Automated Bond Order Assignment as an Optimization Problem

Anna Katharina Dehof, Alexander Rurainski, Hans-Peter Lenhof, Andreas Hildebrandt

Center for Bioinformatics, Saarland University, 66041 Saarbrücken, Germany

anne@bioinf.uni-sb.de

Abstract: Numerous applications in Computational Biology process molecular structures and hence require not only reliable atomic cordinates, but also correct bond order information. Regrettably, this information is not always provided in molecular databases like the Cambridge Structural Database or the Protein Data Bank. Very different strategies have been applied to derive bond order information, most of them relying on the correctness of the atom coordinates. We extended a different ansatz proposed by Wang et al. that assigns heuristic molecular penalty scores solely based on connectivity information and tries to heuristically approximate its optimum. In this work, we present two efficient and exact solvers for the problem replacing the heuristic approximation scheme of the original approach: an ILP formulation and an A* approach. Both are integrated into the upcoming version of the Biochemical Algorithms Library BALL and have been successfully validated on the MMFF94 validation suite.

1 Introduction

Correct bond order information is essential for many algorithms in Computational Structural Biology and Chemistry, since bonds do not only define the connectivity of atoms in a molecule but also define structural aspects like rotatability of individual groups. However, bond order information can often not be directly infered from the available experimental data. Even important molecular databases, like the Protein Data Bank (PDB) [BHN03] and the Cambridge Structural Database [All02], are known to contain erroneous data for connectivity and bond order information [Lab05] or to even omit them entirely. For proteins and nucleic acids, bond orders can be easily deduced due to their building block nature, but this does not hold for other kinds of molecules like ligands. The problem is made much worse by the fact that quite often, the bond order assignment for a given molecule is not unique, even when neglecting symmetries in the molecule. The chemical reasons for this effect are complex and out of scope of this work; here we just want to state that the concept of integer bond orders is only an approximation to a full quantum chemical treatment, and cannot explain all effects occurring in molecules. Important examples are aromatic or delocalized bonds, leading to different resonance structures (c.f. Fig. 1). In addition, formal charges are often not contained in the input files, but atoms carrying a formal charge will obviously show a different bonding pattern. One body of opinion tries to overcome these

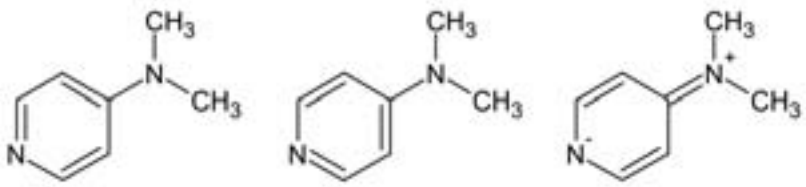

Figure 1: Different resonance structures of 4-(N,N-dimethylamino)pyridine. A bond order assignment program should optimally be able to compute all of these configurations.

obstacles by hand curation which clearly provides the highest reliability. On the other hand, manual data curation does not scale well to large numbers of molecules, and it does not help in conditions where modifications are systematically applied to molecules, e.g. in computational combinatorial chemistry.

In the past decades, the problem of assigning bond orders automatically has been addressed by a number of different approaches. Early methods in the field strongly rely on the correctness of atomic coordinates and focus on reference bond lengths and valence angles [BH92], or additionally consider functional group detection [HRB97] and further molecular features like hybridization states and charges [vABF$^+$96, ZCW07]. The main drawbacks of those approaches are the dependence on correct atomic coordinates and their heuristic nature.

In constrast, exact solvers proposed previously represent the bond order assignment problem as a Maximum Weighted Matching for nonbipartite graphs [Lab05] or as an integer linear programming problem that generates valid Lewis structures (electron dot structures) with minimal formal charge on each atom [FH05].

Recently, Wang et al. [WWKC06] have presented an elegant novel approach to the problem which is implemented in the established Antechamber package, a suite of tools used for the preparation of input structures for molecular mechanics studies. In this approach, a chemically motivated, expert generated penalty function is used to score bond order assignments. This function is then heuristically optimized. However, this procedure has two drawbacks: the score of resulting assignment is not guaranteed to be optimal and the algorithm provides only one solution while there can be more than one assignment with optimal score. In this work, we propose an approach that solves the problem to provable global optimality by discrete optimization techniques. We give an integer linear program formulation for very efficient computation of one optimal assignment and an A* approach for enumerating all optimal or, if desired, all feasible solutions.

2 Methods

The idea behind the bond order assignment algorithm proposed in the work of [WWKC06] is to cast it into a discrete optimization problem. Finding the most probable consistent bond order assignment for a given molecule is addressed by minimizing a total penalty score tps, where each atom is assigned an atomic valence av that is defined as the sum over all bond orders bo of all bonds connected to the atom under consideration:

$$av = \sum_{i=1}^{con} bo_i$$

Here, con denotes the number of bonded atoms. The distance of the calculated av to the atom's most desirable valence value is measured by the atomic penalty score aps: the possible valences of an atom and the corresponding distance penalty scores are stored in a penalty table that uses a rule-based atom type classification derived by Wang et al. The sum over all atomic penalty scores of a molecule now yields the total penalty score

$$tps = \sum_{i=1}^{n} aps_i$$

where n denotes the number of atoms. The smaller the tps of a given bond order assignment, the more reasonable it is. In [WWKC06], minimization now proceeds in a heuristic and greedy manner.

2.1 Integer Linear Program (ILP)

To compute a bond order assignment with guaranteed globally minimal tps, we formulated the aforementioned problem as an integer linear program [PS98] as described below.

Let P be the penalty table. We use the following notations:

- A is the set of all atoms of the molecule under consideration.
- $B(a)$ is the set of bonds of atom $a \in A$ and B denotes the set of all bonds of the molecule.
- $V(a) \subset \mathbb{N}$ contains the possible valences of atom $a \in A$ according to the penalty table P.
- $P(a, v)$ is the entry of P for atom $a \in A$ and valence $v \in V(a)$.

Our approach uses two different classes of variables. For each bond $b \in B$, we introduce a variable $x_b \in \{1, \ldots, \mu\}$, where μ is the maximum bond order considered (in the following, we will set μ to 3, allowing single, double, and triple bonds). For all atoms a and corresponding possible valences v according to the penalty table P we introduce choice

variables $y_{a,v} \in \{0,1\}$. Each $y_{a,v}$ symbolizes whether the corresponding penalty $P(a,v)$ is chosen or not, i.e., penalty $P(a,v)$ contributes to the score iff $y_{a,v} = 1$. Thus, the objective function of our score minimization problem can be formulated as a linear function in $\mathbf{y}$ with penalty prefactors:

$$\min_{\mathbf{y}} \quad \sum_{a \in A} \sum_{v \in V(a)} P(a,v) \cdot y_{a,v}.$$

To ensure that each atom is assigned exactly one valence state, we add the additional linear constraints

$$\sum_{v \in V(a)} y_{a,v} = 1$$

for all $a \in A$. In addition, we have to ensure that the sum of its bond orders equals its chosen valence. The constraints can be formulated as

$$\sum_{v \in V(a)} y_{a,v} \cdot v = \sum_{b \in B(a)} x_b$$

for all $a \in A$, because the left hand side evaluates to valence v iff $y_{a,v} = 1$.

In summary, the score minimization problem can be formulated as the following integer linear program

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{y}} \quad & \sum_{a \in A} \sum_{v \in V(a)} P(a,v) \cdot y_{a,v} & \\
\text{s.t.} \quad & \sum_{v \in V(a)} y_{a,v} \cdot v = \sum_{b \in B(a)} x_b & \forall a \in A, \\
& \sum_{v \in V(a)} y_{a,v} = 1 & \forall a \in A, \\
& y_{a,v} \in \{0,1\} & \forall a \in A, \forall v \in V(a), \\
& x_b \in \{1,2,3\} & \forall a \in A, \forall b \in B(a).
\end{aligned}$$

For the solution of ILPs to provable global optimality, several strategies can be chosen, like the popular pure branch & bound approaches or branch & cut methods [PS98]. We employed the open source solver lp_solve [BEN] which uses a simplex-algorithm-based branch & bound approach [PS98]. It is interesting to note that the penalties in [WWKC06] can all be expressed as powers of two and as such led to short computation times. Still, the problem itself is NP complete [PS98]. Empirically, however, in many test cases the solution of the relaxed linear program, i.e., the above program without the integrality constraints, has been integral and, hence, a solution of the original problem (obtained without any branching). In other cases, the solution of the linear program has been almost integral, leading to only few branch steps. In principle, ILP solvers can also enumerate all optimal solutions. However, in our experiments we have seen a drastic increase in runtime if more than one solution is computed. Thus, the ILP approach is particularly well suited for obtaining *one* optimal bond order assignment.

2.2 The A* approach

In order to be able to efficiently enumerate *all* feasible solutions – optimal and non-optimal ones – we formulated the bond order total penalty minimization problem as an A* search algorithm. This allows enumeration of all assignments in the order of increasing penalty and hence, for instance, to compare the assignments of all solutions for a given molecule up to a user defined penalty threshold. In addition, such an A* algorithm is simpler to implement, and often easier to extend, than an ILP approach; for instance, it is easily possible to influence the order in which solutions with equal score are computed.

As a combinatorial optimization problem, the bond order assignment problem can be represented by a tree, where each layer stands for one of the decisions that have to be made. In our case, the tree has k layers, where k is the number of bonds that have to be assigned. A node at layer i has μ children, where μ is the number of possible bond orders, typically 3, and each edge is labeled with its corresponding order. Hence, by tracing the path from the root to a node w at layer i, we can determine the values of the first i bonds in this particular partial assignment represented by the node w. Thus, the root node corresponds to a completely unassigned molecule with only unknown bond orders, while the leave nodes correspond to complete bond order assignments. If we only add child nodes if the resulting valence state is valid the leaf nodes correspond to the feasible bond order combinations. In order to discriminate between the different combinations, each leaf is assigned its atomic penalty score.

Visiting all nodes in the tree, the optimal bond order assignment can be found in a brute–force manner with exponential runtime. If, additionally, all intermediate nodes are assigned the atomic penalty score of the partial bond order assignment they represent, a greedy search will yield an assignment with heuristically good (but not necessary optimal) atomic penalty score in linear runtime. It can be shown that, if at each intermediate node more information is provided, finding an optimal solution can be guaranteed with greatly improved expected runtime. This leads to the popular A*-search-algorithm [HNR68], which employs a search heuristic to guide the algorithm in descending the tree. More formally, the algorithm associates with each node w a function $f(w) = g^*(w) + h^*(w)$, where $g^*(w)$ describes the score corresponding to the decisions already made and $h^*(w)$ is the so-called search heuristic. For the purposes of the A*-search algorithm, the search heuristic must be an admissible estimate of the score of the best leaf that can be reached starting from node w and descending further down the tree. Here, admissible means that it needs to be 'optimistic': for all nodes w, the estimated cost $h^*(w)$ may never be greater than the lowest real cost to reach a goal node. Given the additional information provided by h^*, the A*-search algorithm always expands one of the nodes with the most promising score, ensuring that the first leaf reached is optimal (roughly speaking, if the algorithm would visit a leaf with worse score first, the search-heuristic would have overestimated the penalty of the real optimal solution, which an admissible heuristic never does).

In addition to the notations introduced in the previous section, we need notations that are adapted to the partial bond order assignments corresponding to each node w in the search tree. We denote the set of all assigned bonds in the node w by $W(B)$, the assigned bonds connected to atom a in node w by $W(a)$, and the set of atoms for which all bonds are

already assigned with a bond order by K. The bond order of an assigned bond is denoted by $bo(b)$. A partial bond order assignment induces a simple lower bound

$$v_w(a) := \sum_{b \in W(a)} bo(b)$$

for the valence of atom a. Assuming a single bond for each unassigned bond of atom a, a tighter lower bound for the valence is given by

$$lo(a) := v_w(a) + \sum_{b \in B(a) \backslash W(a)} 1 = v_w(a) + |B(a) \backslash W(a)|.$$

Thus, the maximum order of an unassigned bond with respect to atom a is given by

$$t(a) := \max\{V(a)\} - lo(a) + 1.$$

Denoting by a_1, a_2 the atoms connected by an unassigned bond b, its maximum bond order equals

$$bo_{max}(b) := \min\{t(a_1), t(a_2)\},$$

yielding an upper bound of the atomic valence of an atom a

$$up(a) := \min \left\{ \max\{V(a)\}, \quad v_w(a) + \sum_{b \in B(a) \backslash W(a)} bo_{max}(b) \right\}.$$

The functions g^* and h^* can then be defined as follows:

$$g^* = \sum_{a \in K} P\left(a, v_w(a)\right) \tag{1}$$

$$h^* = \sum_{a \in A \backslash K} \min_{lo(a) \leq i \leq up(a)} \left\{ P\left(a, i\right) \right\}. \tag{2}$$

The function g^* sums the atomic penalties of all completely assigned atoms in the partial bond order assignment represented by node w, whereas h^* considers all atoms with bonds of unassigned bond order. For the atoms in this set, we compute the minimal atomic penalty possible under the current partial assignment independently of the other atoms in the set: each atom can choose its preferred value for each unassigned bond without considering overall consistency. Obviously, h^* is optimistic.

3 Results

We have implemented and integrated both approaches in the Biochemical Algorithms Library BALL (http://www.ball-project.org, [KL00]). For validating our algorithms, we

molecule	score		number of optimal solutions
	Antechamber	BALL	
DAKCEX.mol2	1	0	2
GETFIU.mol2	1	0	1
GIDMEL.mol2	2	1	7
KEWJIF.mol2	4	0	1
SAFKAL.mol2	1	0	1
JECYIZ.mol2	4	0	1

Table 1: Comparison of the penalties for molecules of the MMFF94 validation suite, where BALL found bond order assignments with smaller penalty score than the assignment heuristically computed by Antechamber.

method	reference is		no solution
	1st solution	optimal	
Antechamber	282 (37.05%)	282 (37.05%)	18 (2.36%)
ILP	401 (52.69%)	401 (52.69%)	4 (0.53%)
A*	473 (62.15%)	599 (78.71%)	4 (0.53%)

Table 2: Performance of the original Antechamber implementation, our ILP formulation and our A*-search algorithm on the MMFF94 validation suite. The second column denotes the number of molecules for which the algorithms return the original bond order assignment as first solution. The third column denotes the number of cases, where the reference bond order assignment was within the solutions with minimal tps (if this is not the case, we need to change the objective function rather than the optimization method to correctly address this molecule). Finally, the fourth column denotes the number of molecules for which no solution was found.

chose to compare the computed results on the MMFF94 validation suite [Hal96]. The MMFF94 Suite contains 761 thoroughly prepared drug like molecules that were originally used for the validation of the Merck Molecular Force Field. We used the penalty table as defined in Wang et al. [WWKC06]. On this data set, A* and ILP had comparable runtimes if generating single solutions only (≈220 seconds for the whole set on a standard PC, where the majority of the time is spent in SMARTS matching).

As can be seen in Tab. 2, both of our methods are able to correctly reproduce significantly more molecules of the MMFF94 validation suite than the original Antechamber approach by Wang et al. In cases where the reference molecule is the only possible assignment with minimal tps, ILP and A* both find the optimal bond order assignment, whereas Antechamber returns non-optimal solutions in 6 cases as shown in Tab. 1.

The difference between the performance of ILP and A*-search are due to fact that the MMFF94 validation suite contains 348 molecules with more than one optimal bond order assignment (with respect to the penalty table of Wang et al.) and that the ILP solver systematically prefers assignments different to the A*-search algorithm. The A*-search always prefers lower bond orders which seems to be the more natural behaviour.

As can also be seen in Tab. 2, the enumeration of all optimal solutions leads to a success rate of 78.71% in reproducing the bond order assignments of the MMFF94 validation suite.

However, it should be kept in mind that in reality, bond order assignment for a single molecule need not have a unique solution; for instance, molecules like benzene show several resonance structures, differing only in their bond order configuration (if aromatic bonds are 'kekulized', i. e. replaced by a compatible pattern of single and double bonds, as needed for most force fields).

Obviously, the quality of the penalty table, e.g., the definition of the atom classes, their allowed valence states, and the choice of the valence state's penalties have a significant influence on the performance of our algorithms. As can be seen in column four of Tab. 2, the current penalty table does not cover all molecules in the MMFF94 validation suite – for four molecules, the required atom classes are missing. Please note that the difference to the Antechamber bailing out rate is a result of the heuristic nature of the optimization proposed in [WWKC06].

4 Conclusion

In this work, we have presented two exact solvers for the connectivity based bond order assignment problem posed by Wang et al. [WWKC06]. Both methods improve considerably upon earlier approximate solution schemes by guaranteeing optimality while retaining highly efficient runtimes.

Our ILP-formulation allows for very rapid computation of an optimal bond order assignment with respect to the underlying penalty tables. In our implementation, the ILP is solved directly by the open source solver lp_solve [BEN]. This approach scales well with increasing number of atoms and bonds and should be preferred if only one optimal assignment is sought. However, when computing more than one solution with the ILP solver, runtimes greatly deteriorated.

In these cases, our A*-approach usually has much better runtime, in particular when enumerating all solutions – optimal and non-optimal ones sorted by their score. In addition, the order in which solutions are returned can be easily influenced. Thus, it has the potential to create ensembles of putative bond order assignments, opening new avenues for probabilistic structure analysis. Furthermore, the A*-search algorithm is simple to implement and independent of external solvers.

So far, only connectivity based information is scored in the search heuristic. The inclusion of structural properties like bond lengths and angles might help to further distinguish between assignments if atomic coordinates are reliable. For large molecules, the employment of more sophisticated optimization techniques as presented in [BBST09] might help to speed up computation times.

Both approachs are fully integrated into the upcoming version of the Biochemical Algorithms Library BALL (http://www.ball-project.org, [KL00]) that can be downloaded from our homepage.

References

[All02] F. H. Allen. The Cambridge Structural Database: a quarter of a million crystal structures and rising. *Acta Crystallogr B*, 58(Pt 3 Pt 1):380–388, Jun 2002.

[BBST09] S. Böcker, Q. B. A. Bui, P. Seeber, and A. Truss. Computing Bond Types in Molecule Graphs. In *Proc. of Computing and Combinatorics Conference (COCOON 2009)*, 2009. To be presented.

[BEN] M. Berkelaar, K. Eikland, and P. Notebaert. lp_solve 5.5. http://lpsolve.sourceforge.net/.

[BH92] J. C. Baber and E. E. Hodgkin. Automatic assignment of chemical connectivity to organic molecules in the Cambridge structural databa. *J Chem Inform Comput Sci*, 32:401–406, 1992.

[BHN03] H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide Protein Data Bank. *Nat Struct Biol*, 10(12):980, Dec 2003.

[FH05] M. Froeyen and P. Herdewijn. Correct bond order assignment in a molecular framework using integer linear programming with application to molecules where only non-hydrogen atom coordinates are available. *J Chem Inf Model*, 45(5):1267–1274, 2005.

[Hal96] T.A. Halgren. MMFF VI. MMFF94s option for energy minimization studies. *J Comp Chem*, 17:490–519, 1996.

[HNR68] P. E. Hart, N. J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, 4:100–107, 1968.

[HRB97] M. Hendlich, F. Rippmann, and G. Barnickel. BALI: Automatic assignment of bond and atom types for protein ligands in the Brookhaven Protein Databank. *J Chem Inform Comput Sci*, 37:774–778, 1997.

[KL00] O. Kohlbacher and H. P. Lenhof. BALL–rapid software prototyping in computational molecular biology. Biochemicals Algorithms Library. *Bioinformatics*, 16(9):815–824, Sep 2000.

[Lab05] P. Labute. On the perception of molecules from 3D atomic coordinates. *J Chem Inf Model*, 45(2):215–221, 2005.

[PS98] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Courier Dover Publications, 1998.

[vABF+96] D. M. van Aalten, R. Bywater, J. B. Findlay, M. Hendlich, R. W. Hooft, and G. Vriend. PRODRG, a program for generating molecular topologies and unique molecular descriptors from coordinates of small molecules. *J Comput Aided Mol Des*, 10(3):255–262, Jun 1996.

[WWKC06] J. Wang, W. Wang, P. A. Kollman, and D. A. Case. Automatic atom type and bond type perception in molecular mechanical calculations. *J Mol Graph Model*, 25(2):247–260, Oct 2006.

[ZCW07] Y. Zhao, T. Cheng, and R. Wang. Automatic perception of organic molecules based on essential structural information. *J Chem Inf Model*, 47(4):1379–1385, 2007.