An Ant Colony Optimization Approach for Solving the Nuclear Magnetic Resonance Structure Based Assignment Problem

Jeyhun Aslanov
Istanbul Technical University
Istanbul, Turkey
jaslanov@itu.edu.tr

Bülent Çatay
Sabanci University
Istanbul, Turkey
catay@sabanciuniv.edu

Mehmet Serkan Apaydın
Sabanci University
Istanbul Sehir University
Istanbul, Turkey
apaydin@sehir.edu.tr

ABSTRACT

Nuclear Magnetic Resonance (NMR) Spectroscopy is an important technique that allows determining protein structure in solution. An important problem in protein structure determination using NMR spectroscopy is the mapping of peaks to corresponding amino acids. Structure Based Assignment (SBA) is an approach to solve this problem using a template structure that is homologous to the target. Our previously developed approach NVR-BIP computed the optimal solution for small proteins, but was unable to solve the assignments of large proteins. NVR-TS extended the applicability of the NVR approach for such proteins, however the accuracies varied significantly from run to run.

In this paper, we propose NVR-ACO, an Ant Colony Optimization (ACO) based approach to this problem. NVR-ACO is similar to other ACO algorithms in a way that it also consists of three phases: the construction phase, an optional local search phase and a pheromone update phase. But it has some important differences from other ACO algorithms in terms of solution construction and pheromone update functions and convergence rules. We studied the data set used in NVR-BIP and NVR-TS. Our new method finds optimal solutions for small proteins and achieves perfect assignment on EIN and higher accuracy on MBP compared to NVR-TS. It is also more robust.

*Abbreviations used: NMR, Nuclear Magnetic Resonance; NOE, Nuclear Overhauser Effect; RDC, Residual Dipolar Coupling; SBA, Structure-Based Assignments; NVR, Nuclear Vector Replacement; BIP, Binary Integer Programming; TS, Tabu Search; ACO, Ant Colony Optimization

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.

Categories and Subject Descriptors

Applied Computing [**Life and medical sciences**]: Computational biology

General Terms

Algorithms

Keywords

NMR; ant colony optimization; backbone resonance assignments; N15-labeled

1. INTRODUCTION

To understand the function of a protein it is often necessary to determine its 3D structure. There are two main techniques for determining the structure of a protein: X-Ray Crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy. X-ray Crystallography is a method of determining the arrangement of atoms within a crystal. NMR is an experimental technique that exploits the magnetic properties of certain atomic nuclei to obtain information about the geometry of the atoms and the bonds between them. These structures are then deposited into the Protein Data Bank [6] where they are available for download. In contrast to X-ray Crystallography, NMR spectroscopy is usually limited to proteins smaller than 35 kDa, although larger structures have been solved. NMR spectroscopy is often the only way to obtain high resolution information on partially or wholly intrinsically unstructured proteins. Not all proteins can be crystallized and studied by X-ray Crystallography. Moreover, NMR allows one to solve protein structure in solution.

The key challenge in NMR spectroscopy is mapping NMR peaks to the atoms. Testing all combinations is intractable and this problem is still solved manually in many NMR laboratories, which may take months to complete. Another complicating factor in solving the assignments is the noise in the data. Due to noise, the peaks may overlap and in addition there may be extra or missing peaks, which complicates the automated solution of the assignments.

Structure Based Assignment (SBA) is a method to solve this problem by using a template structure that is homologous to the target protein. This template provides prior structural information about the target protein and leads to faster resonance assignments. Most of the novel proteins

have structural homologues in Protein Data Bank which enable the application of SBA in determining their structure. SBA has applications in drug design, function prediction and SAR [9].

Previous SBA studies include combinatorial assignment program (CAP) ([1], [15]) which is an RNA assignment algorithm and which performs an exhaustive search over all permutations with an exponential time complexity. Integer Linear Programming-based Assignment (IPASS) [2] is an integer linear programming-based assignment method on perfect as well as noisy peak lists. These approaches as well as MARS [17] require triple resonance experiments. In contrast, Nuclear Vector Replacement (NVR) [21] does not require triple resonance experiments and instead relies on data that requires less spectrometer time and is therefore less expensive to acquire. NVR is a molecular replacement-like approach for SBA of resonances and sparse Nuclear Overhauser Effects (NOEs). NVR-BIP [4] is a binary integer programming formulation of SBA in NVR framework. It computes the exact solution of an optimization problem subject to NOE constraints and obtains high assignment accuracies on small proteins. NVR-TS [5] is a tabu search approach to solve the optimization problem introduced by [4] which has a guided diversification mechanism. NVR-TS also relaxes NOE constraints by using a penalty term. By doing so, it enables searching through infeasible neighbourhoods violating the NOE constraints to reach better solutions. NVR-TS also solved the assignments for large proteins.

In this work we develop an ant colony optimization (ACO) approach to solve the assignment problem introduced in [4]. ACO is a metaheuristic approach developed for solving hard combinatorial optimization problems using the foraging behaviour of ants. Ant system (AS) is the first ACO approach which was applied for solving the travelling salesman problem ([12]). Some early applications include the elitist strategy for ant system (EAS) proposed by [12], rank-based version of ant system (AS_{rank}) by [7], MAX-MIN ant system (MMAS) by [27], and ant colony system (ACS) by [11]. ACO algorithms have also been applied to bioinformatics problems such as protein folding ([23], [24], [25]) and flexible ligand-protein docking [18]. More details on ACO and an extensive review of its applications to various combinatorial optimization problems may be found in [13]. The contributions of this paper are as follows:

- We eliminate the manual step in NVR-BIP and NVR-TS of updating the alignment tensor for the Residual Dipolar Couplings (RDCs) based on previously obtained assignments, thus automating the computation.
- We implement an ACO based algorithm to solve the NMR SBA problem under the NVR framework. To the best of our knowledge this is the first application of ACO to this problem.
- We apply our approach to NVR-TS dataset. Our results demonstrate higher assignment accuracies and more robust results with less computational time.

The remainder of this paper is structured as follows. In Section 2, we give a brief description of NVR-Framework and give definition of the problem. Our algorithm for the NMR SBA problem is described in Section 3. The results and comparison of the algorithm with other approaches is presented in Section 4. In Section 5 we draw conclusion and note some directions for future research.

2. NVR-FRAMEWORK

NVR is a molecular replacement-like approach for SBA of resonances and sparse NOEs. NVR-Expectation-Maximization (NVR-EM) [19] is a polynomial time algorithm that uses maximum bipartite matching in an expectation maximization framework to assign a protein using information from a structural homologue. One set of nodes in NVR-EM's bipartite graph corresponds to the peaks and the other corresponds to the residues. The edges are associated with a weight which corresponds to the probability of assigning that edge. These probabilities form the basis of NVR's scoring function and are computed by using the difference in the backcomputed and observed NMR data, such as RDCs. NVR-EM performs the assignments in two stages: In the first phase, the assignments are performed using only chemical shifts. After five unambiguous assignments are made, the alignment tensor is computed and the RDCs are also added to the computation. The alignment tensor is updated as more assignments are made. NVR-EM has been successfully demonstrated on 3 target proteins with 21 protein templates.

NVR uses only ^{15}N labeled data. NVR does not require triple resonance experiments unlike most other assignment programs, but relies on a few cheap key spectra. These data include chemical shifts which allow to identify individual atoms, Nuclear Overhauser Effect (NOE) data which allows to determine the pair of protons close in space, and Residual Dipolar Couplings (RDCs) which provide global information on the orientation of internuclear vectors. NVR framework has been extended to also accept ^{15}N TOCSY and amide exchange HSQC by [20]. These data types provide side chain proton chemical shifts and solvent accessibility information of the labile protons, respectively, and allow the determination of the amino acid type as well as whether the proton is exposed to the solvent or not, respectively. This resulted in improved assignment accuracies for distant templates of target proteins [3].

NMR SBA problem consists of finding the correspondence between the peaks in the NMR spectrum and the amino acids with the help of a template protein. In the NVR framework, this problem is reduced to minimizing a scoring function that reflects the likelihood of assigning individual peaks to amino acids subject to distance (NOE) constraints. More formally the problem can be formulated as follows [4]:

Notation:
P — set of peaks
A — set of amino acids
s_{ij} — combined score of assigning peak i to amino acid j
NOE_i — set of peaks that have an NOE constraint with peak i
NTH — distance threshold for NOE interaction
$b_{ij} \in \{0, 1\}$ — equals 1 if the distance between amino acids j and l exceeds a distance threshold (NTH); 0 otherwise

$$Z = \min \sum_{i \in P} \sum_{j \in A} s_{ij} x_{ij} \quad (1)$$

$$\sum_{i \in P} x_{ij} = 1, \ \forall j \in A \quad (2)$$

$$\sum_{j \in A} x_{ij} = 1, \ \forall i \in P \quad (3)$$

$$x_{ij} + x_{kl} \leq b_{jl} \ \forall j, l \in A, \ \forall i, k \in P, \ \forall k \in NOE_i \quad (4)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in P, \ \forall j \in A \quad (5)$$

The objective function (1) minimizes the total score associated with assigning NMR peaks to amino acids. Constraints (2) ensure that each amino acid is matched with exactly one peak and constraints (3) make sure that each peak is assigned to exactly one amino acid. The NOE constraints (4) make sure that the distance between the protons corresponding to the NOE is expected to be less than the predetermined threshold. Constraints (5) define the binary variables such that x_{ij} is equal to 1 if peak i is assigned to amino acid j and 0 otherwise.

Finding the set of assignments with the minimum total assignment score within the NVR framework is proven to be NP-hard by using a reduction from the Three Coloring Problem [5].

3. ALGORITHM

Our approach consists of applying ant colony optimization to the SBA problem and the automatic alignment tensor computation. ACO consists of solution construction, local search and pheromone update. These are described in Sections 3.1 and 3.2.

3.1 Ant Colony Optimization Based Approach (NVR-ACO)

ACO is based on the observation of the behaviour of real ant colonies searching for food sources. Real ants deposit an aromatic essence, called pheromone, on the path they walk. Other ants searching for food can sense the existence of pheromone and choose their way according to the level of pheromone. Greater level of pheromone on a path will increase the probability of the ants following that path. The level of pheromone laid on a path is based on the length of the path and the quality of the food source and increases when the number of ants following that path increases. In time all ants are expected to follow the shortest path [14].

ACO simulates the above described behaviour of real ants to solve combinatorial optimization problems with artificial ants. Artificial ants find solutions on a graph in parallel processes using a constructive mechanism guided by artificial pheromone trails and a greedy heuristic known as visibility [10]. Pheromone trail intensity τ_{ij} between nodes i and j represents the collective memory of ants and its amount is proportional to the quality of the solution generated. The visibility η_{ij} is the heuristic information showing the desirability of moving from node i to node j and can be implemented in different ways depending on the particular problem. In addition, the artificial ants may benefit from a local search heuristic in an attempt to improve the solution quality.

Similar to other ACO algorithms, NVR-ACO consists of three phases: a construction phase during which the ants build the solutions, a local search phase where the solutions are further improved, and a pheromone update phase where the pheromone trails are updated based on the quality of the solutions obtained after the construction and the local search phases. The algorithm is summarized in Figure 1.

```
initialize pheromone trails
while (stopping condition not satisfied) do
    for all ant do
        for i = 1 → |P| do
            select a peak (using constrained peak selection)
            assign amino acid (using random selection rule)
        end for
    end for
    elitist (2-opt) local search
    elitist pheromone update
end while
```

Figure 1: Outline of NVR-ACO

3.1.1 Solution Construction

In our approach we assign a pheromone value τ_{ij} to every peak-amino acid matching and we associate each ant with a peak. So, the number of ants is equal to the number of peaks. Initially, ant i is placed on peak i. From this position each ant begins constructing its solution by selecting at each step an amino acid using the random selection rule. During this construction process, we allow both the assignments which violate NOE constraints and the assignment of peak-amino acid pairs where a large difference exists between the predicted and experimental data values such as chemical shifts. This is in contrast to NVR-BIP where such assignments were considered as "infeasible". However, due to noise in the experimental data such assignments need to be tolerated in order to obtain a robust solution for the SBA problem. In order to minimize the violation of NOE constraints and the number of "infeasible" assignments we penalize such assignments. The total score associated with the solution obtained by ant k is calculated as follows:

$$Z^k = \sum_{i \in P_1} \sum_{j \in A} s_{ij} x_{ij} + \sum_{i \in P_2} p, \quad (6)$$

Here P is the set of peaks, P_2 is the set of penalized peaks (corresponding to NOE violations or "infeasible" assignments), and $P_1 = P \setminus P_2$. s_{ij} is the score associated with assigning peak i to amino acid j and p is the penalty term which is calculated as follows:

$$p = \max(s_{ij}) * p_c \quad (7)$$

where $p_c > 0$ is a constant used for controlling the impact of the penalty term. Note that each NOE violation is penalized in NVR-TS whereas NVR-ACO includes a penalty per peak. Our aim in doing so is to prevent overpenalizing an incorrect peak assignment with many NOE violations.

As opposed to most ACO algorithms, our heuristic information consists of two components. Since our objective is to minimize the total score of assigning peaks to amino acids our first heuristic information is $\eta_{ij} = 1/s_{ij}$. The second heuristic function is as follows:

$$\delta_{ij} = \begin{cases} 1 & \text{if an assignment does not violate} \\ & \text{any NOE conditions} \\ \frac{1}{c*p} & \text{otherwise} \end{cases} \quad (8)$$

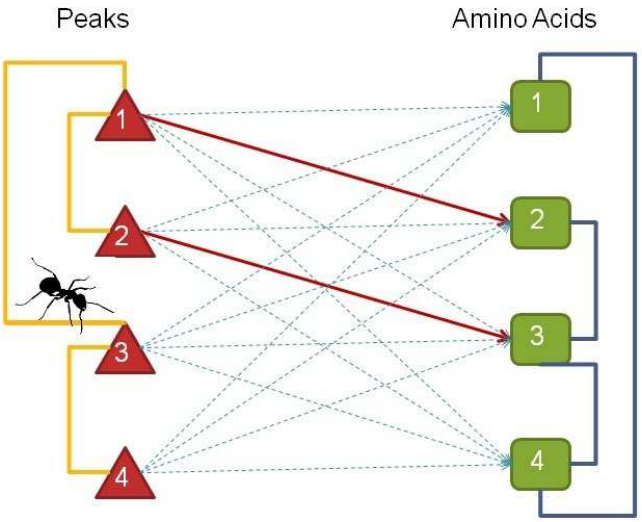

Figure 2: A graphic view of assignment process. Yellow lines between peaks represent NOE relations, blue lines between amino acids represent distance relations.

c is the number of NOE violations caused by assigning peak i to amino acid j. The motivation of this heuristic function is to lower the probability of an assignment which violates more NOE conditions. The random selection rule is as follows:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta \delta_{ij}^\gamma}{\Sigma_{l \in N_i^k} \tau_{il}^\alpha \eta_{il}^\beta \delta_{il}^\gamma} & \text{if } j \in N_i^k \\ 0 & \text{otherwise} \end{cases}, \quad (9)$$

where N_i^k is the set of unassigned amino acids for the k^{th} ant and α, β, and γ are the parameters used to control the influence of pheromone trails and the heuristic information.

After matching an amino acid with the incumbent peak the ant selects the next peak to assign to an amino acid. The peak selection procedure is as follows: Peaks are attributed with a number showing how many amino acids it can be matched with and whether this assignment will violate NOE conditions or not. Then, the peaks are sorted in the non-decreasing order of this value and a peak is randomly selected among the top θ_1 percent of all peaks. We refer to this pseudo-random selection procedure as "constrained peak selection". We also tested random ($\theta_1 = 100$) and sequential peak selection procedures (select the next peak from the peak array) but their performance were inferior.

3.1.2 *Local Search*

Before we update the pheromone trails we use a 2-opt local search procedure to further improve the assignments obtained by the ants. We adopt an elitist approach where the solutions obtained by the best-performing θ_2 percent of ants are involved in the local search using an exhaustive best-improvement strategy. Basically, we swap the assignments leading to the greatest decrease in the total score value and repeat this procedure until no further decrease is possible.

3.1.3 *Pheromone Update*

At the beginning pheromone trails are initialized as follows:

$$\tau_0 = 1/[(1-\rho) * Z^0] \quad (10)$$

where Z^0 is an initial score value. In our implementation we estimated Z^0 by multiplying the mean value of the score matrix with the number of peaks.

The pheromone evaporation is performed as usual by reducing the phero-mone trails by the evaporation factor ρ. For the pheromone reinforcement we use an elitist strategy based on relative weights inspired from [25]. In this approach we sort the ants according to the quality of solutions they have achieved. The pheromone values are updated according to the following formula:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{r=1}^{l} \omega_r \Delta\tau_{ij}^r, \ \forall (i,j) \quad (11)$$

where $(0 < \rho < 1)$ is the evaporation rate, l is the number of elite ants, and ω_r is the weight associated with ant r. We set $l = |P| * \theta_3 \ /100$ where θ_3 is a parameter to determine the number of elite ants and $\omega_r = Z^l/Z^r$. Z^l and Z^r are the scores of the constructed solution for l^{th} and r^{th} ant, respectively. Since we have a minimization objective function the weight of the best solution found so far is the highest within this scheme. The use of relative weight with respect to the best-score prevents over-emphasizing the solutions of the best-performing ants. It also enables better convergence of pheromone trails. $\Delta\tau_{ij}^r$ indicates the amount of pheromone deposited by ant r and is calculated as follows:

$$\Delta\tau_{ij}^r = \begin{cases} 1/Z^r, & \text{if amino acid } j \text{ is assigned} \\ & \text{to peak } i \text{ by ant } r \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

When the system stagnates we re-initialize the pheromone trails by setting Z^0 in equation (10) equal to the best-so-far score. The system is considered in stagnation if the ratio of standard deviation to the mean of the whole pheromone matrix is smaller than ε for I_1 consecutive iterations and the best-so-far score has not improved in the past I_2 consecutive iterations. Once a feasible assignment has been obtained if the algorithm is unable to improve it after I_3 consecutive stagnations we terminate.

3.2 Automating the assignment computation

In our previous approach (NVR-BIP and NVR-TS) we performed the assignments in two stages: first we computed the assignments without RDCs, then based on these assignments we computed the Saupe alignment tensor and added RDCs to the score matrix computation. We then computed the assignments with the new scoring matrix. The computation involved iterating between assignment computation and generating the new scoring matrix until convergence of the assignments and each step required running a separate program manually with the right set of input files.

In this paper we automate this process by incorporating the RDCs at the outset of the assignments and closing the loop between the alignment tensor computation and the assignment computation, iterating until convergence. In particular, we use grid search as in [21] to compute the alignment tensor without requiring a priori assignments, and then at each successive step we update the alignment tensor as before using singular value decomposition (SVD).

Table 1: Experimental design for parameter selection (bold values are selected)

Parameters	Tested Values
α - pheromone intensity parameter	**1**, 2, 3, 4
β - visibility parameter	0, 1, 2, **3**
γ - visibility parameter	**1**, 2, 3
θ_1 - % of peaks used in constrained peak selection	5, **10**, 15
θ_2 - % of ants used in local search	2, 5, 10, **15**, 20, 30
θ_3 - % of ants used in pheromone update	1, 2, **5**, 10, 15
ρ - evaporation constant	0.01, 0.02, **0.05**, 0.10, 0.15, 0.20
ε - constant for stagnation	**0.0001**

Table 2: Accuracy results for large proteins. Columns 3-6 are in percentages.

PDB ID	No. of residues	NVR-TS		NVR-ACO	
		Best Sol.	Avg. Acc.	Best Sol.	Avg. Acc.
EIN	243	83	59	100	96
MBP	348	91	89	91	90

4. RESULTS

We tested NVR-ACO on an Intel® Xeon® CPU E7430 machine with 8 2.13GHz processors each with 128GB total memory. We studied the data set used in NVR-BIP and NVR-TS. Detailed data preparation for the tested proteins is as described in [4], [5] and [19]. After some preliminary tests, we determined the values of I_1 , I_2 , and I_3 as 5, 20 and 5, respectively, and p_c is set to 10. To tune the remaining parameters we performed a preliminary experimental study using the values summarized in Table 1. The numbers in bold show the values we selected to use in our detailed experiments.

We performed ten runs for each protein. The results are reported in Tables 2 and 3. Accuracy is the ratio of correctly assigned peaks to the total number of peaks. "Best Sol." column refers to the accuracy of the best solution (i.e. the solution with the lowest score), achieved in ten runs and "Avg. Acc." column shows the average accuracy of the ten solutions.

Table 2 summarizes the results on the following large proteins: MBP (348 residues) and EIN (243 residues). While we were unable to solve these proteins to optimality due to their sizes in NVR-BIP, NVR-TS was able to find a solution with an accuracy of 91% for MBP and 83% for EIN. The average accuracies were 89% and 59%, respectively. NVR-ACO has a similar performance for MBP: the best solution has an accuracy of 91% while the average accuracy is 90%. On the other hand, it has a remarkable performance for EIN with 100% accuracy in the best solution and an average accuracy of 96%. These results show both the superiority and the robustness of NVR-ACO on these two large proteins. Note that even though the hardware used to obtain the results for NVR-TS and NVR-ACO are different, the experimental conditions are the same, enabling us to compare the accuracies.

Similar to NVR-TS and NVR-BIP, NVR-ACO also finds the optimal solutions for smaller proteins. The assignment accuracies are slightly higher for 1AZF and 1BGI and slighly lower for 3LYZ compared to NVR-TS. These are probably due to the small difference in the initial alignment tensor estimation method. More significantly, NVR-ACO is more robust compared to NVR-TS and it is able to find the optimal solution in most tests.

With NVR-ACO, it takes a few minutes to obtain the assignments for pol η, GB1, SPG, ff2, ubiquitin and about 10-13 minutes for hSRI and lysozyme respectively. For the larger proteins, it takes 8.5 hours for EIN and 20 hours for MBP to obtain the final assignments. In contrast, NVR-BIP running time varied between a few seconds to about 30 minutes on its test set and NVR-TS returned a solution in shorter amount of time than NVR-BIP on NVR-BIP's dataset. It took NVR-TS an average of 37 hours to solve the assignments for MBP and 19.8 hours for EIN on an Intel® Core i7 CPU 960 3.20 Ghz processor with 24GB of RAM.

Finally, it is not practical to compare the assignment accuracy of NVR-ACO with that of other approaches such as that of [26] since the combination of NMR data used in these approaches (and sometimes the accuracy measures) are different. Nevertheless, on EIN, [26] has an assignment accuracy of 99.2% with a relaxed assignment definition which accepts an assignment as correct if its assignment ensemble contains the correct matching, whereas NVR-ACO achieves 100% with a more strict accuracy definition which counts an assignment as correct only if a peak is assigned to the correct residue.

5. CONCLUSION AND FUTURE WORK

In this paper we developed NVR-ACO, an ant algorithm for solving the NMR protein structure-based assignment problem. To the best of our knowledge, this is the first application of ACO to the NMR-SBA problem and it includes the following new mechanisms: a heuristic for peak selection, two visibility functions, and a relative weight approach for pheromone reinforcement. We tested NVR-ACO on NVR-BIP and NVR-TS's dataset. We found the optimal solutions on NVR-BIP's dataset and were able to solve the assignments for the large proteins with a higher accuracy than NVR-TS. We have also automated our computational procedure.

As future work we plan to develop a better score function (for example by using Bayesian statistics), extracting more information from data. Future work also includes automating peak picking and chemical shift referencing as well as

Table 3: Accuracy results for small proteins. The numbers in columns 4-8 are in percentages.

Protein Family	No. of residues	PDB ID	NVR-BIP	NVR-TS		NVR-ACO	
				Best Sol.	Avg. Acc.	Best Sol.	Avg. Acc.
Ubiquitin	72	1UBI	97	97	97	97	97
		1UBQ	97	97	97	97	97
		1G6J	97	97	97	97	97
		1UD7	97	97	97	97	97
		1AAR	97	97	97	97	97
SPG	55	1GB1	100	100	100	100	100
		2GB1	100	100	100	100	100
		1PGB	100	100	100	100	100
Lysozyme	126	193L	100	100	100	100	100
		1AKI	98	98	98	98	98
		1AZF	94	94	94	95	95
		1BGI	97	97	97	98	98
		1H87	100	100	100	100	100
		1LSC	100	100	100	100	100
		1LSE	98	98	98	98	98
		1LYZ	82^a	$82^a, 69^b$	$82^a, 69^b$	$82^a, 79^b$	$82^a, 79^b$
		2LYZ	91	91	92	91	91
		3LYZ	90	90	89	88	88
		4LYZ	91	91	91	91	91
		5LYZ	91	91	89	91	91
		6LYZ	96	96	96	96	96
The Rest	80	ff2	93	93	93	93	93
	96	hSRI	89	89	89	89	89
	31	pol η	100	100	100	100	100
	55	GB1	100	100	100	100	100

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

performing the assignments without TOCSY data and with ambiguous NOEs.

6. ACKNOWLEDGEMENT

This work was supported by following grants to M.S.A.: The Scientific and Technical Research Council of Turkey research support program (program code 1001) [109E027] and EU Marie Curie Grant PIRG05-GA-2009-249267.

7. REFERENCES

[1] H. Al-Hashimi, A. Gorin, A. Majumdar, Y. Gosser, and D.J. Patel. Towards structural genomics of RNA: rapid NMR resonance assignment and simultaneous RNA tertiary structure determination using residual dipolar couplings. *Journal of Molecular Biology*, 318(3):637–649, May 2002.

[2] B. Alipanahi, X. Gao, E. Karakoc, F. Balbach, L. Donaldson, C. Arrowsmith, and M. Li. IPASS: Error tolerant NMR backbone resonance assignment by linear programming. Technical Report CS-2009-16, University of Waterloo, 2009.

[3] M. S. Apaydin, V. Conitzer, and B. R. Donald. Structure-based protein NMR assignments using native structural ensembles. *Journal of Biomolecular NMR*, 40(4):263–276, April 2008.

[4] M.S. Apaydın, B. Çatay, N. Patrick, and B.R. Donald. NVR-BIP: Nuclear vector replacement using binary integer programming for NMR structure-based assignments. *The Computer Journal*, 54(5):708–716, May 2011.

[5] G. Çavuşlar, B. Çatay, and M.S. Apaydın. A tabu search approach for the NMR protein structure-based assignment problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(6):1621–1628, 2012.

[6] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res*, 28:235–242, 2000.

[7] B. Bullnheimer, R.F. Hartl, and C. Strauss. A new rank-based version of the ant system: A computational study. *Central European Journal of Operations Research and Economics*, 7:25–38, 1999.

[8] G.M. Crippen, A. Rousaki, Zhang Y. Revington, M., and E.R.P. Zuiderweg. SAGA: rapid automatic mainchain nmr assignments for large proteins. *Journal of Biomolecular NMR*, 46(4):281–298, 2010.

[9] B. R. Donald and J. Martin. Automated NMR assignment and protein structure determination using sparse dipolar coupling constraints. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 55(2):101–127, August 2009.

[10] M. Dorigo. Ant colony optimization. *Scholarpedia*, 2(3):1461, 2007.

[11] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. In *IEEE Transactions on Evolutionary Computation*, 1997.

[12] M. Dorigo, V. Maniezzo, and A. Colorni. The ant

system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B 26: 2941*, 26(1):29–41, 1996.

[13] M. Dorigo and T. Stützle. *Ant colony optimization. A Bradford Book*. The MIT Press, 2004.

[14] S. Goss, R. Beckers, J.L. Deneubourg, S. Aron, and J.M. Pasteels. *How trail laying and trail following can solve foraging problems for ant colonies*. In: Behavioral Mechanisms of Food Selection, Ed. R.N. Hughes. NATO-ASI Series, vol. G 20, Springer-Verlag: Berlin, 1990. 661-678.

[15] J. Hus, J. Prompers, and R. Brushweiler. Assignment strategy for proteins with known structure. *Journal of Magnetic Resonance*, 157:119–123, 2002.

[16] R. Jang, X. Gao, and M. Li. Integer programming model for automated structure-based NMR assignment. Technical Report CS-2009-32, University of Waterloo, 2009.

[17] Y. Jung and M. Zweckstetter. Mars: robust automatic backbone assignment of proteins. *J. Biomol. NMR*, 30:11–23, 2004.

[18] O. Korb, T. Stützle, and T.E. Exner. An ant colony optimization approach to flexible protein-ligand docking. *Swarm Intelligence*, 1(2):115–134, 2007.

[19] C. Langmead and B. Donald. An expectation/maximization nuclear vector replacement algorithm for automated NMR resonance assignments. *J. Biomol. NMR*, 29:111–138, 2004.

[20] C. Langmead and B. Donald. High-throughput 3D structural homology detection via NMR resonance assignment. In *Proc. IEEE Computational Systems Bioinformatics Conf.*, pages 278–289, 2004.

[21] C. Langmead, A. Yan, R. Lilien, L. Wang, and B. Donald. A polynomial-time nuclear vector replacement algorithm for automated NMR resonance assignments. In *Proc. the 7th Annual Int. Conf. Research in Computational Molecular Biology (RECOMB)*, pages 176–18, Berlin, Germany, April 10-13 2003.

[22] J. Meiler and D. Baker. Rapid protein fold determination using unassigned NMR data. In *Proc. Natl Acad. Sci. USA*, 2003.

[23] A. Shmygelska, R.A. Hernandez, and H.H. Hoos. An ant colony optimization algorithm for the 2D HP protein folding problem. In *ANTS '02 Proceedings of the Third International Workshop on Ant Algorithms*, pages 400–417. Springer-Verlag, 2002.

[24] A. Shmygelska and H.H. Hoos. An improved ant colony optimization algorithm for the 2D and HP protein folding problem. Lecture Notes in Computer Science, 2003. Volume 2671/2003, 993.

[25] A. Shmygelska and H.H. Hoos. An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC Bioinformatics Journal*, 2005.

[26] D. Stratmann, E. Guittet, and Heijenoort van C. Robust structure-based resonance assignment for functional protein studies by NMR. *Journal of Molecular Biology*, 46:157–173, 2010.

[27] T. Stützle and H. H. Hoos. The MAX-MIN ant system and local search for the travelling salesman problem. In T. Bäck, Z. Michalewicz, and X. Yao, editors, *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, pages 309–314. IEEE Press: Piscataway, NJ, 1997.