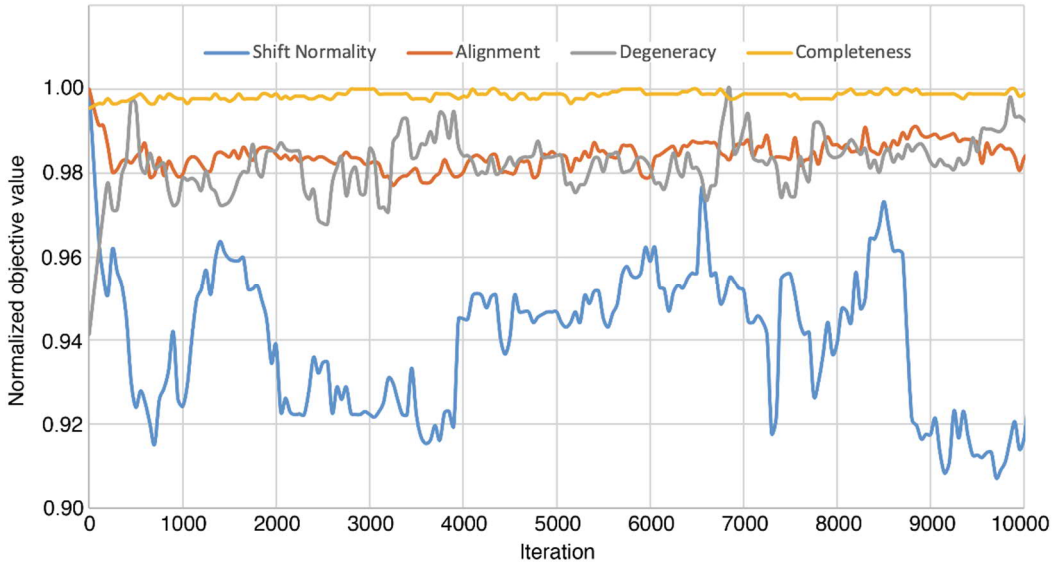

Figure 4.12 : The trend of the objectives “shift normality”, “alignment”, “degeneracy” and “completeness” in 10,000 iterations.

4.4.1.2 Evaluation

The bilateral comparison results are summarized in Table 4.8. The shift normality objective does not create a trade-off with the alignment objective. However, it has a trade-off with the degeneracy and completeness objectives. The alignment objective has a trade-off with the degeneracy and completeness objectives. On the other hand, the degeneracy objective does not have a trade-off with the completeness objective.

Table 4.8 : The objective relations with respect to trade-off.

Objective #1	Objective #2	Trade-off exists
Shift normality	Alignment	No
Shift normality	Degeneracy	Yes
Shift normality	Completeness	Yes
Alignment	Degeneracy	Yes
Alignment	Completeness	Yes
Degeneracy	Completeness	No

Even if there are four objectives in our problem model, they do not create a trade-off between each other. As seen in the results, the objectives shift normality and alignment does not create a trade-off and they can be optimized together. Similarly, the objectives degeneracy and completeness can be optimized together since they do contradict each other. These two groups of objectives create a trade-off.

As a result, there are not four contradicting objectives of the problem, but two groups of objectives: 1) shift normality and alignment 2) degeneracy and completeness. These two groups create a trade-off against each other, and the objectives in the same group are aligned in the group inside. This relation explains why the NSGA2 algorithm creates better results than the NSGA3 algorithm because the NSGA3 algorithm [71] thrives on the many-objective optimization problems where there are more than three objectives. On the other hand, the NSGA2 algorithm provides better results for problems with three or fewer objectives.

4.4.2 Overlapping atoms

In this analysis, we analyzed the success of our algorithms and found out whether they develop success on different parts of the atoms set than the state-of-the-art solutions. For that reason, we compared the atoms assigned correctly by the algorithms and illustrated their results with graphical representation in this subchapter.

4.4.2.1 Results

We analyzed the results we achieved in each algorithm and found out the atoms assigned correctly by each algorithm. In this analysis, we compared these atoms against each other and provided their results in this subchapter by comparing different sets of algorithms against each other.

Empirical results

We compared the atoms assigned correctly by our algorithms with the ones that FLYA and PINE correctly assigned. We collected the results of our algorithms together in this calculation and compared the whole result set of our algorithms against the FLYA and PINE results. We ignored the wrong and unassigned atoms in this calculation. We checked the overlapping atoms assigned by these algorithms correctly and provided the results on the fragments in Table 4.9.

Table 4.9 : Atom counts that were assigned correctly.

	FLYA only	PINE only	Our algorithms only	FLYA & PINE	Our algorithms & FLYA	Our algorithms & PINE	Our algorithms & PINE&FLYA
1st	5 (2%)	14 (5%)	36 (12%)	5 (2%)	11 (4%)	102 (34%)	124 (42%)
2nd	19 (6%)	3 (1%)	6 (2%)	11 (4%)	33 (11%)	8 (3%)	224 (74%)
3rd	16 (6%)	12 (4%)	10 (4%)	12 (4%)	43 (15%)	9 (3%)	179 (64%)
4th	17 (5%)	10 (3%)	17 (5%)	11 (3%)	49 (15%)	9 (3%)	210 (65%)
5th	17 (6%)	11 (4%)	8 (3%)	13 (5%)	34 (12%)	8 (3%)	191 (68%)
6th	19 (6%)	11 (4%)	11 (4%)	17 (6%)	34 (11%)	9 (3%)	208 (67%)
7th	13 (4%)	14 (5%)	25 (8%)	6 (2%)	29 (10%)	12 (4%)	204 (67%)

The numbers in the parenthesis in Table 4.9 are the percentages of the atoms relative to the total number of atoms that were assigned correctly by all of these methods in that fragment. The *Only FLYA* column shows the total number of atoms that were assigned only by FLYA correctly. The other algorithms could not make correct assignments on these atoms. Similarly, the *Only PINE* column shows the total number of atoms assigned by PINE correctly. The other algorithms could not make any correct assignments on these atoms. The *Only Our algorithms* column shows the total number of atoms assigned by our algorithms and could not be assigned by FLYA or PINE. Our algorithms, NSGA2, NSGA3 and GA, are handled together in this calculation. The *FLYA & PINE* column shows the total number of atoms assigned correctly by both FLYA and PINE algorithms and could not be assigned correctly by our algorithms. The *Our algorithms & FLYA* column shows the total number of atoms assigned correctly by our algorithms and FLYA and could not be assigned by PINE correctly. Similarly,

the column *Our algorithms & PINE* shows the total number of atoms assigned correctly by our algorithms and PINE and could not be assigned by FLYA correctly. In the last column, *Our algorithms & PINE & FLYA* shows the total number of atoms that were assigned by our algorithms, PINE and FLYA correctly.

Collaborative comparison

In this subchapter, we showed the percentages of the atoms correctly assigned by each algorithm on the Venn diagram. We showed the results of the 1st fragment in Figure 4.13. On this fragment, 12% of the atoms were assigned correctly only by our algorithms and they could not be assigned by PINE or FLYA correctly. 2% of the atoms were assigned correctly only by FLYA and 5% of atoms only by PINE. Our algorithms and FLYA could assign 4% of the atoms correctly in common, where 34% of the atoms were assigned only by PINE and our algorithms correctly. 42% of the atoms were assigned by our algorithms, FLYA and PINE correctly in common.

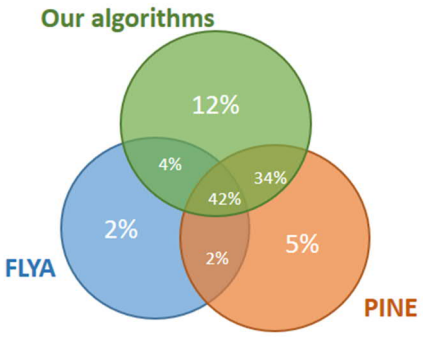

Figure 4.13 : The percentages of the correct assignments on the 1st fragment of the Fes SH2 protein.

Individual comparison

We also checked these results with our single algorithms in the Fes SH2 protein. The GA results were compared in Figure 4.14. 10% of the atoms were assigned correctly only by our GA algorithm, and they could not be assigned either by FLYA or by PINE. 3% of the atoms were assigned correctly only by FLYA. GA and FLYA could assign 3% of the atoms correctly in common. GA and PINE could make in 34% of the atoms correct common assignments. All of these three algorithms could make correct assignments on 41% of the atoms. FLYA and PINE could assign 3% of the atoms in common.

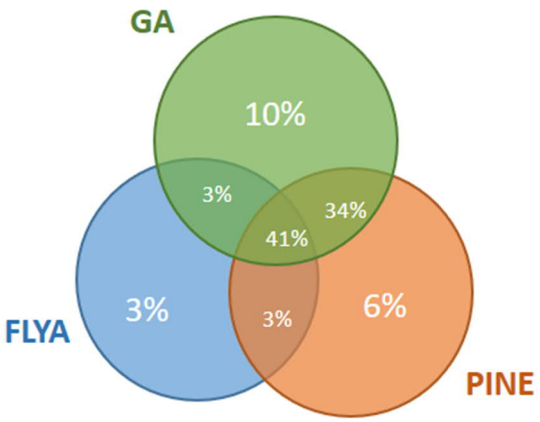

Figure 4.14 : The percentages of the correct assignments for GA on the 1st fragment of the Fes SH2 protein.

NSGA2 results are compared against FLYA and PINE in Figure 4.15. The NSGA2 algorithm could assign 8% of the whole atoms, which could not be assigned by FLYA or by PINE. 2% of the atoms were assigned correctly only by FLYA and NSGA2 algorithm. 33% of the atoms could be assigned only by PINE and NSGA2 algorithms. All of these algorithms could assign 42% of the atoms correctly in common. The 4% of the atoms could be assigned by FLYA and PINE only. The 3% of the atoms were assigned only by FLYA, whereas 8% of the atoms were assigned only by PINE correctly.

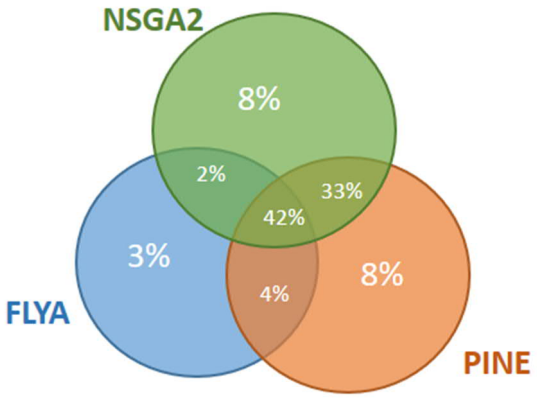

Figure 4.15 : The percentages of the correct assignments for NSGA2 on the 1st fragment of the Fes SH2 protein.

NSGA3 results are compared against FLYA and PINE in Figure 4.16. The 4% of the atoms were assigned correctly only by NSGA3. The 5% of the atoms were assigned only by FLYA correctly, and 38% were assigned only by PINE correctly. All algorithms could make 5% correct assignments in common. NSGA3 and PINE could

assign 4% of the atoms correctly in common. The 42% of the atoms were assigned by FLYA and PINE correctly.

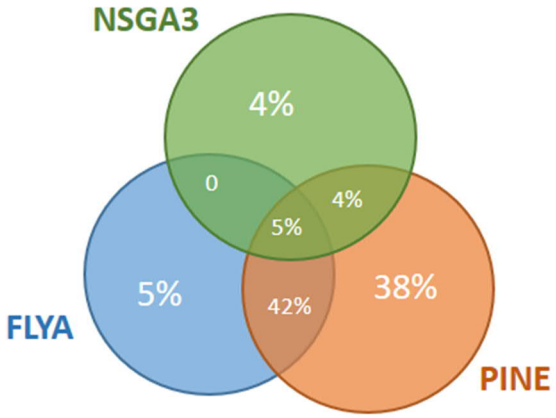

Figure 4.16 : The percentages of the correct assignments for NSGA3 on the 1st fragment of the Fes SH2 protein.

Insider comparison

In a final assessment, we compared the atoms assigned by our algorithms correctly and presented these results in Figure 4.17. GA could make correct assignments on the 11% of the atoms that the other algorithms could not assign. NSGA2 algorithm could assign 5% of the atoms correctly that were not assigned by GA or NSGA3. NSGA3 made correct assignments on the 1% of the atoms that the other algorithms could not assign. NSGA2 and GA could make correct assignments on 74% of the atoms in common. The NSGA2, NSGA3 and GA algorithms assigned 9% of the atoms correctly in common.

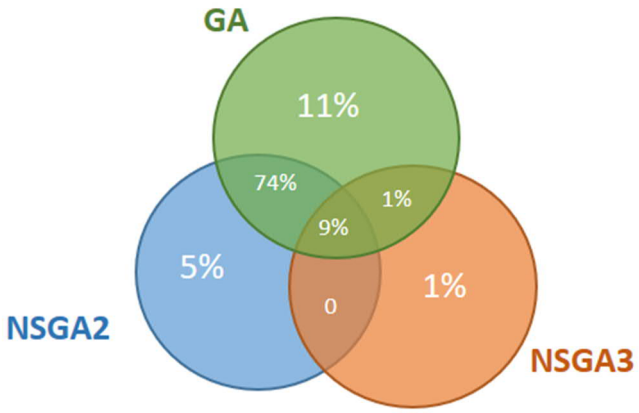

Figure 4.17 : The percentages of the NSGA2, NSGA3, and GA algorithms on the 1st fragment of the Fes SH2 protein.

4.4.2.2 Evaluation

The results of the overlapping atoms analysis showed that there are many atoms, such as 12% of the total atoms, that could be assigned only by our algorithms correctly. For that reason, we compared the individual results of our algorithms against FLYA and PINE. The results showed that our algorithms could make a significant impact on the results and can make correct assignments on the atoms that could not be assigned by FLYA or PINE correctly. As the final assessment, we compared the correctly assigned atoms of our algorithms against each other. These results showed that GA is the best approach that finds more distinctive correct atom assignments among our algorithms. As per the results of this analysis, we developed the hierarchical model to leverage from the correctly assigned atoms of our algorithms, especially from the GA algorithm, which is explained in the following subchapter.

4.4.3 Hierarchical Model

We designed a hierarchical approach to solve the chemical shift assignment problem by combining state-of-the-art programs with our solutions. As seen in the overlapping atoms analysis, our algorithms can successfully make assignments on several atoms, where neither FLYA nor PINE could assign atoms correctly. For that reason, we modeled an approach that will run both state-of-the-art solutions and our algorithms on our datasets with the following method: If the state-of-the-art solution, FLYA or PINE, did an assignment to an atom, then that result is taken as the eventual chemical shift value of that atom. The assignments of the atoms that FLYA or PINE could not assign received the assignments from our algorithm.

4.4.3.1 Results

The results of this approach are given in Table 4.10, along with the actual results of the FLYA and GA algorithms on the same datasets. The results show the total number of atoms that are *correct*, *wrong*, and *unassigned*. The numbers in the parenthesis show the percentage of these atom assignments in the total atom numbers in that fragment. The best results are marked in bold.

In the fragments of the Fes SH2 protein, FLYA could assign 74–85% of the atoms correctly, whereas 6–19% of the atoms remained unassigned. With the hierarchical approach, 83–85% of the atoms were assigned correctly, and all of the atoms are

assigned. In the datasets of the Sso7d protein, FLYA could make 82–93% correct assignments. Our hierarchical method increased this ratio to 89–97% of correct assignments. In the first fragment of the Fes SH2 protein, the ratio of the correct atom assignments was 77% in FLYA. This ratio is increased to 84% by the hierarchical model. The new model could assign 24 more atoms correctly than FLYA did in this fragment.

Table 4.10 : FLYA, GA, and hierarchical model results on the Fes SH2 protein.

Protein	Dataset	Classification	FLYA	GA	Hierarchical Model
Fes SH2	**1st**	Correct	274 (77%)	245 (69%)	**298 (84%)**
		Wrong	38 (11%)	111 (31%)	58 (16%)
		Unassigned	44 (12%)	0	0
	2nd	Correct	287 (85%)	228 (67%)	**287 (85%)**
		Wrong	30 (9%)	110 (33%)	51 (15%)
		Unassigned	21 (6%)	0	0
	3rd	Correct	277 (78%)	225 (63%)	**301 (85%)**
		Wrong	27 (8%)	131 (37%)	55 (15%)
		Unassigned	52 (15%)	0	0
	4th	Correct	287 (79%)	237 (65%)	**308 (84%)**
		Wrong	26 (7%)	128 (35%)	57 (16%)
		Unassigned	52 (14%)	0	0
	5th	Correct	285 (80%)	227 (64%)	**296 (83%)**
		Wrong	30 (8%)	129 (36%)	60 (17%)
		Unassigned	41 (12%)	0	0
	6th	Correct	287 (79%)	233 (64%)	**306 (85%)**
		Wrong	26 (7%)	129 (36%)	56 (15%)
		Unassigned	49 (14%)	0	0
	7th	Correct	265 (74%)	252 (71%)	**297 (83%)**
		Wrong	24 (7%)	105 (29%)	60 (17%)
		Unassigned	68 (19%)	0	0
Sso7d	**1st**	Correct	335 (93%)	314 (87%)	**349 (97%)**
		Wrong	12 (3%)	47 (13%)	12 (3%)
		Unassigned	14 (4%)	0	0 (0%)
	2nd	Correct	323 (91%)	307 (86%)	**334 (94%)**
		Wrong	19 (5%)	48 (14%)	21 (6%)
		Unassigned	13 (4%)	0	0 (0%)
	3rd	Correct	588 (82%)	381 (53%)	**641 (89%)**
		Wrong	42 (6%)	340 (47%)	80 (11%)
		Unassigned	91 (13%)	0	0

Additionally, FLYA could not assign 12% of the atoms on this dataset. With the new model, all of the atoms were assigned. The results are very similar on each data fragment. On the seventh fragment of the Fes SH2 protein, the correct assignment ratio of FLYA was 74%. The hierarchical method assigned 83% of the atoms correctly and could correctly assign 32 atoms more than FLYA did, which is 9% of the whole atom set. As seen on the table, the ratio of the correct atom assignments is increased in

almost all datasets and all of the atoms are assigned by the new hierarchical model ultimately. The only drawback of this model is the increase in the number of wrong atom assignments when it is compared to FLYA.

In addition to FLYA, we applied the hierarchical model with the PINE algorithm. These results are given in Table 4.11. The results show the total number of *correct*, *wrong*, and *unassigned* atoms. The numbers in the parenthesis show the percentage of these atom assignments in the total atom numbers in that fragment. The best results are marked in bold.

Table 4.11 : PINE, GA, and hierarchical model results on the Fes SH2 protein.

Protein	Fragment	Classification	PINE	GA	Hierarchical Model
Fes SH2	**1st**	Correct	245 (69%)	245 (69%)	**265 (74%)**
		Wrong	54 (15%)	111 (31%)	91 (26%)
		Unassigned	57 (16%)	0	0
	2nd	Correct	246 (73%)	228 (67%)	**266 (79%)**
		Wrong	51 (15%)	110 (33%)	72 (21%)
		Unassigned	41 (12%)	0	0
	3rd	Correct	233 (65%)	225 (63%)	**245 (69%)**
		Wrong	72 (20%)	131 (37%)	111 (31%)
		Unassigned	51 (14%)	0	0
	4th	Correct	240 (66%)	237 (65%)	**268 (73%)**
		Wrong	69 (19%)	128 (35%)	97 (27%)
		Unassigned	56 (15%)	0	0
	5th	Correct	249 (70%)	227 (64%)	**277 (78%)**
		Wrong	53 (15%)	129 (36%)	79 (22%)
		Unassigned	54 (15%)	0	0
	6th	Correct	253 (70%)	233 (64%)	**278 (77%)**
		Wrong	62 (17%)	129 (36%)	84 (23%)
		Unassigned	47 (13%)	0	0
	7th	Correct	246 (69%)	252 (71%)	**264 (74%)**
		Wrong	71 (20%)	105 (29%)	93 (26%)
		Unassigned	40 (11%)	0	0
Sso7d	**1st**	Correct	263 (73%)	314 (87%)	**297 (82%)**
		Wrong	55 (15%)	47 (13%)	64 (218%)
		Unassigned	43 (12%)	0	0
	2nd	Correct	268 (75%)	307 (86%)	**302 (85%)**
		Wrong	43 (12%)	48 (14%)	53 (15%)
		Unassigned	44 (12%)	0	0
	3rd	Correct	607 (84%)	381 (53%)	**623 (86%)**
		Wrong	67 (9%)	340 (47%)	98 (14%)
		Unassigned	47 (7%)	0	0

In the fragments of the Fes SH2 protein, the PINE algorithm assigned 65–73% of the atoms correctly. PINE assigned 15–20% atoms incorrectly, whereas 11–16% of the atoms remained unassigned. On the same fragments, the hierarchical model made 69–79% correct atom assignments and in half of these datasets and it assigned all of the

atoms. In the datasets of the Sso7d protein, PINE assigned 73–84% of the atoms correctly and 9–15% incorrectly. The remaining 7–12% of the atoms could not be assigned. The hierarchical model could make 82–86% correct assignments on these datasets. In all of the fragments, all of the atoms were assigned. In the first fragment of the Fes SH2 protein, the ratio of the correct atom assignments was 69% in PINE. This ratio is increased to 74% by the hierarchical model. The new model could assign 20 more atoms correctly than PINE did.

Additionally, PINE and FLYA could not assign 13% and 11% of the atoms on average for these datasets, respectively. The new model could assign almost all of the atoms and provided full completeness on the datasets. As seen on the tables, the correct atom assignments are increased in all datasets and almost all of the atoms are assigned by the new hierarchical model ultimately.

4.4.3.2 Evaluation

In the fragments of the Sso7d protein, the hierarchical model assigned in average 9% more atoms correctly than PINE did. For all the datasets we used, the hierarchical model could reach 6% more correct assignments than PINE and 5% more correct assignments than FLYA. This ratio of the improvement of the new model is increased to 10% of the atoms in these datasets. As a result, with the new hierarchical model, we could increase the number of correct atom assignments in all fragments. At the same time, the new model could assign all of the atoms in all datasets and it provided full completeness.

On the other hand, the number of wrong assignments is increased in the new hierarchical approach compared to FLYA and PINE. This impact can be further analysed in future work.

4.5 Discussion

The percentages of the total number of atoms correctly assigned by the algorithms are summarized in Table 4.12. These values are directly extracted from the results of the algorithms in Table 4.2 and Table 4.6. The results of our algorithms that outperformed any of the state-of-the-art algorithms are marked in bold.

Table 4.12 : The percentages of the atoms that are assigned correctly on the seven fragments of the Fes SH2 protein.

Protein	Fragment	NSGA2	NSGA3	GA	FLYA	PINE
Fes SH2	1st	67%	7%	**69%**	77%	69%
	2nd	**74%**	8%	67%	85%	73%
	3rd	**67%**	9%	63%	78%	65%
	4th	**69%**	8%	65%	79%	66%
	5th	67%	9%	64%	80%	70%
	6th	62%	7%	64%	79%	70%
	7th	67%	9%	**71%**	74%	69%
Sso7d	1st	**88%**	8%	**87%**	93%	73%
	2nd	**90%**	8%	**86%**	91%	75%
	ALL	67%	9%	53%	82%	84%

These results show that our algorithms NSGA2 and GA could provide better results than PINE in almost all datasets concerning the total number of the correct atom assignments. Our algorithm NSGA3 could not produce such good results as NSGA2 and GA concerning the total number of the correct assigned atoms. Our further analysis showed that our problem has only two objectives that create a trade-off. This outcome explains the performance of the NSGA3 algorithm since NSGA3 is thriving on the many-objective optimization problems with more than three objectives.

Our algorithms had better performance than FLYA with respect to the runtime on the Fes SH2 protein fragments and similar runtime of the datasets of Sso7d protein. Additionally, all of our algorithms could assign almost all of the atoms in all fragments, whereas PINE and FLYA could not fulfill the completeness objective as our algorithms did. On the other hand, fulfilling the completeness objective increased the number of wrong atom assignments in our algorithms. Nevertheless, our NSGA2 algorithm could still make fewer wrong assignments than PINE did in the fragments of the Sso7d protein. Furthermore, we observed in our results on the Sso7d protein that splitting a large protein into smaller fragments improved our algorithms' results dramatically; the percentage of the correct assignment is increased around 28% and the percentage of the incorrect assignments is decreased around 28%.

In order to leverage the success of our algorithms in the correct atom assignments, we performed further analysis on the algorithms' results. We compared the atoms that were assigned correctly by all algorithms against each other. As a result, we figured out that our GA and NSGA2 algorithms could perform correct assignments on different atoms, where neither PINE nor FLYA could succeed. This analysis shows

that our algorithms can provide good solutions on some parts of the datasets, where FLYA or PINE could not produce any successful results.

In order to take advantage of this success of our algorithms, we developed a hierarchical model. We firstly ran FLYA on a dataset and accepted all of its assignments. For the remaining atoms, which FLYA could not assign, we received the assignments from our GA algorithm. We employed the same approach for the PINE algorithm by first running the PINE algorithm and then GA on the remaining unassigned atoms. The hierarchical model results showed that this model improved the success of the state-of-the-art algorithms concerning the total number of correct assignments. In other words, GA could make correct assignments on some parts of the datasets, where FLYA or PINE could not make any assignments at all. Furthermore, this new model could assign almost all of the atoms, whereas FLYA or PINE could not make a complete assignment. On the other hand, the number of wrong assignments is also increased with the new hierarchical model compared to the state-of-the-art algorithms. This influence can be minimized with further study.

Our algorithms can make a fully automatic chemical shift assignment for both backbone and sidechain atoms of macromolecules without any manual interactions. Additionally, there does not exist any need for fine-tuning or parameter optimization for our algorithms. They can produce the results without any dataset or protein-specific changes. The results showed that our approaches could provide good results in a very efficient time interval for several different experiments. Especially for some of the datasets, our approach can produce better results in a shorter time than the current state-of-the-art algorithms for the problem. Additionally, the development of the problem and its interactions with the algorithms is implemented on the MOEA framework, which enables the extension of the problem with new algorithms easily.

5. CONCLUSION AND RECOMMENDATIONS

In this thesis, we developed a novel method with evolutionary algorithms for solving the chemical shift assignment problem of the Nuclear Magnetic Resonance (NMR) spectroscopy. Initially, we presented a comprehensive background of the problem and studied the state-of-the-art solutions for this problem. Then, we proposed our methods that solve the chemical shift assignment problem with evolutionary algorithms. Furthermore, we applied our methods to several different experiments and proposed additional improvements on our methods depending on our further analysis of the results.

Initially, we analyzed the background of the chemical shift assignment problem in detail from the computer science and natural sciences perspective. The problem itself is originated from the NMR spectroscopy experiment steps, and the methods to solve this problem are in the computer science research area. Therefore, both research areas are studied in detail. Afterwards, the existing methods with the baseline algorithms for solving this problem are investigated with their existing challenges.

Then, we proposed our methods to solve the chemical shift assignment problem. We applied the standard genetic algorithm (GA) and the multi-objective optimization algorithms Non-dominated Sorting Genetic Algorithm II (NSGA2) and Non-dominated Sorting Genetic Algorithm III (NSGA3) on this problem. While defining the methods, we first defined the problem model, along with the existing conditions and the score function. We applied fine-tuning and necessary changes to the algorithms depending on the NMR-specific behavior. Afterwards, we applied further improvements to our methods, such as optimizing the diversity in the population and the size of the initial population, the initialization method of the population, the fine-tuning on the crossover rate, mutation rate and the repetition parameter values. We also implemented hybrid algorithms by combining a hill-climbing algorithm and our proposed algorithms.

We applied our algorithms on several different datasets and analyzed the results concerning performance and efficiency. We compared our algorithms' results against the state-of-the-art algorithms: FLYA and PINE. For the seven fragments of the Fes SH2 protein, our algorithms NSGA2 or GA provided more correct assignments than PINE did in all of these fragments, except the fifth and sixth fragments. We also performed experiments on the Sso7d protein. The results on the fragments of this protein showed that our algorithms NSGA2 and GA provided better results than PINE.

Additionally, we observed from the third dataset of the Sso7d protein that the behavior of our algorithms differs when the protein size increases. The results showed in this dataset that more conventional methods yielded better results than the multi-objective approach for this dataset. Furthermore, our algorithms had better runtime than FLYA on the fragments of the Fes SH2 protein and similar runtime of the fragments of the Sso7d protein. Our algorithms could assign all of the atoms in all datasets, which creates a considerable completeness success of the problem. In contrast, FLYA and PINE algorithms could not provide a complete assignment. Furthermore, we observed in our results on the Sso7d protein that splitting a large protein into smaller fragments improved our algorithms' results dramatically: on average 28% of the incorrectly assigned atoms are assigned correctly.

Our approaches could provide around a 70% success rate on these datasets and assign many atoms that other methods could not assign. The percentage of correct assignments for these fragments by the NSGA2 and GA algorithms was comparable to those from FLYA and PINE, whereas the NSGA3 algorithm yielded less than 10% correct assignments. The results of our objective analysis showed that there are only two objectives of the problem creating trade-off, which was the reason behind the results of the NSGA3 algorithm since it provides better results on the problems with more than three objectives.

Lastly, we worked on a different approach that would leverage the success rate of our algorithms and would use their advantage of the high number of correct assignments on the unassigned parts from the state-of-the-art algorithms. After our analysis, we figured out that our algorithms could provide good solutions on the parts of the datasets, where FLYA or PINE could not make any successful assignments at all. In order to take advantage of this success of our algorithms, we developed a hierarchical model. The hierarchical method combines a state-of-the-art algorithm and our

algorithm GA and takes advantage of each algorithm. This approach improved the overall success of the algorithms by providing more correct atom assignments than FLYA or PINE could produce alone.

Our algorithms focus on the completeness objective and assign almost all atoms to maximize the overall completeness. This approach increases the total number of correct assignments in total. On the other hand, the ratio of the wrong assignments increases with this approach simultaneously. We propose further work to focus on a balanced combination of the completeness objective and ratio of the wrong assignments. One can investigate the problem model to reduce the total number of assigned atoms in our methods with the motivation that instead of assigning an atom wrongly, it should stay unassigned. Even the hierarchical model can be more optimized by decreasing the ratio of the wrong assignments but still keeping the success on the atom assignments, where only our algorithms could provide correct assignment results.

Additional further work can be performed on the score function of the problem. We designed four different objectives and realized that only two of them create a trade-off. Therefore, the score function can be optimized by removing some parts of the objectives from the formula to improve the performance of the algorithms. Another further work can be combining three of our algorithms together to leverage each other's success and to provide better results.

As a result, we implemented a novel method that could successfully make assignment of the significant part of the backbone and sidechain atoms of macromolecules fully automatically without any manual interactions and without any protein-specific and data-specific parameter changes. When it is compared to the manual assignment process, our method will reduce the total time and cost spent on the chemical shift assignment in NMR spectroscopy and it does not require a spectrometer specialist for the assignments; so it provides reproducible results that do not rely on the subjective spectroscopist decisions. When it is compared to the state-of-the-art algorithms, our approach could provide around 70% success rate on these datasets and could assign many atoms that could not be assigned by other methods. The percentage of correct assignments for these datasets by our NSGA2 and GA algorithms were comparable to those from the two very well known, successful state-of-the-art algorithms: FLYA and PINE. In almost all the experiments, our algorithm could provide better result than one

of the state-of-the-art algorithms. Additionally, in this thesis, the multi-objective optimization techniques are implemented on the chemical shift assignment problem for the first time. The multi-objective model of optimization enables an easy expansion of the problem model with new objectives and constraints such as the volume and the shape of the peaks, which have not been evaluated yet on this problem. Therefore, our implementation will provide further flexibility for the chemical shift assignment problem model. Furthermore, our method is implemented on the MOEA framework, which enables further collaboration with the existing and even with the new algorithms on the framework easily.

REFERENCES

[1] **Dyer M. and Greenhil C.** (2000). The complexity of counting graph homomorphisms, *Random Structures and Algorithms, 17*:260-289.

[2] **Wüthrich K.** (1986). *NMR of Proteins and Nucleic Acids*, Wiley, New York.

[3] **Yee A. A., Savchenko A., Ignachenko A., Lukin J., Xu X., Skarina T., Evdokimova E., Liu C. S., Semesi A. … Arrowsmith C. H.** (2005). NMR and X-ray crystallography, complementary tools in structural proteomics of small proteins. *Journal of the American Chemical Society, 127*:16512-16517.

[4] **Cavanagh J., Fairbrother W. J., Palmer III A. G., Rance M. and Skelton N. J.** (2007). *Principles and Practice Protein NMR Spectroscopy. Elsevier*, San Diego.

[5] **RCSB Protein Data Bank.** Retrieved on 21.12.2020 from https://www.rcsb.org/pdb/home/home.do.

[6] **Leutner M., Gschwind R. M., Liermann J., Schwarz C., Gemmecker G. and Kessler H.** (1997). Automated backbone assignment of labeled proteins using the threshold accepting algorithm. *Journal of Biomolecular NMR, 11*: 31–43.

[7] **Lukin J. A., Gove A. P., Talukdar S. N. and Ho C**. (1997). Automated probabilistic method for assigning backbone resonances of (^{13}C,^{15}N)-labeled proteins. *Journal of Biomolecular NMR, 9*: 151–166.

[8] **Buchler N. E. G., Zuiderweg E. R. P., Wang H. and Goldstein R. A.** (1997). Protein heteronuclear NMR assignments using mean-field simulated annealing. *Journal of Magnetic Resonance, 125*: 34–42, 1997.

[9] **Zimmerman D. E., Kulikowski C. A., Huang Y., Feng W., Tashiro M., Shimotakahara S. … Montelione G.T.** (1997). Automated analysis of protein NMR assignments using methods from artificial intelligence. *Journal of Molecular Biology, 269*: 592–610.

[10] **Andrec M. and Levy R. M.** (2002). Protein sequential resonance assignments by combinatorial enumeration using $^{13}C\alpha$ chemical shifts and their (i, i–1) sequential connectivities. *Journal of Biomolecular NMR, 23*: 263–270, 2002.

[11] **Atreya H. S., Sahu S. C., Chary K. V. R. and Govil G.** (2000). A tracked approach for automated NMR assignments in proteins (TATAPRO). *Journal of Biomolecular NMR, 17*: 125–136.

[12] **Gronwald W., Willard L., Jellard T., Boyko R.F., Rajarathnam K., Wishart D. S, Sönnichsen F. D. and Sykes B. D.** (1998). CAMRA: Chemical shift based computer aided protein NMR assignments. *Journal of Biomolecular NMR, 12*: 395–405, 1998.

[13] **Hyberts S. G. and Wagner G.** (2003). IBIS—a tool for automated sequential assignment of protein spectra from triple resonance experiments. *Journal of Biomolecular NMR, 26*: 335–344.

[14] **Zimmerman D., Kulikowski C. A., Wang L., Lyons B. A. and Montelione G. T.** (1994). Automated sequencing of amino acid spin systems in proteins using multidimensional HCC(CO)NH-TOCSY spectroscopy and constraint propagation methods from artificial intelligence. *Journal of Biomolecular NMR, 4*: 241–256.

[15] **Bartels C., Güntert P., Billeter M. and Wüthrich K.** (1996). GARANT—a general algorithm for resonance assignment of multidimensional nuclear magnetic resonance spectra. *Journal of Computational Chemistry, 18*: 139–149.

[16] **Schmidt E. and Güntert P.** (2012). A New Algorithm for Reliable and General NMR Resonance Assignment. *Journal of the American Chemical Society. 128*: 13112–13122.

[17] **Moseley H. N. B, Monleon D. and Montelione G. T.** (2001). Automatic determination of protein backbone resonance assignments from triple resonance nuclear magnetic resonance data. *Methods in Enzymology. 339*: 91–108.

[18] **Du D. Z., Pardalos P. M.** (1998). *Handbook of Combinatorial Optimization.* Springer, Boston, MA. isbn: 978-0-7923-5019-4. Doi: https://doi.org /10.1007/978-1-4613-0303-9

[19] **Hoos H. H. and Stützle T.** (2005). *Stochastic Local Search: Foundations and Applications.* Elsevier, San Francisco.

[20] **Selman B., Gomes C.** (2006). Hill-climbing Search Chapter in book *Encyclopedia of Cognitive Science*. Wiley. DOI: 10.1002/0470018860. s00015.

[21] **Glover F., Kochenberger G. A.** (2003). *Handbook of Metaheuristics.* Kluwer Academic Publishers.

[22] **Glover F. and Laguna M.** (1997). *Tabu Search.* Kluwer Academic Publishers.

[23] **Dorigo M. and Di Caro G.** (1999). Ant colony optimization: a new meta-heuristic. *IEEE Xplore*, doi: 10.1109/CEC.1999.782657.

[24] **Holland J. H.** (1992). Adaptation in Natural and Artificial Systems, An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. *A Bradford Book*. isbn: 9780262082136.

[25] **Smith S. F.** (1980). *A Learning system based on genetic adaptive algorithms*, (Ph. D. Thesis), Univ. of Pittsburgh.

[26] **Hillier F. S., Lieberman G. J.** (2010). *Introduction to Operations Research*, McGraw-Hill International Edition. isbn: 978-007-126767-0.

[27] **Goldberg D. E.** (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company.

[28] **Holland J. H.** (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI.

[29] **Deb K.** (2008). *Multi-Objective Optimization using Evolutionary Algorithms.* Wiley. isbn: 978-0-470-74361-4 (P/B).

[30] **Coello Coello C.A., Lamont G.B., Van Veldhuizen D.A.** (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems.* Springer, New York.

[31] **Deb K., Jain H.** (2013). An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation*. *18*:577 – 601, 2013.

[32] **Center for Biomolecular Magnetic Resonance** (BMRZ), Goethe University Frankfurt am Main. Retrieved on 06.02.2021 from http://www.bmrz.de /liquid-state-nmr-instr/950-mhz-bruker-avance-iii.

[33] **Neil E. J.** (2007). *NMR Spectroscopy Explained.* Wiley-Interscience.

[34] **Keeler J.** (2002). *Understanding NMR Spectroscopy*. Wiley.

[35] **F. W. Muskett.** (2011). *Sample Preparation, Data Collection and Processing. Protein NMR Spectroscopy: Practical Techniques and Applications,* 10:5-21. Wiley. doi: https://doi.org/10.1002/9781119972006.ch1.

[36] **Garrett D. S., Powers R., Gronenborn A. M. and Clore G.M.** (2011). A common sense approach to peak picking in two-, three-, and four-dimensional spectra using automatic computer analysis of contour diagrams. *Journal of Magnetic Resonance*, 213:357-363.

[37] **Koradi R., Billeter M., Engeli M., Güntert P. and Wüthrich K.** (1998). Automated Peak Picking and Peak Integration in Macromolecular NMR Spectra Using AUTOPSY. *Journal of Magnetic Resonance, 135*, 288–297.

[38] **Bartels C., Xia T. H., Billeter M., Güntert P. and Wüthrich K.** (1995). The program EASY for computer-supported NMR spectral analysis of biological macromolecules. *Journal of Biomolecular NMR, 6*:1-10.

[39] **Würz J. M. and Güntert P.** (2017). Peak picking multidimensional NMR spectra with the contour geometry based algorithm CYPICK. *Journal of Biomolecular NMR, 67*:63–76.

[40] **Kuszewski J., Schwieters C. D., Garrett D. S., Byrd R. A., Tjandra N. and Clore G. M.** (2004). Completely automated, highly error-tolerant macromolecular structure determination from multidimensional Nuclear Overhauser Enhancement spectra and chemical shift assignments. *Journal of the American Chemical Society, 126*:6258-6273.

[41] **Huang Y. J., Tejero R., Powers R. and Montelione G. T.** (2005). A topology constrained distance network algorithm for protein structure determination from NOESY data. *Proteins: Structure, Function, and Bioinformatics, 62*:587-603.

[42] **Herrmann T., Güntert P. and Wüthrich K.** (2002). Protein NMR structure determination with automated NOE assignment using the new software CANDID and the torsion angle dynamics algorithm DYANA. *Journal of Molecular Biology, 319*:209-227.

[43] **Schmidt E.** (2014). *Automated resonance assignment of biomolecular NMR spectra Institute of Biophysical Chemistry.* Goethe University, Frankfurt am Main.

[44] **Güntert P.** (2009). Automated structure determination from NMR spectra. *European Biophysics Journal, 38*:129-143.

[45] **Schwieters C. D., Kuszewski J. J., Tjandra N. and Marius Clore G.** (2003). The Xplor-NIH NMR molecular structure determination package. *Journal of Magnetic Resonance, 160*:65-73.

[46] **Brünger A. T., Adams P. D., Clore G. M., DeLano W. L., Gros P., Grosse Kunstleve R. W. … Warren G. L.** (1998). Crystallography & NMR system: A new software suite for macromolecular structure determination. *Acta Crystallographica Section D, 54*:905-921.

[47] **J. M. Würz, Kazemi S., Schmidt E., Bagaria A. and Güntert P.** (2017). NMR-based automated protein structure determination. *Archives of Biochemistry and Biophysics, 628*:24-32, 2017.

[48] **Schlick T.** (2006). *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer, New York.

[49] **Schmucki R., Yokoyama S. and Güntert P.** (2009). Automated assignment of NMR chemical shifts using peak-particle dynamics simulation with the DYNASSIGN algorithm. *Journal of Biomolecular NMR, 43*:97–109, 2009.

[50] **Bahrami A., Assadi A.H., Markley J.L., Eghbalnia H.R.** (2009). Probabilistic interaction network of evidence algorithm and its application to complete labeling of peak lists from protein NMR spectroscopy. *PLoS Computational Biology, 5*:e1000307, 2009.

[51] **Leutner M., Gschwind R.M., Liermann J., Schwarz C., Gemmecker G., and Kessler H.** (1997). Automated backbone assignment of labeled proteins using the threshold accepting algorithm. *Journal of Biomolecular NMR,, 11*: 31–43, 1997.

[52] **Andrec M. and Levy R.M.** (2002). Protein sequential resonance assignments by combinatorial enumeration using 13Cα chemical shifts and their (i, i–1) sequential connectivities. *Journal of Biomolecular NMR, 23*: 263–270.

[53] **Atreya H. S., Sahu S. C., Chary K. V. R. and Govil G.** (2000). A tracked approach for automated NMR assignments in proteins (TATAPRO). *Journal of Biomolecular NMR, 17*: 125–136.

[54**] Huang Y. P. J., Moseley H. N. B., Baran M. C. , Arrowsmith C., Powers R., Tejero R. … Montelione G.T.** (2005). An integrated platform for automated analysis of protein NMR structures. *Methods in Enzymology, 394*:111–141, 2005.

[55] **Guerry P., Herrmann T.** (2011). Advances in automated NMR protein structure determination. *Quarterly Reviews of Biophysics, 44*:257–309.

[56] **Eghbalnia H. R., Bahrami A., Wang L. Y., Assadi A., Markley J. L.** (2005). Probabilistic identification of spin systems and their assignments including coil-helix inference as output (PISTACHIO). *Journal of*

Biomolecular NMR, 32:219–233.

[57] **Li K. B. and Sanctuary B. C.** (1997). Automated resonance assignment of proteins using heteronuclear 3D NMR. 2. Side chain and sequence-specific assignment. *Journal of Chemical Information and Computer Sciences, 37*:467–477.

[58] **Xu J., Sanctuary B. C. and Gray B. N.** (1993) Automated extraction of spin coupling topologies from 2D NMR correlation spectra for protein proton resonance assignment. *J Chem Inf Comput Sci, 33*:475–489.

[59] **Xu J., Straus S. K., Sanctuary B. C. and Trimble L.** (1994) Use of fuzzy mathematics for complete automated assignment of peptide 1H 2D NMR spectra. *Journal of Magnetic Resonance,* B *103*:53–58.

[60] **Jung Y. S., Zweckstetter M.** (2004). Mars - Robust automatic backbone assignment of proteins. *Journal of Biomolecular NMR, 30*:11-23.

[61] **Munkres J.** (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics, 5*:32–38.

[62] **Vosselman G.** (1992). *Relational Matching.* Springer, New York.

[63] **Maden Yilmaz E. and Güntert P.** (2015). NMR structure calculation for all small molecule ligands and nonstandard residues from the PDB Chemical Component Dictionary. *Journal of Biomolecular NMR, 63*:21-37.

[64] **Schmidt E., Ikeya T., Takeda M., Löhr F., Buchner L., Ito Y., Kainosho M. and Güntert P.** (2014). Automated resonance assignment of the 21 kDa stereo-array isotope labeled thioldisulfide oxidoreductase DsbA. *Journal of Magnetic Resonance\ 249*:88–93.

[65] **Schmidt E., Gath J., Habenstein B., Ravotti F., Székely K., Huber M. … Güntert P.** (2013). Automated solid-state NMR resonance assignment of protein microcrystals and amyloids. *Journal of Biomolecular NMR, 56*:243–254, 2013

[66] **Schmidt E. and Güntert P.** (2013). Reliability of exclusively NOESY-based automated resonance assignment and structure determination of proteins. *Journal of Biomolecular NMR, 57*:193–204, 2013.

[67] **Aeschbacher T., Schmidt E., Blatter M., Maris C., Duss O., Güntert P. … Schubert M.** (2013). Automated and assisted RNA resonance assignment using NMR chemical shift statistics. *Nucleic Acids Research\ 41*:e172.

[68] **Krähenbühl B., El Bakkali I., Schmidt E., Güntert P. and Wider G.** (2014). Automated NMR resonance assignment strategy for RNA via the phosphodiester backbone based on high-dimensional through-bond APSY experiments. *Journal of Biomolecular NMR, 59*:87–93.

[69] **Hadka D. and others.** (2009- 2019) A Free and Open Source Java Framework for Multi-objective Optimization, Retrieved on 2018 – 2020, from http://moeaframework.org/.

[70] **Deb K., Pratap A., Agarwal, S., and Meyarivan T.** (2000). A fast elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on*

Evolutionary Computation.

[71] **Deb K. and Jain H.** (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation.*

[72] **Güntert P. and Kazemi S.** (2017). *Automated Assignment and Structure Calculation with CYANA.* Lecture EMBO Workshop Basel.

[73] **Güntert P, Mumenthaler C. and Wüthrich K.** (1997). Torsion angle dynamics for NMR structure calculation with the new program DYANA. *Journal of Molecular Biology*, *273*:283–298.

[74] **Güntert P.** (2003). Automated NMR protein structure calculation with CYANA. *Progress in Nuclear Magnetic Resonance Spectroscopy, 43*:105–125.

[75] **Lee W., Bahrami A., Dashti H. T., Eghbalnia H., Tonelli M., Westler W. M. and Markley J. L.** (2019). I-PINE web server: an integrative probabilistic NMR assignment system for proteins. *Journal of Biomolecular NMR, 73*:213-222.

[76] **Scott A., Pantoja-Uceda D., Koshiba S., Inoue M., Kigawa T., Terada T. … Güntert, P.** (2005). Solution structure of the Src homology 2 domain from the human feline sarcoma oncogene Fes. *Journal of Biomolecular NMR, 31(4)*:357-61.

[77] **Scott A., Pantoja-Uceda D., Koshiba S., Inoue M., Kigawa T., Terada T. … Güntert P.** (2004). NMR assignment of the SH2 domain from the human feline sarcoma oncogene FES. *Journal of Biomolecular NMR, 30*, 463–464.

[78] **Scott A., López-Méndez B. and Güntert P.** (2006). Fully automated structure determinations of the Fes SH2 domain using different sets of NMR spectra. *Magnetic Resonance in Chemistry*, *44*, S83–S88.

[79] **Clark A.T., McCrary B.S., Edmondson S.P. and Shriver J.W.** (2004) Thermodynamics of core hydrophobicity and packing in the hyperthermophile proteins Sac7d and Sso7d. *Biochemistry, 43*, 2840-2853.

[80] **Yamazaki T. et al.** (1998) Segmental isotope labeling for protein NMR using peptide splicing. *Journal of the American Chemical Society*, *120*, 5591–5592.

[81] **Wider G. and Wüthrich K.** (1999). NMR spectroscopy of large molecules and multimolecular assemblies in solution. *Current Opinion in Structural Biology*, *9*, 594–601.

[82] **Iwai H., Züger S., Jin J. and Tam P.H.** (2006). Highly efficient protein trans-splicing by a naturally split DnaE intein from Nostoc punctiforme. *FEBS Lett, 580,* 1853–1858.

APPENDICES

APPENDIX A: The graphical representation of assignments done by the (a) GA, (b) FLYA, and (c) PINE algorithms on the Fes SH2 protein.
APPENDIX B: Chemical shift assignment results on the 2nd and 4th fragments of the Fes SH2 protein with different iteration numbers of NSGA2, NSGA3 and GA algorithms.
APPENDIX C: The graphical representation of assignments done by the (a) GA, (b) FLYA, and (c) PINE algorithms on Sso7d protein.

APPENDIX A

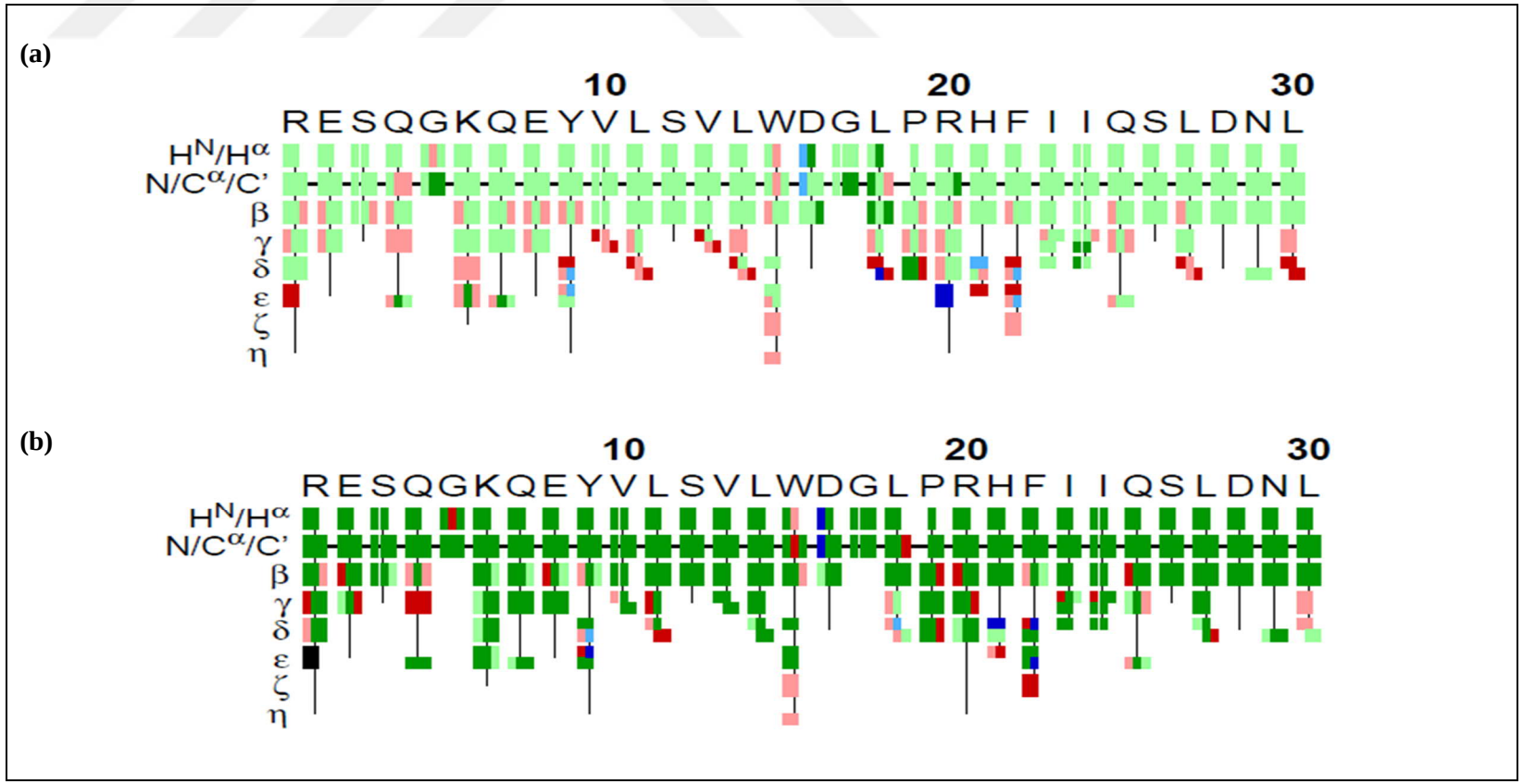

Figure A.1: The graphical representation of assignments done by the (a) GA, (b) FLYA, and (c) PINE algorithms on the Fes SH2 protein.

(c)

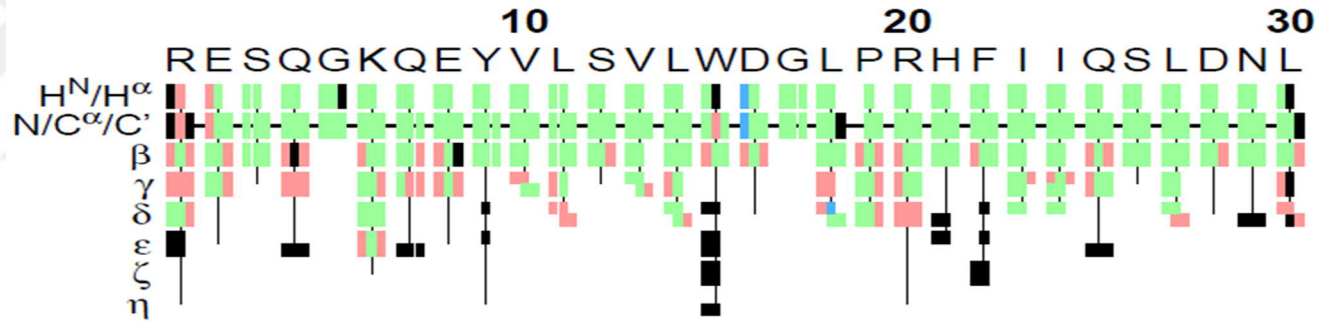

Figure A.1 (continued): The graphical representation of assignments done by the (a) GA, (b) FLYA, and (c) PINE algorithms on the Fes SH2 protein.

APPENDIX B

Table B.1 : Chemical shift assignment results on the 2nd and 4th fragments of the Fes SH2 protein with different iteration numbers of NSGA2, NSGA3 and GA algorithms.

Fragment	Class	FLYA	PINE	10,000 Iterations			100,000 Iterations			1,000,000 Iterations		
				NSGA2	NSGA3	GA	NSGA2	NSGA3	GA	NSGA2	NSGA3	GA
2	Correct	287 (85%)	218 (64%)	249 (74%)	27 (8%)	228 (67%)	238 (70%)	24 (7%)	216 (64%)	244 (72%)	26 (8%)	221 (85%)
	Wrong	30 (9%)	80 (24%)	89 (26%)	311 (92%)	110 (33%)	100 (30%)	314 (93%)	122 (36%)	94 (28%)	312 (92%)	117 (35%)
	Unassigned	21 (6%)	40 (12%)	0	0	0	0	0	0	0	0	0
	Run time (h)	0.9	0.1	0.6	0.7	0.6	1.0	1.3	1.1	6.1	7.5	6.1
4	Correct	287 (79%)	204 (56%)	252 (69%)	31 (8%)	237 (65%)	247 (68%)	19 (5%)	237 (65%)	246 (67%)	22 (6%)	233 (64%)
	Wrong	26 (7%)	106 (29%)	113 (31%)	334 (92%)	128 (35%)	118 (32%)	346 (95%)	128 (35%)	119 (33%)	344 (94%)	132 (36%)
	Unassigned	52 (14%)	55 (15%)	0	0	0	0	0	0	0	0	0
	Run time (h)	1.0	0.1	0.5	0.6	0.5	0.9	1.2	1.2	7.2	7.9	6.4

APPENDIX C

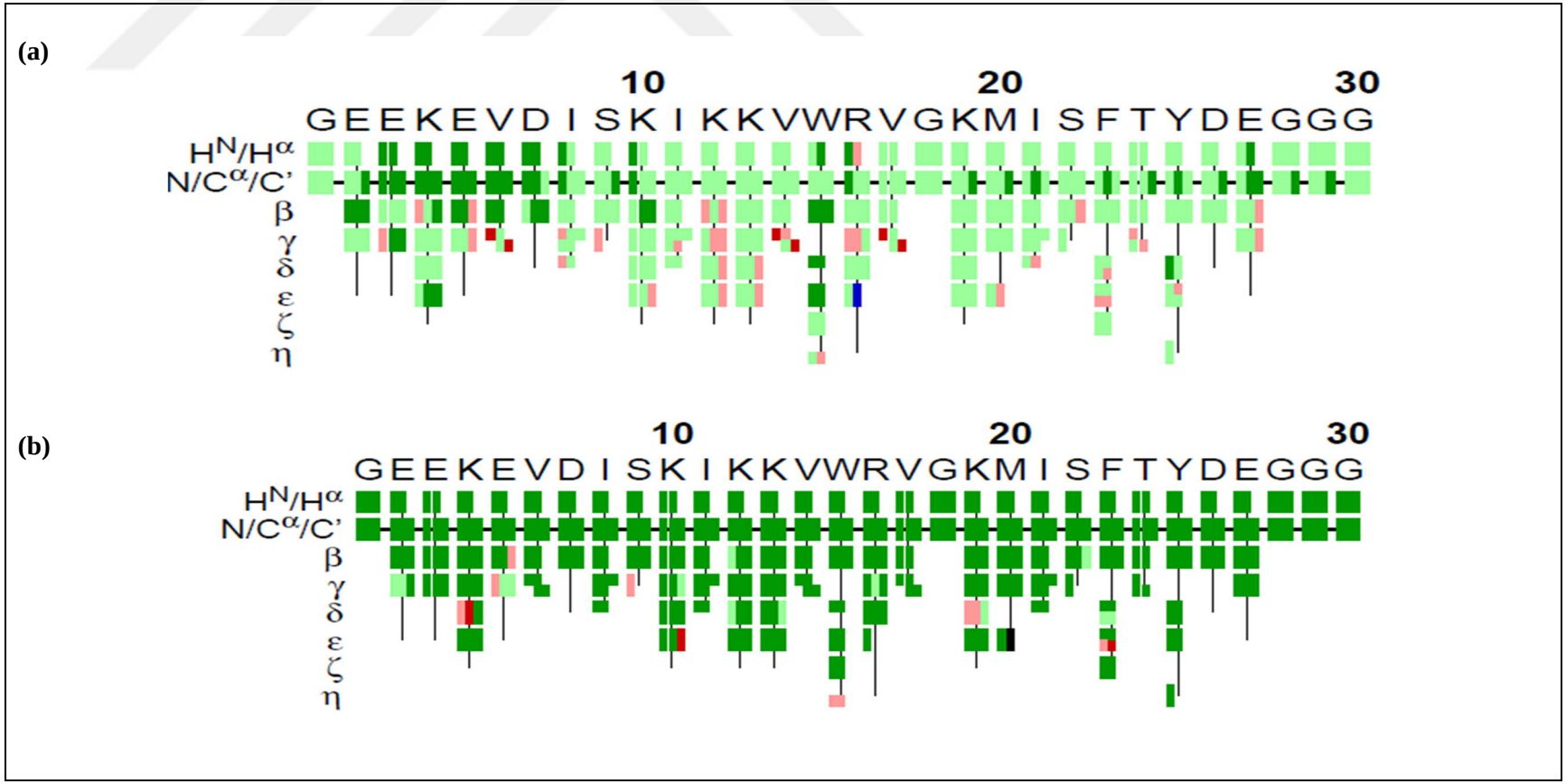

Figure C.1 : The graphical representation of assignments done by the (a) GA, (b) FLYA, and (c) PINE algorithms on the Sso7d protein.

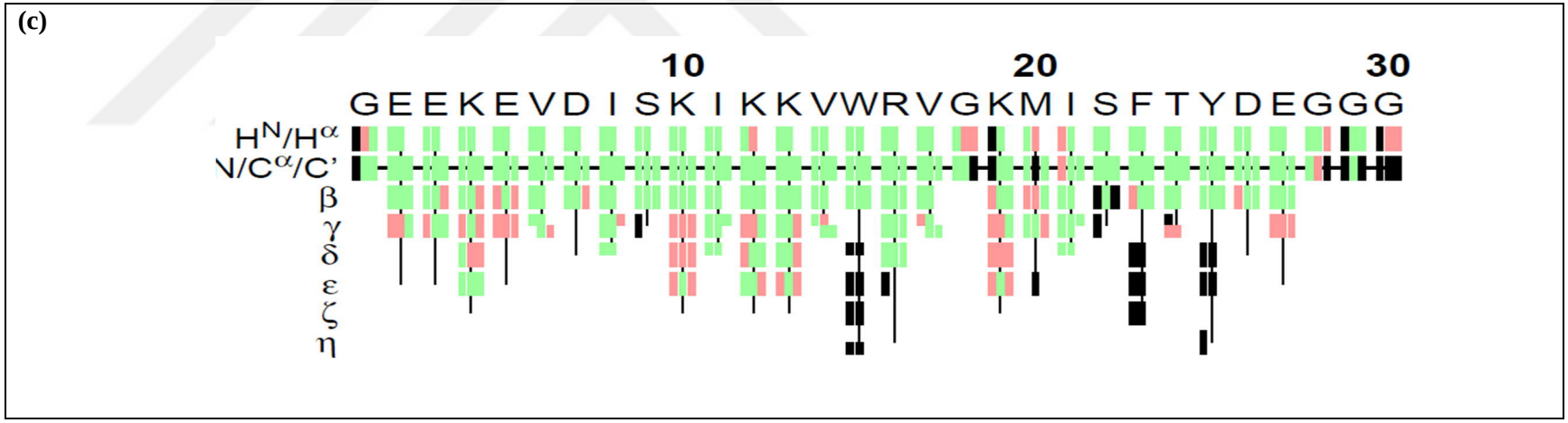

Figure C.1 (continued): The graphical representation of assignments done by the (a) GA, (b) FLYA, and (c) PINE algorithms on the Sso7d protein.

CURRICULUM VITAE

Name Surname **:** Emel MADEN YILMAZ

EDUCATION **:**

- **M.Sc.** **:** 2009, Goethe University Frankfurt, Department of Physics, Computational Science Programme

- **B.Sc.** **:** 2007, Istanbul Technical University, Faculty of Electrical and Electronics, Department of Computer Engineering
  (Erasmus Exchange: 2006- 2007, Technical University of Munich, Department of Informatics)

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2020 – Present, Unit Manager at Mercedes Benz Otomotiv Ticaret ve Hizmetler A.Ş., İstanbul, Turkey.
- 2017 – 2020, Project Manager at Mercedes Benz Türk A.Ş., İstanbul, Turkey.
- 2015 – 2017, Project Manager at Innova IT Solutions, İstanbul, Turkey.
- 2011 – 2015, Test Manager and Subproject Lead at Bosch Siemens Home Appliances, İstanbul, Turkey.
- 2010 – 2011, Technical Lead at Telefónica O2 Germany GmbH & Co. OHG, Munich, Germany.
- 2009, Master of Science graduation from Goethe University, Frankfurt Germany with GPA 1.5/ 5.0 (best score: 1.0).
- 2008 – 2009, Scientific Researcher at Goethe University, Frankfurt am Main, Germany.
- 2007 – 2009, Software Engineer at Goethe Graduate Academy, Goethe University, Frankfurt am Main, Germany.

- 2007, Software Engineer at Telefónica O2 Germany GmbH & Co. OHG, Munich, Germany.
- 2006 – 2007, Erasmus Award of European Union and selection for the Technical University of Munich.
- 2002 – 2005, Success Scholarship from Istanbul Technical University.
- 2003 – 2005, Dean's Honor, Istanbul Technical University.
- 2002, Honor certificate graduation from Trabzon High School of Science.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Maden Yilmaz, E.**, Güntert, P. and Etaner-Uyar, A. Ş., 2021. Evaluation of multi-objective optimization algorithms for NMR chemical shift assignment, *Molecules.* 26(12), 3699, doi: https://doi.org/10.3390/molecules26123699.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Maden Yilmaz, E.** and Güntert, P., 2015. NMR structure calculation for all small molecule ligands and nonstandard residues from the PDB Chemical Component Dictionary, *Journal of Biomolecular NMR.* 63:21-37, doi: 10.1007/s10858-015-9959-y.