A Branch–&–Cut Algorithm for the Optimal Solution of the Side Chain Placement Problem

E. Althaus, O. Kohlbacher, H.-P. Lenhof,
P. Müller

MPI–I–2000–1-001 March 2000

Author's Address

E. Althaus, O. Kohlbacher, H.-P. Lenhof, P. Müller
Max-Planck-Institut für Informatik
Im Stadtwald
66123 Saarbrücken
{althaus|oliver|len|pmueller}@mpi-sb.mpg.de

Acknowledgements

This work was supported by the DFG research cluster "Informatikmethoden zur Analyse und Interpretation großer genomischer Datenmengen", grant LE952/2-1.

Abstract

Rigid–body docking approaches are not sufficient to predict the structure of a protein complex from the unbound (native) structures of the two proteins. Accounting for side chain flexibility is an important step towards fully flexible protein docking. This work describes an approach that allows conformational flexibility for the side chains while keeping the protein backbone rigid. Starting from candidates created by a rigid–docking algorithm, we demangle the side chains of the docking site, thus creating reasonable approximations of the true complex structure. These structures are ranked with respect to the binding free energy. We present two new techniques for side chain demangling. Both approaches are based on a discrete representation of the side chain conformational space by the use of a rotamer library. This leads to a combinatorial optimization problem. For the solution of this problem we propose a fast heuristic approach and an exact, albeit slower, method that uses branch–&–cut techniques. As a test set we use the unbound structures of three proteases and the corresponding protein inhibitors. For each of the examples, the highest–ranking conformation produced was a good approximation of the true complex structure.

Contents

1 Introduction

Numerous methodologies have been developed to predict the structure of a protein complex from the structures of its single components. The majority of these techniques are rigid–body approaches, *i.e.*, the two proteins A and B are considered to be completely rigid. Based on this assumption, several algorithms have been proposed [35, 38, 7, 20, 18, 3, 13, 2, 29, 12, 26]. These methods usually succeed in the correct prediction of a protein complex structure if the bound structures are sufficiently close to the unbound ("native") structures. But even small changes in the conformations of a few side chains cause rigid body approaches to fail inevitably. Studies of the conformational changes between bound and unbound conformations [32] show that in most cases the protein backbone remains rigid and the problem reduces to the prediction of the side chain conformations of the protein–protein interface.

Side chain placement has been discussed in depth not only for the *ab initio* prediction of protein structures and homology modeling, but also for the protein docking problem [37, 39]. There is a wide range of techniques available for side chain placement. Due to the high dimensionality of the conformational space of all side chains, exhaustive search is only tractable for a very small number of side chains [6, 39]. Other approaches use Monte Carlo methods [1, 14], local homology modeling [23], or self–consistent field theory [21] to place the side chains.

Ponder and Richards [33] observed that the conformations occuring in proteins can be adequately described by a rather small set of so-called rotamers for each amino acid. These rotamer libraries can be used to reduce the side chain placement problem to a combinatorial optimization problem: search for the set of rotamers with the minimum energy, *i.e.* the global minimum energy conformation (GMEC).

As the number of rotamer combinations is very high (typical binding sites include 40–60 side chains, yielding up to 10^{60} combinations), efficient methods are required to identify the GMEC or suboptimal solutions sufficiently close to the GMEC. One of these methods is based on the dead–end elimination (DEE) theorem by Desmet *et al.* [10].

Methods for side chain demangling were also successfully applied to protein docking. Totrov and Abagyan [37] used a Monte Carlo approach to predict the structure of a lysozyme-antibody complex. Weng *et al. [39] optimized the side chains in the binding site of three protease-inhibitor complexes using exhaustive search. Jackson* et al. [16] employed a self-consistent mean field approach and a Langevin dipole model for the side chain placement in several protein complexes.

An algorithm for ligand docking with flexible side chains has been published by Leach [24, 25]. This approach uses Dead End Elimination and the A* algorithm to place the side chains of the receptor molecule.

We present new techniques for protein interface refinement and their application to the docking of unbound structures. Using a rigid docking algorithm [26, 27], we create a set of 60 candidates for the complex structure. This set usually contains several approximations of the true complex structure. For each of these candidates we determine the side chains in the protein interface and assign a set of rotamers from a rotamer library to each side chain. In order to find a good approximation of the minimum energy conformation of the protein interface, we have to solve the GMEC problem for this set of rotamers, *i.e.*, we have to find the set of rotamers with the minimum energy for the complex.

We assume that the total energy of the complex may be decomposed into contributions from the template (the complex without the residues of the binding site), contributions from the single rotamers, and contributions arising from the pairwise interaction of different rotamers. First, all these energies are calculated using a simple Molecular Mechanics force field. Then, the DEE theorem is iteratively applied. This theorem provides a simple criterion for the exclusion of rotamers that cannot be part of the GMEC. This reduces the number of rotamer combinations by several orders of magnitude.

We have developed and implemented two techniques to solve the GMEC problem. The first approach is based on the so-called multi–greedy (MG) strategy. This technique has been introduced by Moon and Howe [30] and was successfully applied to protein-ligand docking by Rarey *et al.* [34]. It constructs a tree of possible rotamer combinations incrementally, avoiding combinatorial explosion by limiting the number of nodes in each layer of the tree. If the number of nodes is sufficiently large, the best solutions in the tree have energies close to that of the GMEC.

The second approach is a branch–&–cut algorithm. We formulated the GMEC problem as an Integer Linear Program (ILP) and investigated the structure of the GMEC polytope, which is defined as the convex hull of all feasible solutions. We were able to identify some classes of facet-defining inequalities and we devised a separation algorithm for a subclass of inequalities. This approach can also be transferred to related minimization problems with a similar energetic decomposition.

The conformations generated by these techniques are then subjected to a geometry optimization with a Molecular Mechanics force field (side chains only). Finally, the binding free energy of the optimized structure is estimated using an approach proposed by Jackson and Sternberg [17].

We tested our approach on the unbound conformations of three protease-inhibitor complexes from the PDB [4]: 1TPO/4PTI, 5CHA/2OVO, and 1SBC/2CI2. For each of these examples, the unbound structures as well as the complex structure have been determined, rendering them an ideal test set. The MG approach was able to predict the correct complex structure for all examples considered. In every case, the structure ranked as number one was a good approximation of the true complex conformation. The branch–&–cut algorithm was used in the example of 1TPO/4PTI to confirm that the solutions calculated using the MG approach are close to the global minimum. The separation between true positives and false positives still requires improvement and we hope to solve this problem using more detailed rotamer libraries and an improved energetic evaluation.

In Section 2 we describe our techniques in detail. Then, we present the results of our experiments with three protease–inhibitor complexes in Section 3. A discussion of our results concludes this work in Section 4.

2 The docking algorithm

The input of the rigid docking consists of two proteins A and B. These proteins are subjected to a rigid docking, yielding a set of candidate conformations. For the 60 best candidates out of this set, we perform a side chain demangling. Finally, the candidates are scored according to their binding free energy.

2.1 Rigid docking

A rigid docking is performed for the proteins A and B, using the algorithm described in [26] and [27]. The algorithm uses geometric complementarity and simple chemical fitness functions to create a large set of potential complex structures. Out of this set, the 60 best candidates with respect to geometric complementarity are selected.

2.2 Side chain demangling

The rigid docking algorithm, which generates a set of promising candidates, neglects the side chain flexibility. Therefore, many of the candidates have strong overlaps and incorrectly placed side chains. In order to obtain physically meaningful conformations, we have to demangle the side chains of the protein interface (the binding site).

Determination of the binding site First, we have to identify all residues of A and B that belong to the binding site. We consider a residue to be part of the binding site if any of its atoms is within 6 Å of any atom of the other protein. All residues that fulfill this condition are marked and kept in a list BS. Side chains without rotamers (ALA, GLY, and CYS engaged in disulphide bonds) are excluded from this list. For each residue in this list, we determine its set of possible rotamers from the rotamer library of Dunbrack et al. [11], assuming that the side chain can occur in only one of these conformations. We have now decomposed the protein residues into two disjoint sets: one set comprises all residues that have rotamers and belong to the binding site as defined by our cut–off criterion, and the other set comprises all remaining residues (= the template).

The global minimum energy conformation (GMEC) is now defined as the combination of rotamers yielding the lowest total energy. This entails a combinatorial optimization problem where we must search a huge conformational space for the GMEC. In order to demangle the side chains of the binding site, we have to identify the GMEC or a good approximation thereof. By the following decomposition we can express the total energy of the system as a function of the selected rotamers:

$$E^{total} = E^{tpl} + \sum_i E_{i_r}^{tpl} + \sum_i \sum_{j<i} E_{i_r,j_s}^{pw} \tag{2.1}$$

where E^{tpl} is the potential energy of the template (= system without residues of the binding site), $E_{i_r}^{tpl}$ is the potential energy of side chain i in rotameric state r (short: rotamer i_r) interacting with the atoms of the template, and E_{i_r,j_s}^{pw} is the pairwise potential energy between side chain i in rotameric state r and side chain j in rotameric state s. Since rotamers that heavily overlap with the rigid template cannot be part of the GMEC, we remove all rotamers whose interaction energy with the template is larger than 100 kJ/mol.

For a given set of rotamers, the single energy components of Equation 2.1 can be easily calculated as the computational effort scales at most quadratically in the number of atoms. The difficulty arises in finding the global minimum as this means picking the optimal combination of rotamers for the side chains. Fortunately, the search space can be significantly reduced by the so–called *dead–end elimination (DEE) theorem* [10], which can be stated as follows: if for a pair of rotamers (i_r, i_t),

$$E_{i_r}^{tpl} + \sum_{j \neq i} \min_s E_{i_r,j_s}^{pw} > E_{i_t}^{tpl} + \sum_{j \neq i} \max_s E_{i_t,j_s}^{pw} \tag{2.2}$$

holds, then the rotamer i_r can be safely ignored in the search for the global minimum. Iteratively applying the DEE theorem reduces the number of combinations of rotamers by several orders of magnitude.

Search of minimum energy conformation In order to find an optimal combination of rotamers, we have pursued two alternative approaches: a tree–based multi–greedy (MG) method and a branch–&–cut algorithm based on an ILP (Integer Linear Programming) formulation of our problem. We will first describe the greedy method and then explain the more technical branch–&–cut algorithm.

2.2.1 The multi–greedy method

In this approach, an enumeration tree representing all possible rotamer combinations is built. The tree consists of $k = |BS|$ layers, each representing one side chain. Each path from the root to a leaf represents a possible combination of rotamers. The label of the node in layer i determines which rotamer is selected for side chain i. Every node also possesses an energy label. For the construction, we start out with an artificial root node v, which is given an energy label $E(v) = E^{tpl}$ and a rotamer label $r(v) = null$. For each side chain i, we add a new layer to the tree. This layer is constructed by adding a node for each rotamer of i to each leaf of the previous layer (see Figure 2.1). Each new node x for a rotamer i_s is labeled with the corresponding rotamer ($r(x) = i_s$) and its energy label is set to

$$E(x) = E_{i_s}^{tpl} + E(parent(x)) + \sum_{y \in pred(x)} E_{i_s,r(y)}^{pw},$$

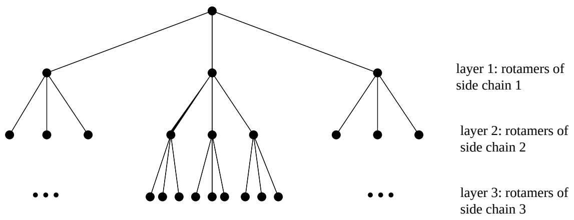

Figure 2.1: The tree for three side chains, each having three rotamers.

where $pred(x)$ denotes the set of nodes that lie on the path from x to v (x excluding) in the tree. Thus, each node is labeled with the sum of the interaction energies of all rotamers on the path from the node to the root.

When the construction of a layer is completed, we sort the list of current leaves according to their energy labels. If the number of leaf nodes exceeds a certain bound `MAX_NODES`, we keep only the best `MAX_NODES` leaves and remove all others. As we keep always at most `MAX_NODES` nodes on each layer, we can be sure that the data structure does not grow exponentially.

Carrying out these steps for all k side chains in the list BS creates a layered tree of height k, with layer i containing only rotamers of side chain i (see Figure 2.1). The energy label of a leaf gives the potential energy resulting from interactions among the side chains and between the side chains and the template for the specific combination of rotamers defined by the rotamer labels on the path. We thus just have to pick the path from the leaf with the lowest energy label to the root in order to obtain the minimum energy conformation of all remaining conformations in the tree.

2.2.2 The branch–&–cut algorithm

The GMEC problem can be formulated as follows: Given a list BS of residue side chains and sets $V_i = \{i_1 \dots i_{n_i}\}$ of possible rotameric states for all $i \in BS$, determine for each side chain i the rotamer i_r, such that the potential energy $E^{total} = E^{tpl} + \sum_i E^{tpl}_{i_r} + \sum_i \sum_{j<i} E^{pw}_{i_r,j_s}$ of the selected conformation is minimized. We assume that all potential energies $E^{tpl}_{i_r}$ and $E^{pw}_{i_r,j_s}$ have already been computed. Let E_{max} be the largest energy value among these potential energies.

We reformulate the problem as a minimization problem on an undirected graph $G = (V, E)$. For each rotamer i_r, we create a node v with weight $E(v) = E^{tpl}_{i_r} - E_{max}$ and for each pair (i_r, j_s) of rotamers with $i \neq j$, we create an edge uv with weight $E(uv) = E^{pw}_{i_r,j_s} - E_{max}$. The resulting graph is k-partite with partitions V_i, $i = 1 \dots k$ where all the nodes and edges have negative weights. The partition V_i is called the ith column of the graph. For a node v of the graph, we define $c(v)$ to be the column of this node. The possible rotamer sets correspond to the subgraphs of G with the following property: Every subgraph consists of exactly k nodes, one node out of each column, and the induced edges. We call these subgraphs rotamer graphs. The weight of a rotamer graph is the sum of the weights of its nodes and edges. Note that the weight of a rotamer graph and the energy of its corresponding rotamer combination differ exactly by $E^{tpl} + (k + \binom{k}{2}) \cdot E_{max}$. Thus, the GMEC problem can be seen as determining the rotamer graph of G with minimal weight.

The Integer Linear Program We now transform this graph–theoretic description into an integer program by introducing a binary decision variable x_v for each node v and x_{uv} for each edge uv. If a node (edge) belongs to the rotamer graph, the value of the variable x_v (x_{uv}) is 1 and otherwise it is 0. For brevity, we say that a node or an edge is selected if the corresponding binary decision variable is 1. The basic constraint system of the GMEC problem is the following:

$$\min \sum_{v\in V} E(v)x_v + \sum_{uv\in E} E(uv)x_{uv}$$

$$s.t. \quad \sum_{v\in V_i} x_v = 1 \quad \text{for all } i \in \{1 \dots k\} \tag{2.3}$$

$$x_{uv} \le x_v \quad \text{for all } uv \in E \tag{2.4}$$

$$x_{uv} \le x_u \quad \text{for all } uv \in E \tag{2.5}$$

$$x_v \in \{0,1\} \quad \text{for all } v \in V \tag{2.6}$$

$$x_{uv} \in \{0,1\} \quad \text{for all } uv \in E \tag{2.7}$$

The constraints (2.3) enforce that exactly one node of each partition, *i.e.*, exactly one rotamer for each side chain, is selected. The constraints (2.4) and (2.5) guarantee that an edge can only be selected if both endpoints are selected, *i.e.*, we consider only pairwise interactions between selected rotamers. Since only one edge from a certain column i to a node v can be selected, we can tighten these inequalities to

$$\sum_{u\in V_i} x_{uv} \;\le\; x_v \text{ for all } v \in V,\, i \ne c(v). \tag{2.8}$$

Note that the above integer linear program has subgraphs of rotamer graphs as feasible solutions. Of course, the weight of a complete rotamer graph is smaller than the weight of any of its subgraphs, so that the optimal solution of the integer linear program is a rotamer graph.

Branch–&–Cut is the most common technique to handle hard combinatorial optimization problems. It works as follows: We relax the integer linear program by dropping the integality condition and solve the resulting linear program. If the solution $\bar{x}$ is integral we have the optimal solution. Otherwise, we search for a valid inequality $fx \le f_0$ that cuts off the solution $\bar{x}$, *i.e.*, $fy \le f_0$ for all feasible solutions y and $f\bar{x} > f_0$; the set $\{x \mid fx = f_0\}$ is called a cutting plane. The search for the cutting plane is called the separation problem. Any cutting plane found is added to the linear program and the linear program is resolved. The generation of cutting planes is repeated until either an optimal solution is found or the search for the cutting plane fails. In the second case a branch step follows: We generate two subproblems by setting one fractional variable x_v (x_{uv}) to 0 in the first subproblem and to 1 in the second subproblem and solve these subproblems recursively. This gives rise to an enumeration tree of subproblems.

For details about branch-&-cut and integer programming and a discussion of improvements on this method, see the book of Wolsey [40].

The GMEC polyhedron We call the convex hull P of all feasible solutions the GMEC polyhedron. Since its facets are the most promising cutting planes, we studied the structure of the GMEC polyhedron. Before we can prove that an inequality defines a facet, we have to determine the dimension of the polyhedron. We assume that the reader is familiar with polyhedral theory; for an introduction, see [31, 40].

Every polyhedron $P \subseteq \mathbf{R}^n$ can be written as $P = \{x \in \mathbf{R}^n \mid Ax \leq d\}$ for some $m \times n$-matrix A and some m-vector d. Note that this description of a polyhedron is not unique. For i, $1 \leq i \leq m$ let A_i be the i-th row of A. If $A_i x = d_i$ for some $x \in \mathbf{R}^n$ the constraint $A_i x \leq d_i$ is called *active* for x. Let M be the set of indices, so that $i \in M$ if and only if $A_i x \leq d_i$ is active for all $x \in P$. Furthermore let $(A^=, d^=)$ be the corresponding rows of (A, d). In other words, $A^= x \leq d^=$ are the constraints of $Ax \leq d$ which hold with equality for all points of the polyhedron.

Lemma 1 $\dim(P) = |V| - k + |E|$

Proof: A fundamental lemma about polyhedral theory (see [31]) states that

$$dim(P) + rank(A^=, d^=) = |V| + |E|,$$

where $rank(A^=, d^=)$ is the number of linearly independent rows of $(A^=, d^=)$. Note that $rank(A^=, d^=)$ is independent of the particular description of P.

Thus $\dim(P) \leq |V| - k + |E|$, because we have k linearly independent equalities in our partial description of the polyhedron. In order to show that $\dim(P) \geq |V| - k + |E|$, we have to show that there are $|V| - k + |E| + 1$ affinely independent feasible solutions.

A *basic solution* b is a feasible solution with $b_{uv} = 0$ for all $uv \in E$. For i, $1 \leq i \leq k$ we define $b(i)$ to be the node v of V_i with $b_v = 1$. For any node v let e_v be the unit vector of this node, *i.e.*, the vector with entry 1 at the position of v and with entry 0 at all other positions. Analogously for any edge uv, let e_{uv} be the unit vector of this edge. For an 0/1-vector x we do not distinguish between the vector and the corresponding subgraph.

Let b be a basic solution. Beside b, we construct the solutions where one node of b is replaced by another one in the same column, and we construct one solution for every edge by including the edge, both its endpoints and for the remaining columns we take the nodes of the basic solution. More formally, we look at the following set S of feasible solutions:

$$\begin{aligned} S = \ & \{b\} \cup \{b + e_v - e_{b(c(v))} \mid \text{ for all } v \in V \text{ with } b_v = 0\} \\ & \cup \{b + e_{uv} + e_u - e_{b(c(u))} + e_v - e_{b(c(v))} \mid \text{ for all } uv \in E\} \end{aligned}$$

The size of the set is $1 + |V| - k + |E|$. Let M be the matrix with rows corresponding to the solutions of S. We have to show that the solutions are affinely independent, *i.e.*, if one subtracts one row of M from the others, the remaining rows are linear independent. We denote the $n \times n$ identity matrix by I_n. We subtract b from all other solutions and get the matrix M' with rows $e_v - e_{b(c(v))}$ for all $v \in V$ and rows $e_{uv} + e_u - e_{b(c(u))} + e_v - e_{b(c(v))}$ for all $uv \in E$. Now we drop all columns i of M' with $b_i = 1$, *i.e.* all columns that have an entry -1. The resulting matrix has the following block-structure

$$\begin{pmatrix} I_{|V|-k} & 0 \\ A & I_{|E|} \end{pmatrix}$$

for some suitable matrix A. This matrix has full rank since it is triangular. Thus the matrix M' has full column rank and all solutions of S are affinely independent.

∎

Lemma 2 *The non-negativity constraints $x_{uv} \geq 0$ define facets for all $uv \in E$. The one-node-aggregation constraints (2.8) define facets for P.*

Proof: A feasible constraint defines a facet, if it is active for $|V| - k + |E|$ affinely independent solutions.

For the first part, we have to show that $x_{uv} = 0$ for $|V| - k + |E|$ are affinely independent solutions. This is obvious, because all but one of the solutions in the proof of Lemma 1 satisfy this equality. The second part follows if we choose a basic solution b with $b_v = 0$. Then $\sum_{u \in V_i} x_{uv} = x_v$ for all but one selected solutions.

∎

Definition: Let h, i, j be pairwise distinct column indices, (S_1, S_2) be a partition of V_i, (T_1, T_2) be a partition of V_j and (R_1, R_2) be a partition of V_h. Furthermore, all sets are assumed to be non-empty. We define the edge-incompatibility constraint for $S_1, S_2, T_1, T_2, R_1, R_2$ as

$$\sum_{u \in S_2, v \in T_1} x_{uv} + \sum_{v \in T_2, w \in R_1} x_{vw} + \sum_{w \in R_2, u \in S_1} x_{uw} \leq 1 \tag{2.9}$$

Lemma 3 *The edge-incompatibility constraints are feasible for P and define facets.*

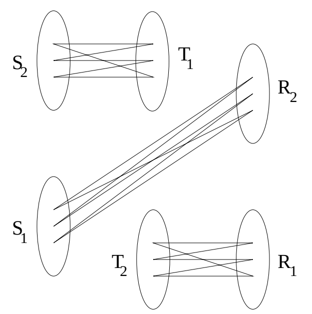

Figure 2.2: Only one of these edges can be selected.

Proof: We first show the feasibility of the constraint. If any edge of the constraint is selected, no other edge of the constraint can be selected, since at least one of its endpoints cannot be selected; see Figure 2.2.

To show that the constraints are facet-defining, we have to argue differently from above, where we could show directly that there are $|V| - k + |E|$ affinely independent solutions for which the equality of the constraint holds. Here, we use an indirect way to show that there are $|V|-k+|E|$ affinely independent solutions for which the equality of the constraint holds. Look at any constraint $\sum_{v\in V}\pi_v x_v + \sum_{uv\in E}\pi_{uv}x_{uv} \le \pi_0$. Let $F=\{p^1,\dots p^t\}$ be the set of feasible solutions for which the constraint is active. The constraint defines a facet if all solutions (μ,μ_0) of the set of equalities

$$\sum_{v\in V}\mu_v p_v^i + \sum_{uv\in E}\mu_{uv}p_{uv}^i = \mu_0 \text{ for all } p^i \in F$$

are linear combinations of the coefficients of the constraint $\sum_{v\in V}\pi_v x_v + \sum_{uv\in E}\pi_{uv}x_{uv} = \pi_0$ and the feasible equations $A^{=}x=b^{=}$ of the problem. So all solutions of the above systems must be of the form $\mu_v=\alpha\pi_v+\lambda_{c(v)}$, $\mu_{uv}=\alpha\pi_{uv}$ and $\mu_0=\alpha\pi_0+\sum_{i=1}^k\lambda_k$ for some $\alpha\in\mathbf{R}$, $\lambda\in\mathbf{R}^k$ (see Theorem 3.6. in chapter I of [31]).

To get rid of the linear combinations with the coefficients of the feasible equations of the problem (*e.g.*, to get rid of λ), we define a normal form of a constraint. Let b be a basic solution. We call a constraint in normal form with respect to b if $\pi_v=0$ for all v with $b_v=1$. Every constraint has (at least) one equivalent constraint in normal form with respect to b, since one can subtract the equations $\sum_{v\in V_i}x_v=1$ with suitable factors λ_i.

Thus a constraint in normal form with respect to b defines a facet of P if all solutions of the equation system

$$\begin{aligned}\sum_{v\in V}\mu_v p_v^i + \sum_{uv\in E}\mu_{uv}p_{uv}^i &= \mu_0 \text{ for all } p^i\in F\\ \mu_v &= 0 \text{ for all } v \text{ with } b_v=1\end{aligned} \tag{2.10}$$

are of the form $\mu_v=\alpha\pi_v$, $\mu_{uv}=\alpha\pi_{uv}$ and $\mu_0=\alpha\pi_0$ for some $\alpha\in\mathbf{R}$.

Choose any basic solution b, such that at least one edge xy of the constraint is selectable, *i.e.* $b_x=1$ and $b_y=1$. The constraint (2.9) is in normal form with respect to b. Let F be the set of all feasible solutions where exactly one edge of the constraint is chosen. We first argue that $\mu_v=0$ for all nodes $v\in V$. By definition of the normal form, this is true for all nodes of the basic solution. Let $v\notin V_h\cup V_i\cup V_j$. Look at the following two solutions: the nodes of the basic solution together with the edge xy and the solution where v and $b(c(v))$ are exchanged and the same edge is selected. More formally, we look at the solutions $b+e_{xy}$ and $b+e_{xy}+e_v-e_{b(c(v))}$. Since $\mu_{b(c(v))}=0$, the difference of the two resulting constraints is $\mu_v=0$. Now let $v\in V_h\cup V_i\cup V_j$. Let wr be an edge of the constraint that has neither endpoint in the same row as v. Take any solution where this edge and $b(c(v))$ are selected and the solution where $b(c(v))$ is replaced by v. Again the difference between the two constraints is $\mu_v=0$.

Now we conclude that $\mu_{wr}=\mu_0$ for all edges wr in the constraint (Look at any solution where only this edge is selected.), and that $\mu_{wr}=0$ for all edges not in the constraint (Look at any solution where this edge and one edge of the constraint is selected and the solution where only the edge of the constraint is selected. It it possible to select an edge of the constraint, because every node in $S_1\cup S_2\cup T_1\cup T_2\cup R_1\cup R_2$ has an adjacent edge in the constraint). This proves that all solutions of the equation system 2.10 have the required form and we are done.

∎

Definition: Let h, i, j be pairwise distinct column indices, $S \subset V_i$, $T \subset V_j$ with $S, T \neq \emptyset$, and let R_1, R_2 be a partition of V_h with $R_i \neq \emptyset$ for $i = 1, 2$. We define the node-aggregation constraints for S, T, R_1, R_2 as

$$\sum_{u \in S, v \in T} x_{uv} + \sum_{u \in S, w \in R_1} x_{uw} + \sum_{v \in T, w \in R_2} x_{vw} \leq \sum_{u \in S} x_u + \sum_{v \in T} x_v \tag{2.11}$$

Lemma 4 *The node-aggregation constraints (2.11) are feasible for P and define facets.*

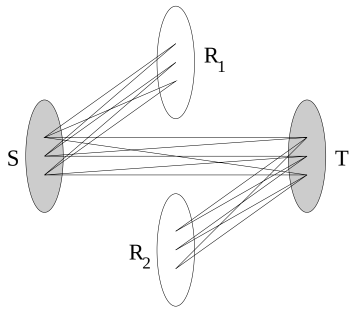

Figure 2.3: One can select only as many edges as nodes in the shaded sets.

Proof: Again we start with the feasibility proof. If a node in S, a node in T, and a node in R_1 are selected we can select exactly one edge going from S to T and one edge going from S to R_1 but no edges from T to R_2. If a node in S is selected but no node in T we can not select an edge from S to T or from T to R_2. Furthermore, we can select at most one edge from S to R_1. If neither a node from S nor a node from T is selected we cannot select any of the edges of the constraint. The other cases are analogous (compare Figure 2.3).

We use the same indirect way to show that the constraints are facet-defining as in the proof of Lemma 3. Choose any basic solution b with $b_u = 0$ for all $u \in S \cup T$. The constraint (2.11) is in normal form with respect to b. Let F be the set of feasible solutions that satisfy the constraint with equality. Furthermore let (μ, μ_0) be the solutions of the equation system 2.10 for F. Let w be the node in V_h with $b_w = 1$ and w.l.o.g. assume $w \in R_1$ and r is any node of R_2. Since the constraint is active for b we conclude that $\mu_0 = 0$. Furthermore, we obtain:

1. $\mu_v = 0$ for all $v \in V$ that are not in the constraint (Look at the difference between the constraints for the solutions b and $b + e_v - e_{b(c(v))}$).

2. $\mu_{uv} = 0$ for all $uv \in E$ that are not in the constraint (Look at the difference between the constraint of any active solution p with $p_{uv} = 1$ and $p - e_{uv}$).

3. $\mu_u = -\mu_{uv}$ for all $u \in S$, $v \in R_1$ with $uv \in E$ (Look at the difference between the constraints of $b + e_v - e_{b(c(v))}$ and $b + e_v - e_{b(c(v))} + e_u - e_{b(c(u))} + e_{uv}$).

4. $\mu_u = -\mu_{uv}$ for all $u \in T$, $v \in R_2$ with $uv \in E$ (The same argument).

5. $\mu_v = -\mu_{uv}$ for all $u \in S$, $v \in T$ with $uv \in E$ (Look at the difference between the constraints for the solutions $b + e_u - e_{b(c(u))} + e_{uw}$ and $b + e_u - e_{b(c(v))} + e_{uw} + e_v - e_{b(c(v))} + e_{uv}$).

6. $\mu_u = -\mu_{uv}$ for all $u \in S$, $v \in T$ with $uv \in E$ (Look at the difference between the constraints for the solutions $b + e_v - e_{b(c(v))} + e_r - e_w + e_{rv}$ and $b + e_v - e_{b(c(v))} + e_r - e_w + e_{rv} + e_u - e_{b(c(u))} + e_{uv}$).

Thus $\mu_u = -\mu_{uv}$ for all $u \in V$ and $uv \in E$ of the constraint. Since the subgraph induced by the constraint is connected, all μ_v have the same value. Thus all solutions of the equation system 2.10 have the required form and we are done.

∎

We have implemented a branch-&-cut algorithm for the GMEC problem using the C++ class library LEDA [28], ABACUS [19] for the administration of the branch-and-cut tree, and CPLEX [15] to solve the linear programs.

We use the node-summation equations (2.3) and the one-node-aggregation constraints (2.8) as the initial constraint system. The only constraints that are separated are the node-aggregation constraints (2.11) with $|S| = |T| = 1$.

We use a simple, brute force algorithm. We iterate over all triples (u, v, h), where $u, v \in V$, h is a column of the graph G and $c(u), c(v), h$ are pairwise distinct. The optimal partition is obtained by iterating over the nodes of V_h and putting a node w into R_1 if $x_{uw} > x_{vw}$ and into R_2 otherwise.

The running time is $O(\sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{h=1}^{k} (|V_i||V_j||V_h|)) = O(|V|^3)$.

2.3 Energetic evaluation

After determining an optimal rotamer set for each candidate, the free energy of association is determined using the method proposed by Jackson and Sternberg [17]. We assign the rotamers of the minimum energy conformation to the respective side chains and then optimize the side chains of the binding–site using the AMBER force field. We decompose the binding free energy ΔG_{ass} as follows:

$$\Delta G_{ass} = \Delta G^{ES} + \Delta G^{cav} + \Delta G^{conf} + \Delta G^{vdW} \tag{2.12}$$

ΔG^{ES} represents the electrostatic contribution, ΔG^{cav} the cavitation free energy in water, ΔG^{conf} the change in conformational entropy of the proteins, and ΔG^{vdW} the change in the van der Waals free energy.

The electrostatic contribution ΔG^{ES} to the binding free energy is composed as follows:

$$\Delta G^{ES} = \Delta\Delta G^{A}_{solv} + \Delta\Delta G^{B}_{solv} + \Delta G^{AB}_{int} \tag{2.13}$$

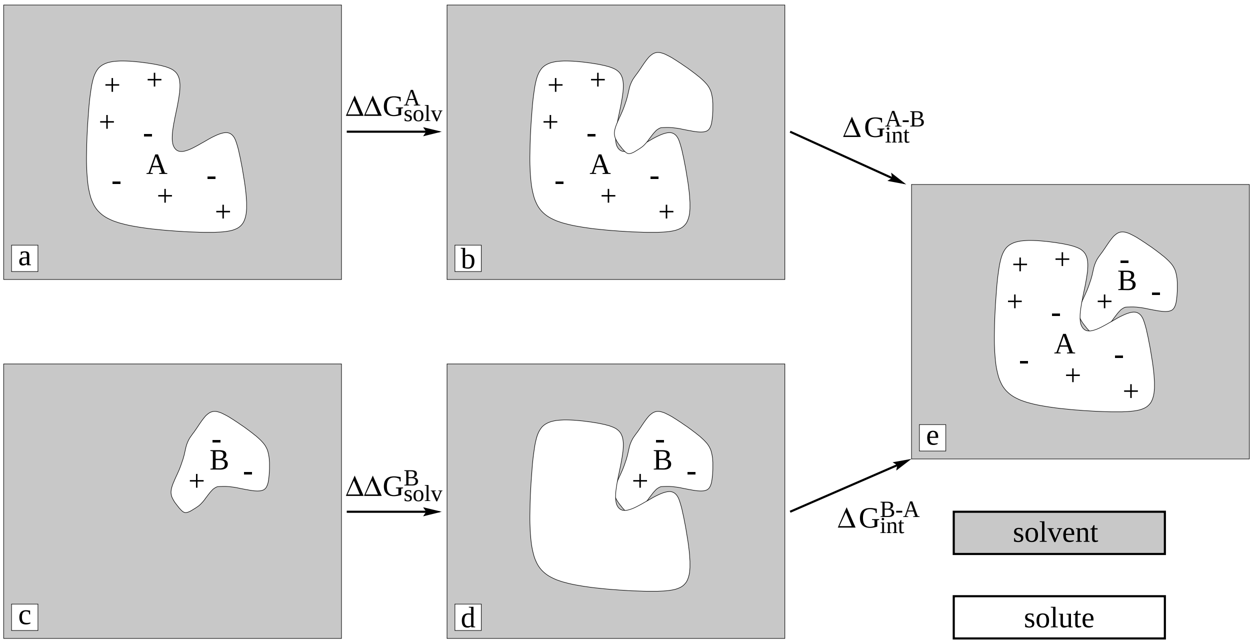

Figure 2.4: Calculation of the electrostatic contributions to the binding free energy.

To evaluate the electrostatic contribution, we use the scheme in Figure 2.4. First, we calculate the total electrostatic energy of A in water (dielectric constant of water $\varepsilon_w = 80$, dielectric constant of the protein $\varepsilon_p = 2$). Then, we create a complex-shaped low dielectric constant cavity and charge only A. B remains uncharged. The change in solvation free energy on binding $\Delta\Delta G^{A}_{solv}$ for A is given by the difference between these two energies. The interaction energy ΔG^{AB}_{int} is then determined as the energy of the point charges of B in the field caused by A. The same process is finally repeated for protein B in the same manner. In our experiments, we used cubic grids with a spacing of 0.5 Å.

Cavitation free energy The change in cavitation free energy is calculated as a linear function in the change of the molecular surface area

$$\Delta G_{cav} = \gamma\ (A_{AB} - A_A - A_B), \tag{2.14}$$

where A_{AB}, A_A, and A_B are the molecular surface areas of the complex, protein A, and protein B, respectively, and γ is a constant ($0.289\ \mathrm{kJmol^{-1}\AA^{-2}}$). We use the algorithm by Connolly [8] to calculate the molecular surface areas with a probe radius of 1.4 Å. All calculations are performed using the FDPB code of our C++ application framework BALL [5, 22]. We use the PARSE set of atom radii and charges by Sitkoff et al. [36].

3 Experimental results

We applied our techniques to three protease-inhibitor complexes. The structures of three proteases A and the corresponding protein inhibitors B were taken from the PDB: subtilisin carlsberg (1SBC) with chymotrypsin inhibitor 2 (2CI2), β–trypsin (1TPO) with trypsin inhibitor (4PTI), and α–chymotrypsin A (5CHA, chain A) with ovomucoid (2OVO). Water was removed from each of these complexes, hydrogens were added and their positions were optimized using the AMBER [9] force field.

Combinatorial complexity The iterative application of the DEE theorem reduced the combinatorial complexity usually by about ten orders of magnitude, thus greatly simplifying the solution of the GMEC problem. Numbers for the three examples are given in the following table:

complex	no. of side chains	# rotamer combinations before DEE	after DEE
1TPO/4PTI	51	$6 \cdot 10^{60}$	$8 \cdot 10^{45}$
1SBC/2CI2	50	$1 \cdot 10^{55}$	$2 \cdot 10^{46}$
5CHA/2OVO	54	$1 \cdot 10^{56}$	$2 \cdot 10^{48}$

Ranking of the structures In the case of 1TPO/4PTI, the AMBER energy already gave a clear signal; the best four candidates were approximations of the complex structure (see Fig. 3.1). The estimated binding free energy (as described in Section 2) for this example also ranked an approximation of the true complex structure as number one (see Fig. 3.2).

For the example 1SBC/2CI2, the AMBER energy was not sufficient to correctly predict the true complex structure. However, when ranked according to the binding free energy, the best candidate was a good approximation of the true complex structure for all examples (see Fig. 3.3, 3.2, and 3.4).

We also tried to predict the structure of these examples using the binding free energy alone (*i.e.*, without a prior side chain demangling). None of the complexes was correctly predicted. Hence, side chain demangling as well as an advanced scoring method are both needed to predict the correct complex structure in these cases.

Quality of the heuristic For the case of 1TPO/4PTI, we checked the quality of the results obtained from the MG method by calculating the optimal solution using the branch–

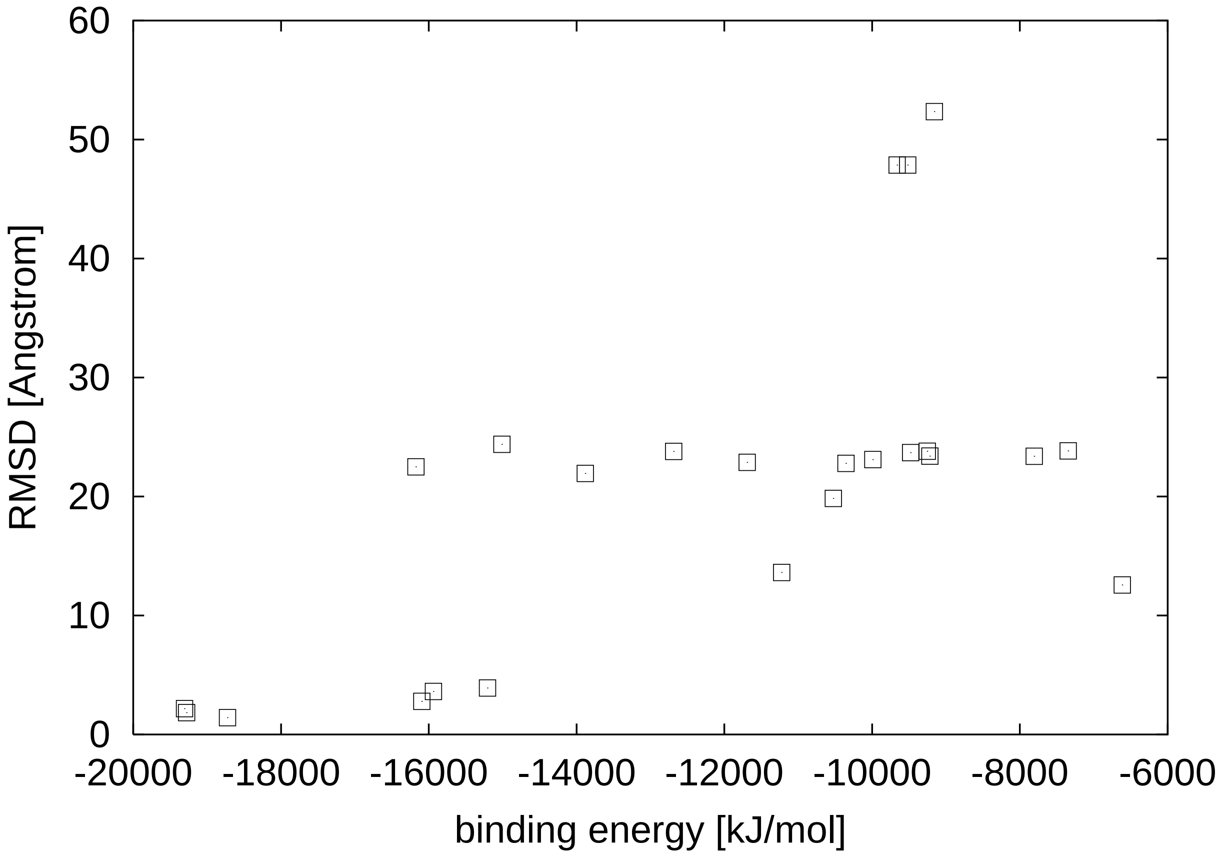

Figure 3.1: 1TPO/4PTI - AMBER energy of the candidates (MG).

&-cut algorithm. We performed this calculation for 34 out of the 60 candidates. In most cases, the solution found by the heuristic was close to the optimal solution (see Fig. 3.5). The average error in the AMBER energies was about 8000 kJ/mol. However, the quality of the solutions was better for good approximations of the true complex structure. Among the 34 candidates considered, 7 were good approximations (RMSD[1] ≤ 4 Å). Only one of these seven candidates was further than 1000 kJ/mol away from the optimum.

Side chain placement For the protein docking problem not all side chains are of equal importance. However, the placement of a single side chain may determine the success of the docking. A prominent example is LYS:15 of 4PTI. When compared to the bound structure, this side chain has to turn down to the core of 1TPO to fit properly. Fig. 3.6 shows the rotamers of this side chain. However, it is not neccesary for LYS:15 to assume the fully extended position as in the complex structure, it just has to turn down a bit further than it does in the unbound structure to avoid clashes with the backbone of 1TPO. Although the correct placement of the side chain is not achieved, the selected rotamer is sufficient to allow for a reasonable binding mode.

[1]root mean square deviation of all atoms

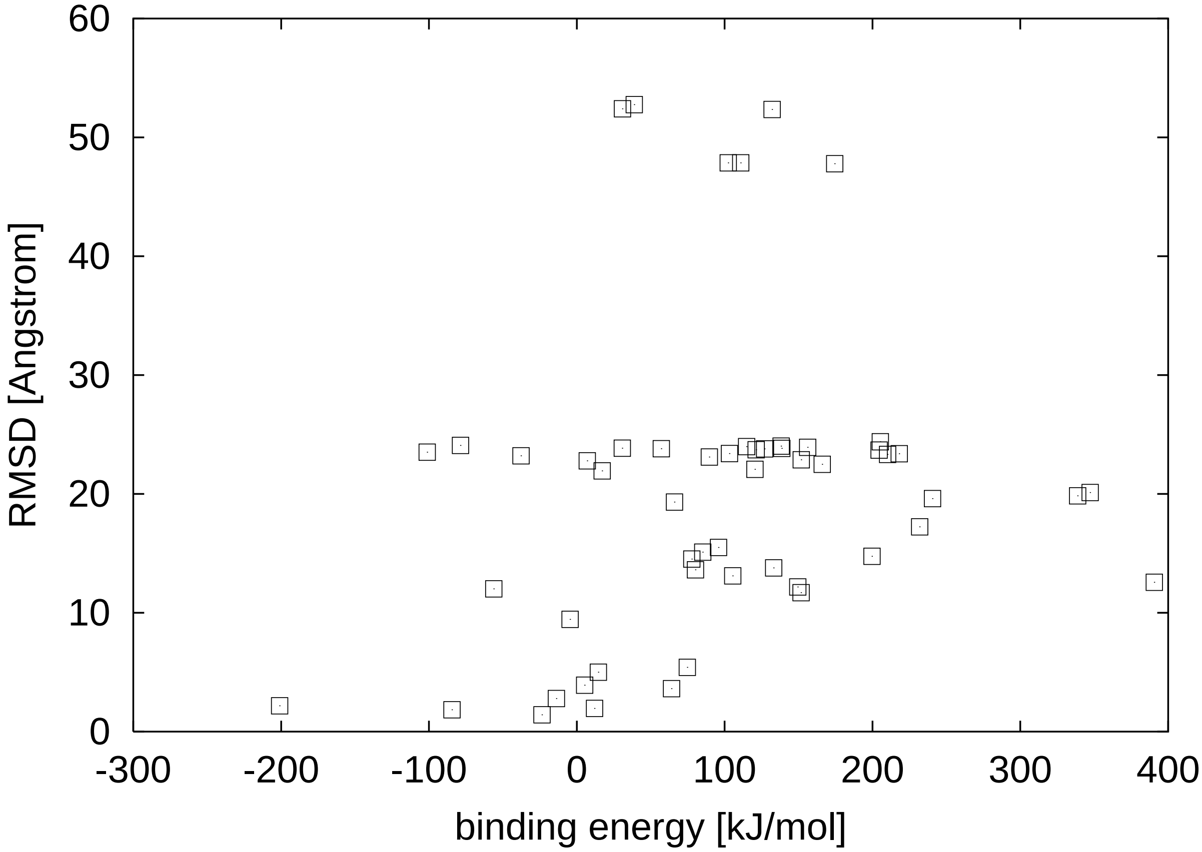

Figure 3.2: 1TPO/4PTI - ranking of the candidates according to their binding free energy (MG).

Running times We compared the running times of the different parts of our docking algorithm on a SUN Enterprise 10000 (333 MHz UltraSparc II processors, 12 GB RAM, Solaris 7, g++-2.95.2 with -O2). We averaged the running times for each stage of the algorithm over all 60 candidates considered.

stage	avg. time [min]
energies and DEE	30
ILP	14
multi–greedy	3
side chain optimization	5
energetic evaluation	70

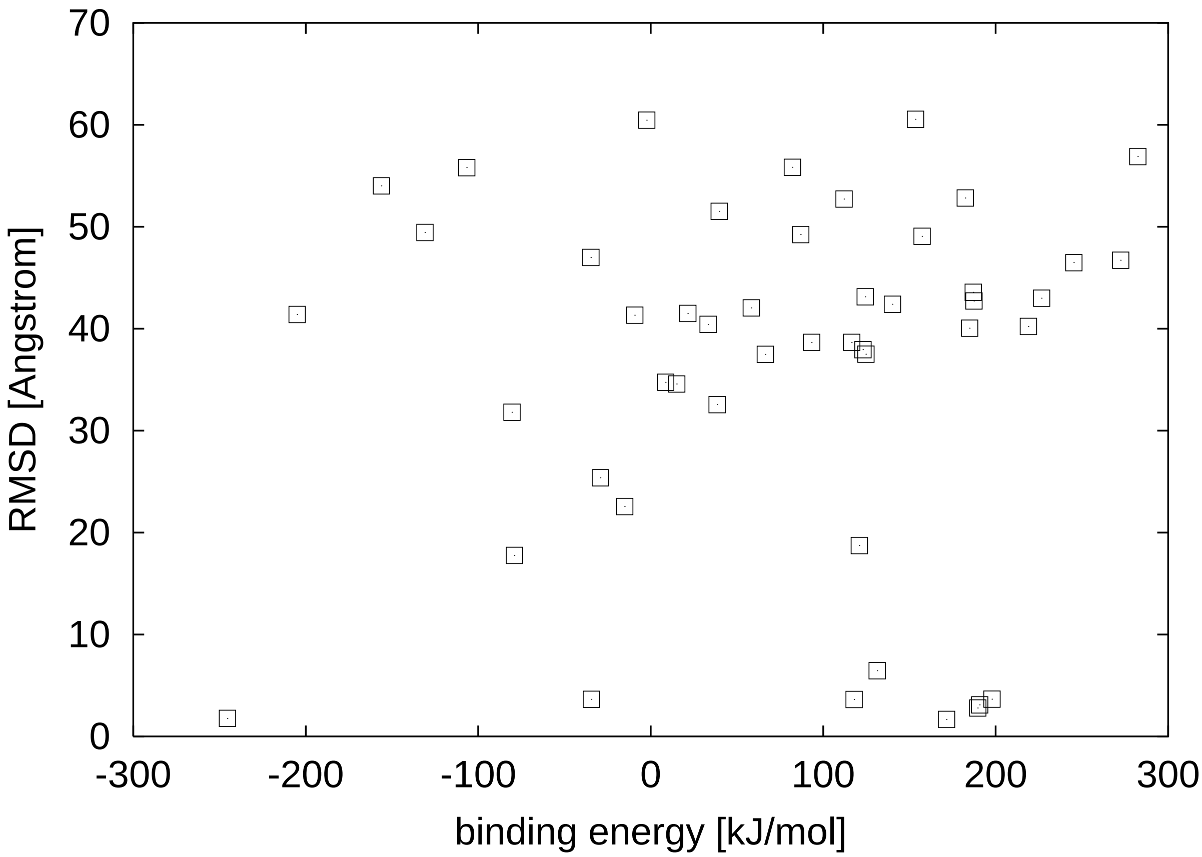

Figure 3.3: 1SBC/4CI2 - ranking of the candidates according to their binding free energy (MG).

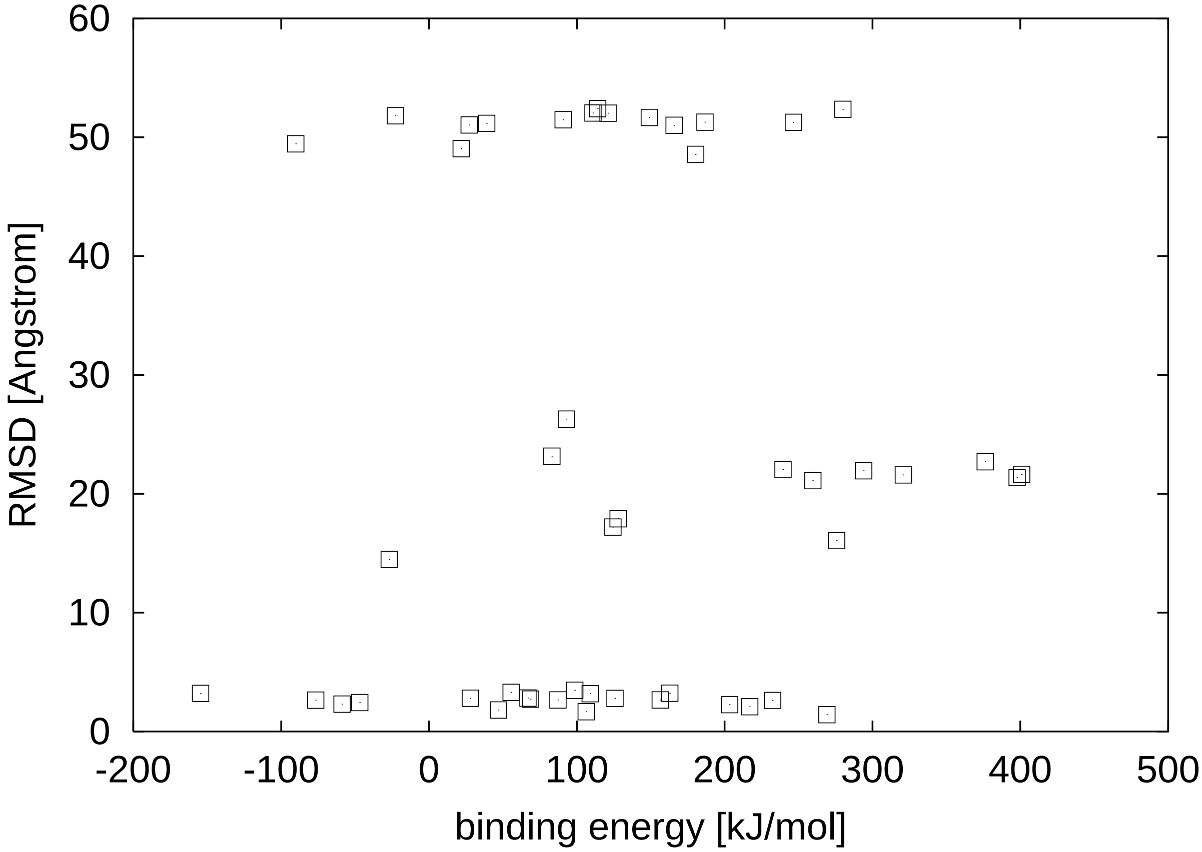

Figure 3.4: 5CHA/2OVO - ranking of the candidates according to their binding free energy (MG).

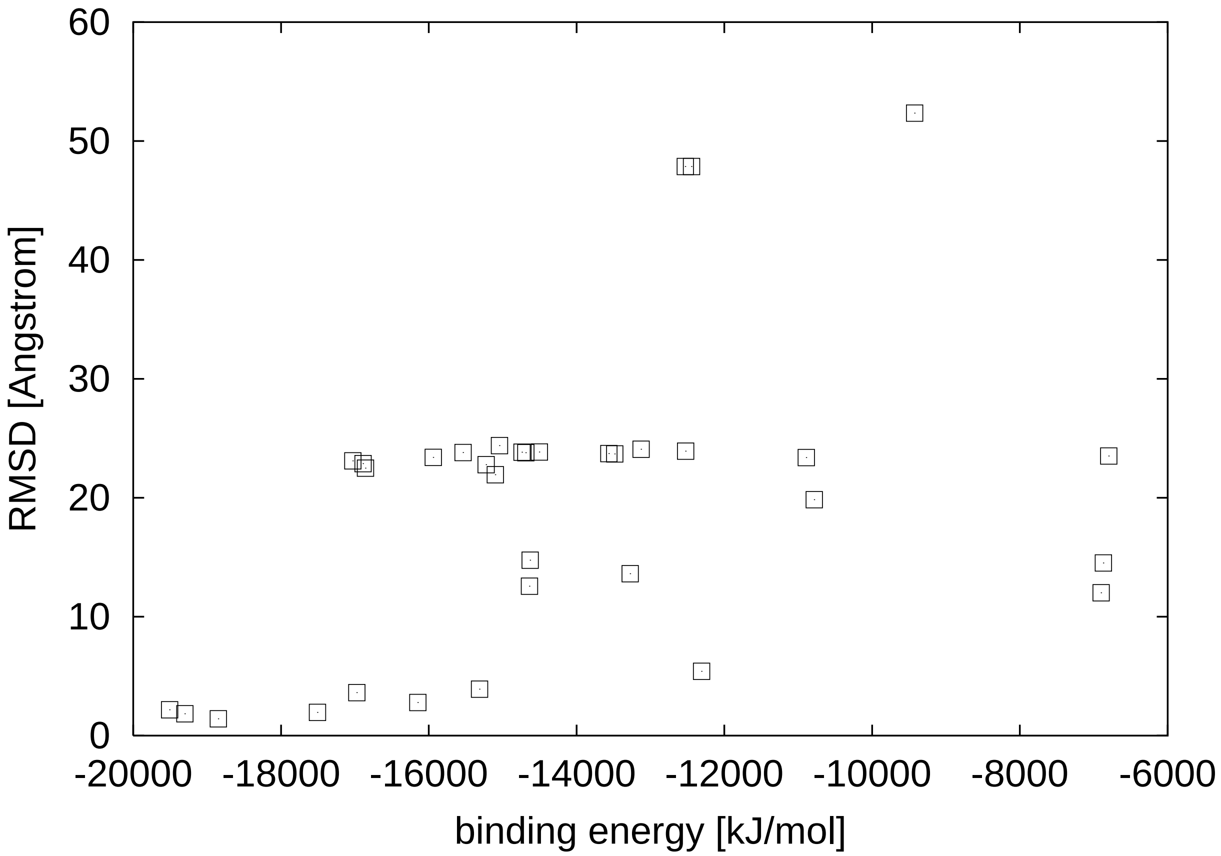

Figure 3.5: 1TPO/4PTI - AMBER energy of the candidates (ILP).

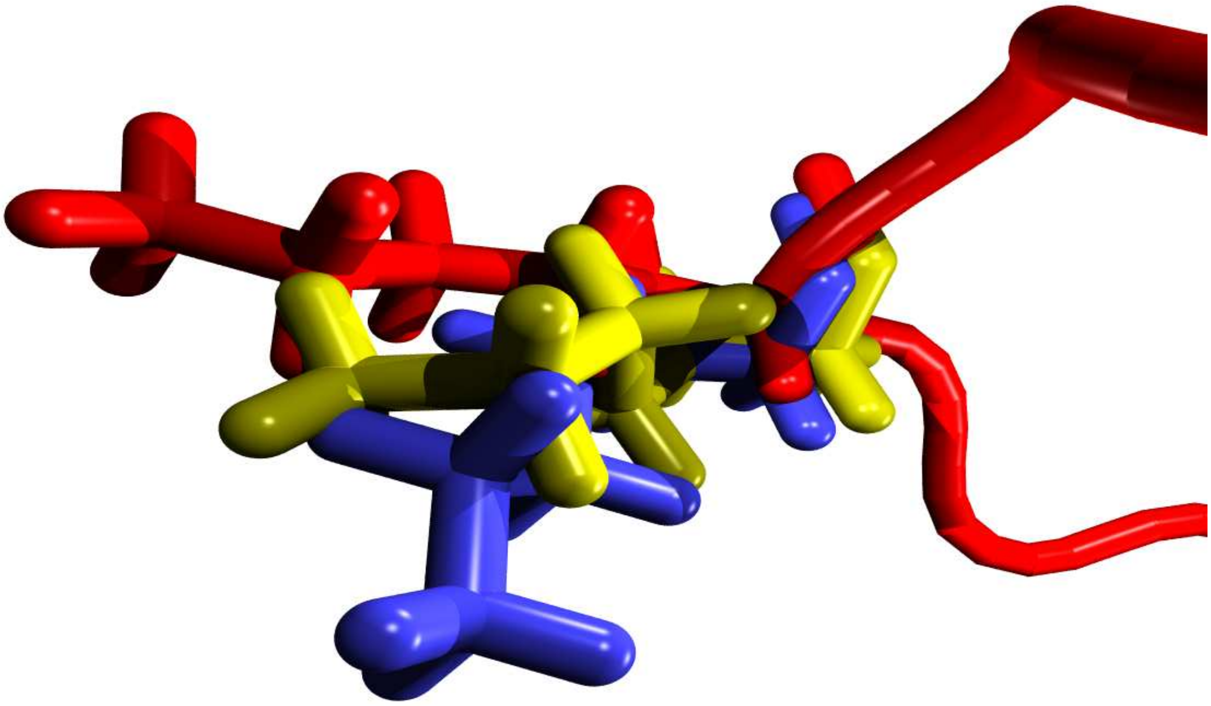

Figure 3.6: Placement of LYS15 of 4PTI (red: complex conformation from 2PTC, blue: conformation from 4PTI, yellow: optimal rotamer).

4 Discussion

Our docking approach determines an optimal placement for the side chains in the protein interface or a good approximation of this conformation, while keeping the protein backbone rigid. By reducing the conformational space of the side chains to discrete rotamers, we are able to formulate the problem as a combinatorial optimization problem. Two techniques are presented to solve this problem. The first technique is a heuristic multi–greedy approach, which is fast but does not necessarily produce an optimal solution. We verified the quality of the heuristic by comparing it to an optimal solution calculated by a branch–&–cut algorithm. When ranking the solutions with respect to their binding free energy, both methods were able to predict the correct complex structure in the examples we tested. We found that the heuristic method produces solutions that are very close to the optimal solution, although the side chains are not optimally placed.

The correct placement of the side chains is to a large extent dependent on the force field used to calculate the interaction energies. The AMBER force field does not include solvation contributions, so we hope to improve our results by integrating appropriate methods to descriptions of solvent effects.

In our experience, the failure to distinguish between true and false positives is largely caused by fluctuations in the solvation free energy. We plan to improve our results by using smaller grid spacings in the FDPB calculations and by using the difference in reaction field energy instead of the difference in total electrostatic energy. We also intend to improve the branch-&-cut approach by developing new separation routines and by identifying new classes of facet–defining inequalities.

Bibliography

[1] R. Abagyan and M. Totrov. Biased probability Monte Carlo conformational searches and electrostatic calculations for peptides and proteins. *J. Mol. Biol.*, 235:983–1002, 1994.

[2] F. Ackermann, G. Herrmann, F. Kummert, S. Posch, G. Sagerer, and D. Schomburg. Protein docking combining symbolic descriptions of molecular surfaces and grid-based scoring functions. In *Proceedings of the Third International Conference on Intelligent Systems in Molecular Biology*, pages 3–11, Menlo Park, California, 1995. AAAI Press.

[3] D. J. Bacon and J. Moult. Docking by least-squares fitting of molecular surface patterns. *J. Mol. Biol.*, 225:849–858, 1992.

[4] F. Bernstein, T. Koetzle, G. Williams, E. Meyer Jr, M. Brice, J. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112:535, 1977.

[5] N. Boghossian, O. Kohlbacher, and H.-P. Lenhof. BALL: Biochemical Algorithms Library. In J. Vitter and C. Zaroliagis, editors, *Algorithm Engineering, 3rd International Workshop, WAE'99, Proceedings*, volume 1668 of *Lecture Notes in Computer Science (LNCS)*, pages 330–344. Springer, 1999.

[6] R. E. Bruccoleri and J. Novotny. Antibody modeling using the conformational search program CONGEN. *Immunomethods*, 1:96–106, 1992.

[7] J. Cherfils, S. Duquerroy, and J. Janin. Protein-protein recognition analysis by docking simulation. *Proteins*, 11:271–280, 1991.

[8] M. L. Connolly. Shape complementarity at the hemoglobin $\alpha_1\beta_1$ subunit interface. *Biopolymers*, 25:1229–1247, 1986.

[9] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. *J. Am. Chem. Soc.*, 117:5179–5197, 1995.

[10] J. Desmet, M. D. Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in the protein side-chain positioning. *Nature*, 356:539–542, April 1992.

[11] R. L. Dunbrack and F. E. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science*, 6:1661–1681, 1997.

[12] D. Fischer, S. L. Lin, H. J. Wolfson, and R. Nussinov. A geometry-based suite of molecular docking processes. *J. Mol. Biol.*, 248:459–477, 1995.

[13] M. Helmer-Citterich and A. Tramontano. PUZZLE: a new method for automated protein docking based on surface shape complementarity. *J. Mol. Biol.*, 235:1021–1031, 1994.

[14] L. Holm and C. Sander. Fast and simple monte carlo algorithm for side-chain optimization in proteins: application to model building by homology. *Proteins*, 14:213–223, 1992.

[15] ILOG. *ILOG CPLEX 6.5 : user's manual.* ILOG, Bad Homburg, march 1999 edition, 1999.

[16] R. M. Jackson, H. A. Gabb, and M. J. E. Sternberg. Rapid refinement of protein interfaces incorporating solvation: Application to the protein docking problem. *J. Mol. Biol.*, 276:265–285, 1998.

[17] R. M. Jackson and M. J. E. Sternberg. A continuum model for protein-protein interactions: Application to the protein docking problem. *J. Mol. Biol.*, 250:258–275, 1995.

[18] F. Jiang and S. H. Kim. Soft docking: matching of molecular surface cubes. *J. Mol. Biol.*, 219:79–102, 1991.

[19] M. Jünger and S. Thienel. Introduction to ABACUS - A branch-and-CUt system. Technical report, Informatik, Universität zu Köln, 1997.

[20] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A. Friesem, C. Afalo, and I. A. Vakser. Molecular surface recognition: Determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Natl. Acad. Sci. USA*, 89:2195–2199, 1992.

[21] P. Koehl and M. Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J. Mol. Biol*, 239:249–275, 1994.

[22] O. Kohlbacher and H.-P. Lenhof. Rapid software prototyping in computational molecular biology. In *Proceedings of the German Conference on Bioinformatics (GCB'99)*, 1999.

[23] C. A. Laughton. Prediction of protein side-chain conformations from local three-dimensional homology relationships. *J. Mol. Biol.*, 235:1088–1097, 1994.

[24] A. R. Leach. Ligand docking to proteins with discrete side-chain flexibility. *J. Mol. Biol.*, 235:345–356, 1994.

[25] A. R. Leach and A. P. Lemon. Exploring the conformational space of protein side chains using dead-end elimination and the a* algorithm. *Proteins: Struct., Function, and Genet.*, 33:227–239, 1998.

[26] H.-P. Lenhof. An algorithm for the protein docking problem. In D. Schomburg and U. Lessel, editors, *Bioinformatics: From nucleic acids and proteins to cell metabolism. GBF Monographs Volume 18*, pages 125–139, 1995.

[27] H.-P. Lenhof. New contact measures for the protein docking problem. In *Proc. of the First Annual International Conference on Computational Molecular Biology RECOMB 97*, pages 182–191, 1997.

[28] K. Mehlhorn, S. Näher, M. Seel, and C. Uhrig. *The LEDA user manual: version 3.8.* Max-Planck-Institut für Informatik, Saarbrücken, 1999.

[29] M. Meyer, P. Wilson, and D. Schomburg. Hydrogen bonding and molecular surface shape complementarity as a basis for protein docking. *J. Mol. Biol.*, 264(1):199–210, 1996.

[30] J. Moon and W. Howe. Computer design of bioactive molecules: A method for receptor–based de novo ligand design. *Proteins: Struct. Funct. Genetics*, 11:314–328, 1991.

[31] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, New York;Chichester;Brisbane, 1988.

[32] R. Norel, S. L. Lin, D. Xu, H. J. Wolfson, and R. Nussinov. Molecular surface variability and induced conformational changes upon protein-protein association. In R. H. Sarma and M. H. Sarma, editors, *Structure, Motion, Interaction and Expression of Biological Macromolecules. Proceedings of the Tenth Conversation. State University of New York*, pages 33–51. Adenine Press, 1998.

[33] J. Ponder and F. Richards. Tertiary templates for proteins – use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol. Biol.*, 193:775–791, 1987.

[34] M. Rarey, B. Kramer, T. Lengauer, and G. Klebe. A fast flexible docking method using an incremental construction algorithm. *J. Mol. Biol.*, 261:470–489, 1997.

[35] B. K. Shoichet and I. D. Kuntz. Protein docking and complementarity. *J. Mol. Biol.*, 221:79–102, 1991.

[36] D. Sitkoff, K. A. Sharp, and B. Honig. Accurate calculation of hydration free energies using macroscopic solvent models. *J. Phys. Chem.*, 98(7):1978–1988, 1994.

[37] M. Totrov and R. Abagyan. Detailed ab initio prediction of lysozyme-antibody complex with 1.6 Å accuracy. *Nat. Struct. Biol.*, 1:259–263, 1994.

[38] P. H. I. Walls and M. J. E. Sternberg. New algorithm to model protein-protein recognition based on surface complementarity. *J. Mol. Biol.*, 228:277–297, 1992.

[39] Z. Weng, S. Vajda, and C. Delisi. Prediction of protein complexes using empirical free energy functions. *Protein Science*, 5:614–626, 1996.

[40] L. A. Wolsey. *Integer programming.* Wiley-interscience series in discrete mathematics and optimization. Wiley & Sons, New York, 1998.

I N F O R M A T I K

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Im Stadtwald
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2000-4-001	J. Kautz, W. Heidrich, K. Daubert	Bump Map Shadows for OpenGL Rendering
MPI-I-2000-1-001	E. Althaus, O. Kohlbacher, H. Lenhof, P. Müller	A branch and cut algorithm for the optimal solution of the side-chain placement problem
MPI-I-1999-4-001	J. Haber, H. Seidel	A Framework for Evaluating the Quality of Lossy Image Compression
MPI-I-1999-3-005	T.A. Henzinger, J. Raskin, P. Schobbens	Axioms for Real-Time Logics
MPI-I-1999-3-004	J. Raskin, P. Schobbens	Proving a conjecture of Andreka on temporal logic
MPI-I-1999-3-003	T.A. Henzinger, J. Raskin, P. Schobbens	Fully Decidable Logics, Automata and Classical Theories for Defining Regular Real-Time Languages
MPI-I-1999-3-002	J. Raskin, P. Schobbens	The Logic of Event Clocks
MPI-I-1999-3-001	S. Vorobyov	New Lower Bounds for the Expressiveness and the Higher-Order Matching Problem in the Simply Typed Lambda Calculus
MPI-I-1999-2-008	A. Bockmayr, F. Eisenbrand	Cutting Planes and the Elementary Closure in Fixed Dimension
MPI-I-1999-2-007	G. Delzanno, J. Raskin	Symbolic Representation of Upward-closed Sets
MPI-I-1999-2-006	A. Nonnengart	A Deductive Model Checking Approach for Hybrid Systems
MPI-I-1999-2-005	J. Wu	Symmetries in Logic Programs
MPI-I-1999-2-004	V. Cortier, H. Ganzinger, F. Jacquemard, M. Veanes	Decidable fragments of simultaneous rigid reachability
MPI-I-1999-2-003	U. Waldmann	Cancellative Superposition Decides the Theory of Divisible Torsion-Free Abelian Groups
MPI-I-1999-2-001	W. Charatonik	Automata on DAG Representations of Finite Trees
MPI-I-1999-1-007	C. Burnikel, K. Mehlhorn, M. Seel	A simple way to recognize a correct Voronoi diagram of line segments
MPI-I-1999-1-006	M. Nissen	Integration of Graph Iterators into LEDA
MPI-I-1999-1-005	J.F. Sibeyn	Ultimate Parallel List Ranking ?
MPI-I-1999-1-004	M. Nissen, K. Weihe	How generic language extensions enable “open-world” desing in Java
MPI-I-1999-1-003	P. Sanders, S. Egner, J. Korst	Fast Concurrent Access to Parallel Disks
MPI-I-1999-1-002	N.P. Boghossian, O. Kohlbacher, H.-. Lenhof	BALL: Biochemical Algorithms Library
MPI-I-1999-1-001	A. Crauser, P. Ferragina	A Theoretical and Experimental Study on the Construction of Suffix Arrays in External Memory

MPI-I-98-2-018	F. Eisenbrand	A Note on the Membership Problem for the First Elementary Closure of a Polyhedron
MPI-I-98-2-017	M. Tzakova, P. Blackburn	Hybridizing Concept Languages
MPI-I-98-2-014	Y. Gurevich, M. Veanes	Partisan Corroboration, and Shifted Pairing
MPI-I-98-2-013	H. Ganzinger, F. Jacquemard, M. Veanes	Rigid Reachability
MPI-I-98-2-012	G. Delzanno, A. Podelski	Model Checking Infinite-state Systems in CLP
MPI-I-98-2-011	A. Degtyarev, A. Voronkov	Equality Reasoning in Sequent-Based Calculi
MPI-I-98-2-010	S. Ramangalahy	Strategies for Conformance Testing
MPI-I-98-2-009	S. Vorobyov	The Undecidability of the First-Order Theories of One Step Rewriting in Linear Canonical Systems
MPI-I-98-2-008	S. Vorobyov	AE-Equational theory of context unification is Co-RE-Hard
MPI-I-98-2-007	S. Vorobyov	The Most Nonelementary Theory (A Direct Lower Bound Proof)
MPI-I-98-2-006	P. Blackburn, M. Tzakova	Hybrid Languages and Temporal Logic
MPI-I-98-2-005	M. Veanes	The Relation Between Second-Order Unification and Simultaneous Rigid E-Unification
MPI-I-98-2-004	S. Vorobyov	Satisfiability of Functional+Record Subtype Constraints is NP-Hard
MPI-I-98-2-003	R.A. Schmidt	E-Unification for Subsystems of S4
MPI-I-98-2-002	F. Jacquemard, C. Meyer, C. Weidenbach	Unification in Extensions of Shallow Equational Theories
MPI-I-98-1-031	G.W. Klau, P. Mutzel	Optimal Compaction of Orthogonal Grid Drawings
MPI-I-98-1-030	H. Brönniman, L. Kettner, S. Schirra, R. Veltkamp	Applications of the Generic Programming Paradigm in the Design of CGAL
MPI-I-98-1-029	P. Mutzel, R. Weiskircher	Optimizing Over All Combinatorial Embeddings of a Planar Graph
MPI-I-98-1-028	A. Crauser, K. Mehlhorn, E. Althaus, K. Brengel, T. Buchheit, J. Keller, H. Krone, O. Lambert, R. Schulte, S. Thiel, M. Westphal, R. Wirth	On the performance of LEDA-SM
MPI-I-98-1-027	C. Burnikel	Delaunay Graphs by Divide and Conquer
MPI-I-98-1-026	K. Jansen, L. Porkolab	Improved Approximation Schemes for Scheduling Unrelated Parallel Machines
MPI-I-98-1-025	K. Jansen, L. Porkolab	Linear-time Approximation Schemes for Scheduling Malleable Parallel Tasks
MPI-I-98-1-024	S. Burkhardt, A. Crauser, P. Ferragina, H. Lenhof, E. Rivals, M. Vingron	q-gram Based Database Searching Using a Suffix Array (QUASAR)
MPI-I-98-1-023	C. Burnikel	Rational Points on Circles
MPI-I-98-1-022	C. Burnikel, J. Ziegler	Fast Recursive Division
MPI-I-98-1-021	S. Albers, G. Schmidt	Scheduling with Unexpected Machine Breakdowns
MPI-I-98-1-020	C. Rüb	On Wallace's Method for the Generation of Normal Variates
MPI-I-98-1-019		2nd Workshop on Algorithm Engineering WAE '98 - Proceedings
MPI-I-98-1-018	D. Dubhashi, D. Ranjan	On Positive Influence and Negative Dependence
MPI-I-98-1-017	A. Crauser, P. Ferragina, K. Mehlhorn, U. Meyer, E. Ramos	Randomized External-Memory Algorithms for Some Geometric Problems
MPI-I-98-1-016	P. Krysta, K. Loryś	New Approximation Algorithms for the Achromatic Number
MPI-I-98-1-015	M.R. Henzinger, S. Leonardi	Scheduling Multicasts on Unit-Capacity Trees and Meshes
MPI-I-98-1-014	U. Meyer, J.F. Sibeyn	Time-Independent Gossiping on Full-Port Tori