CS 466
Introduction to Bioinformatics
Lecture 8

Mohammed El-Kebir

September 27, 2018

Course Announcements

Instructor:

- Mohammed El-Kebir (melkebir)
- Office hours: Mondays, 3:15-4:15pm

TA:

- Anusri Pampari (pampari2)
- Office hours: Thursdays, 11:00-11:59am in SC 4105

Homework 2 will be released Sept. 28 and will be due Oct. 5

Midterm on Oct. 10, 7-9pm, 1310 DCL

Outline

- Protein Contact Map Overlap

Reading:

- Lecture notes
- Caprara, A., Carr, R., Istrail, S., Lancia, G., & Walenz, B. (2004). 1001 Optimal PDB Structure Alignments: Integer Programming Methods for Finding the Maximum Contact Map Overlap. *Journal of Computational Biology*, *11*(1), 27–52. http://doi.org/10.1089/106652704773416876

How to Compare Two Protein Structures?

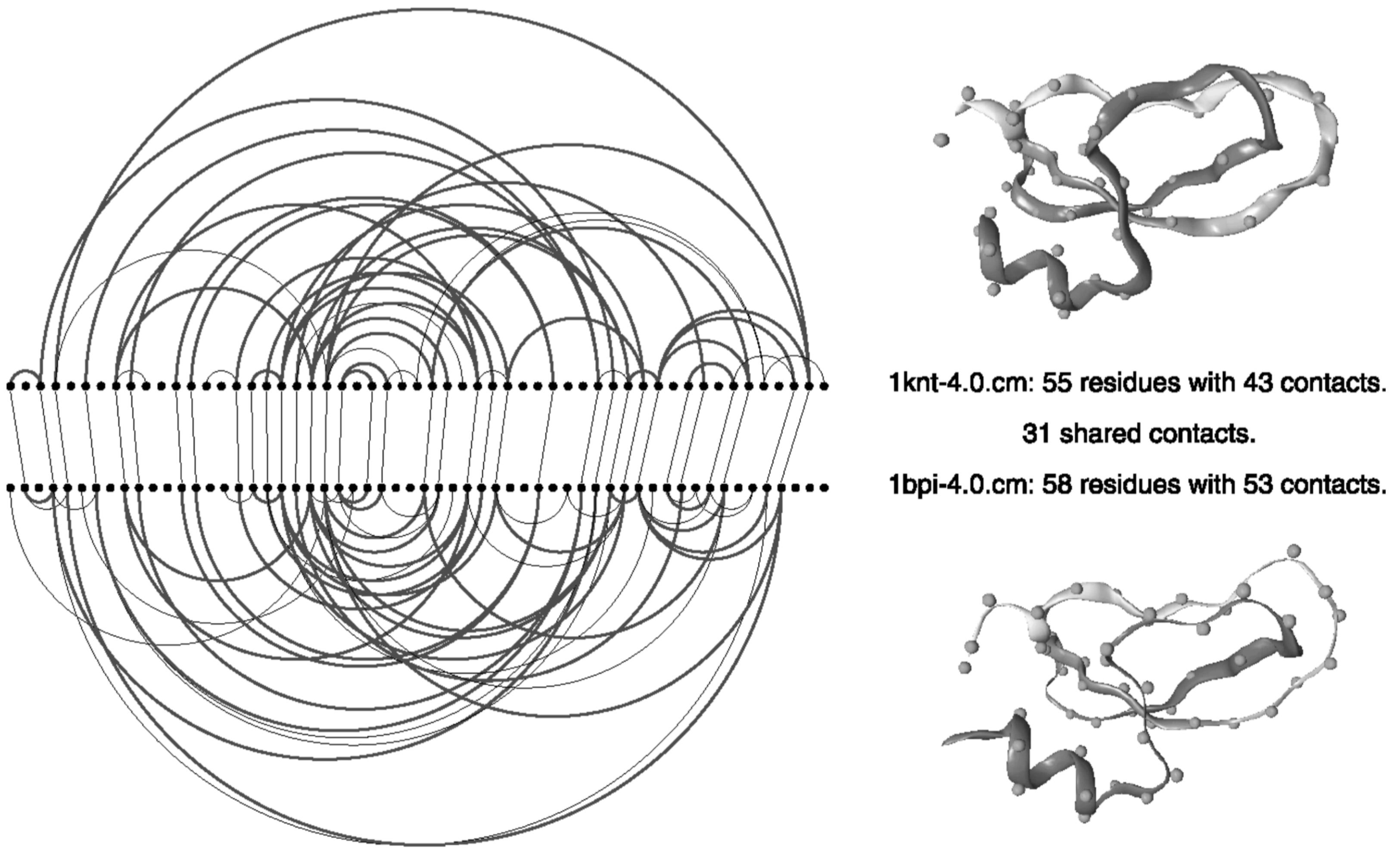


FIG. 1. An optimal alignment of two 4Å threshold contact maps of proteins 1bpi and 1knt.

Caprara, A., Carr, R., Istrail, S., Lancia, G., & Walenz, B. (2004). 1001 Optimal PDB Structure Alignments: Integer Programming Methods for Finding the Maximum Contact Map Overlap. *Journal of Computational Biology*, *11*(1), 27–52.

Contact Map Overlap: Example Instance

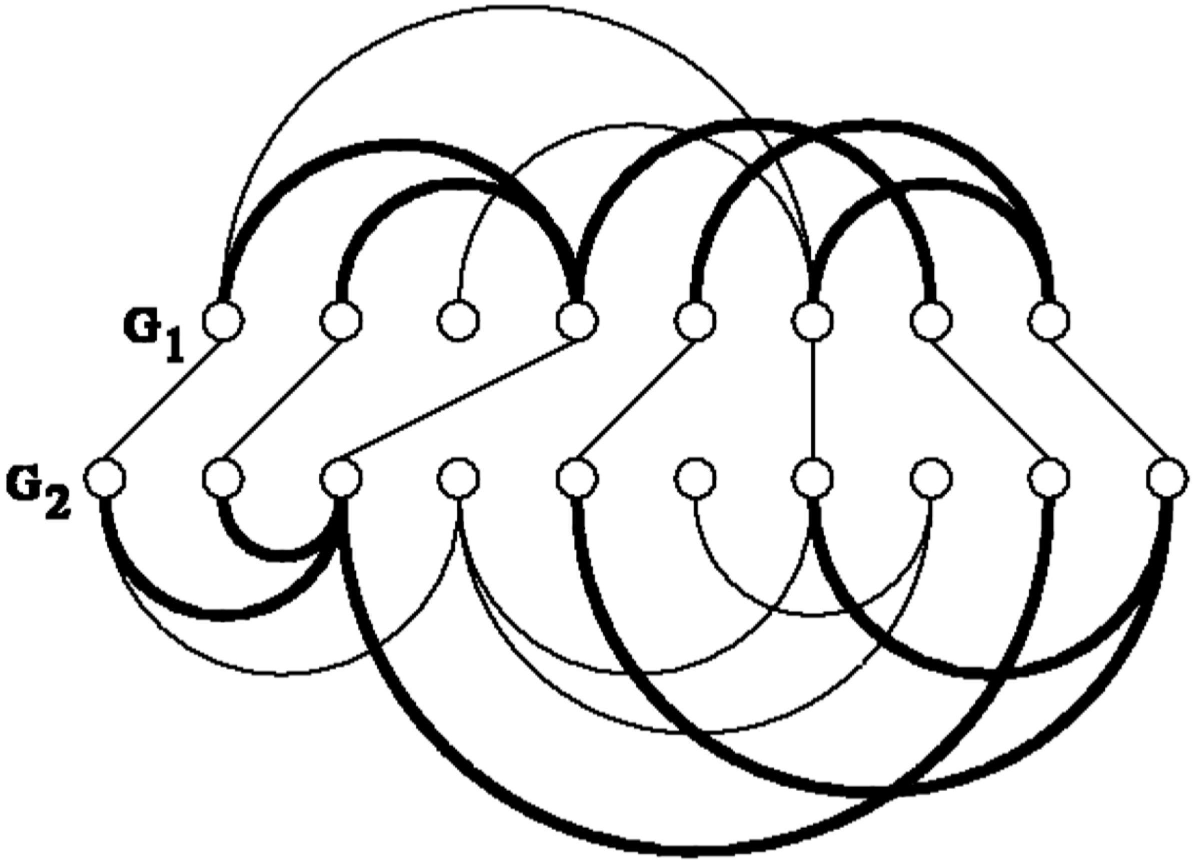


FIG. 2. An alignment of value 5.

Caprara, A., Carr, R., Istrail, S., Lancia, G., & Walenz, B. (2004). 1001 Optimal PDB Structure Alignments: Integer Programming Methods for Finding the Maximum Contact Map Overlap. *Journal of Computational Biology*, *11*(1), 27–52.

Contact Map Overlap: Equivalent Representations

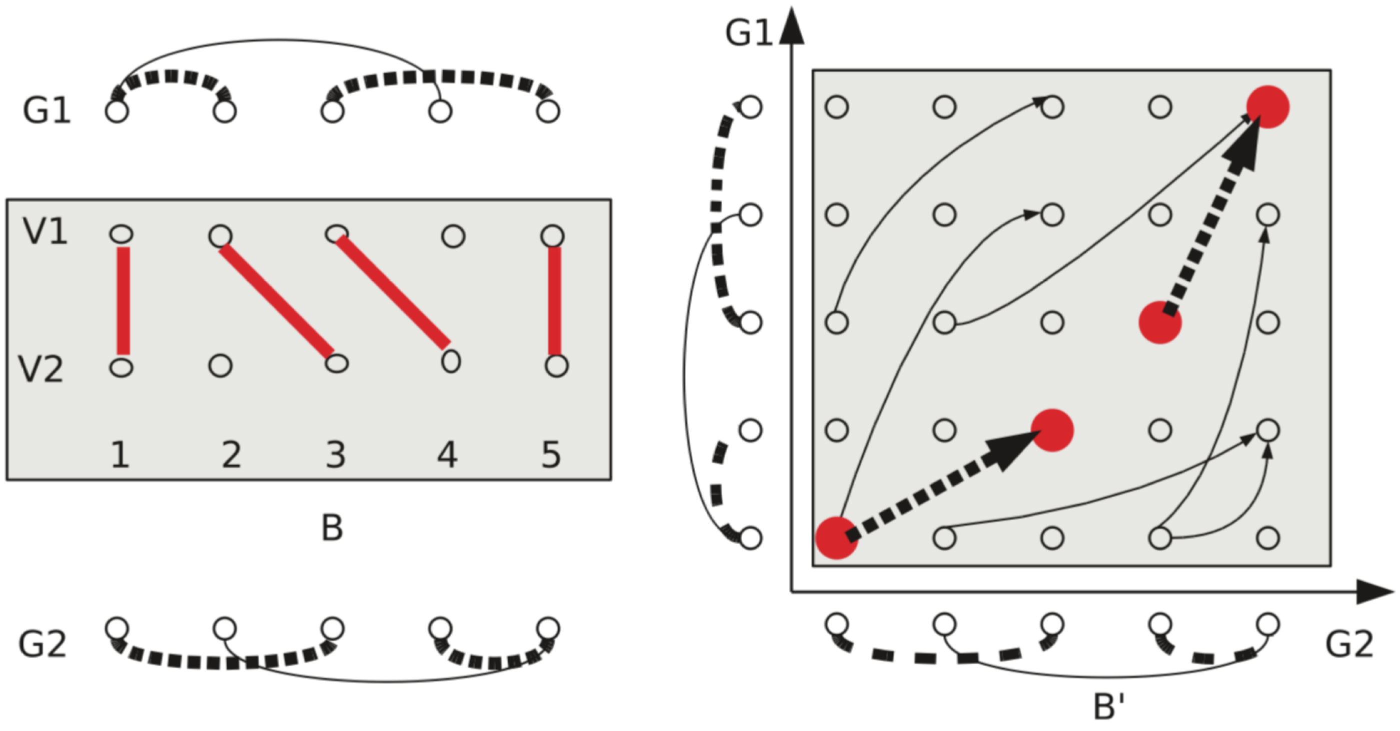


FIG. 2. Relationship between a matching in a bipartite graph B and a feasible path in the corresponding grid graph B'. **(Left)** Two contact maps $G1$ and $G2$, and a matching in the bipartite graph B (in the grey area). Note that B is a complete graph, but for the sake of simplicity only the edges of the considered matching $(M = \{(1,1)(2,3),(3,4)(5,5)\})$ are visualized. According to (1), $w(M) = 2$. **(Right)** The same matching is visualized in the grid alike graph B' as an increasing set of vertices $\{(1,1)(2,3),(3,4)(5,5)\}$ which we call a feasible path. It activates the arcs $((1,1)(2,3)$ and $(3,4)(5,5))$. The score of the path is the number of these arcs (i.e., 2 in this case).

DOI: 10.1089/cmb.2009.0196

Integer Linear Programming

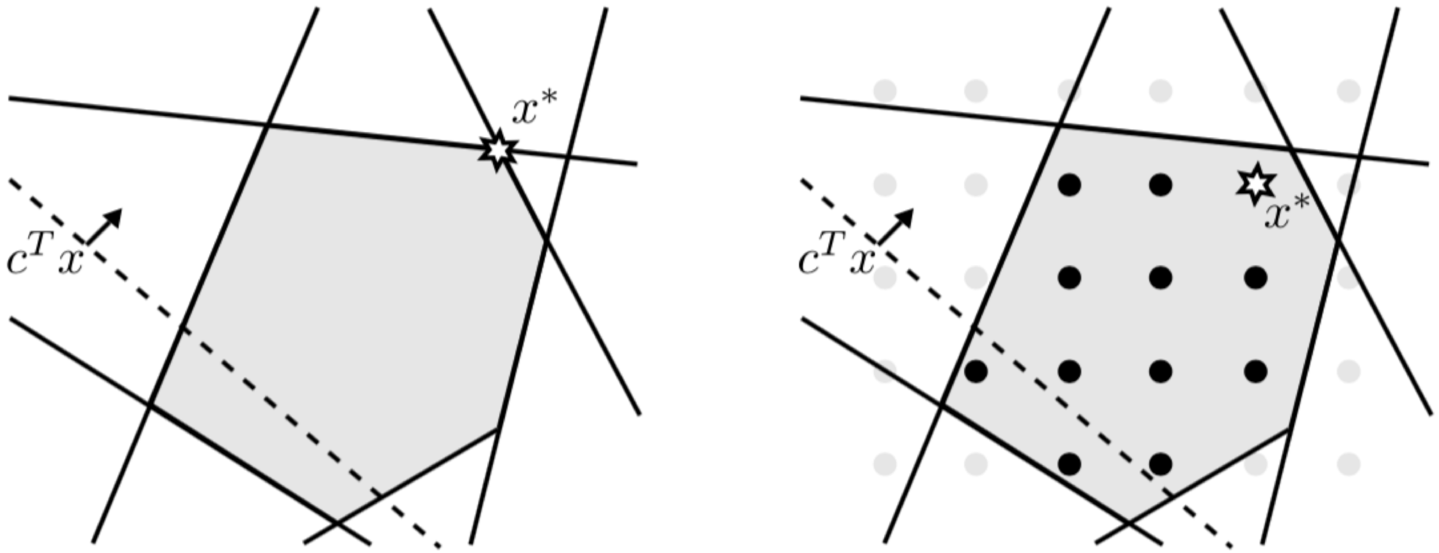


Figure 2.1: In gray, a polyhedron that is described by six constraints $a_i x \leq b_i$. The objective function $c^T x$ increases in the direction in which the arrow points. The optimal solution x^* is denoted by a star. Left: The linear program. Right: The integer linear program. Here, only the integer points within the polyhedron are feasible, which are colored black. The LP relaxation of this ILP is the LP problem that is visualized on the left side. The optimal objective function value of the LP relaxation is always an upper bound on the optimal objective function value of the ILP.

Inken Wohlers. Exact algorithms for pairwise protein structure alignment. PhD thesis, VU University Amsterdam, 2012

Separating Cutting Planes

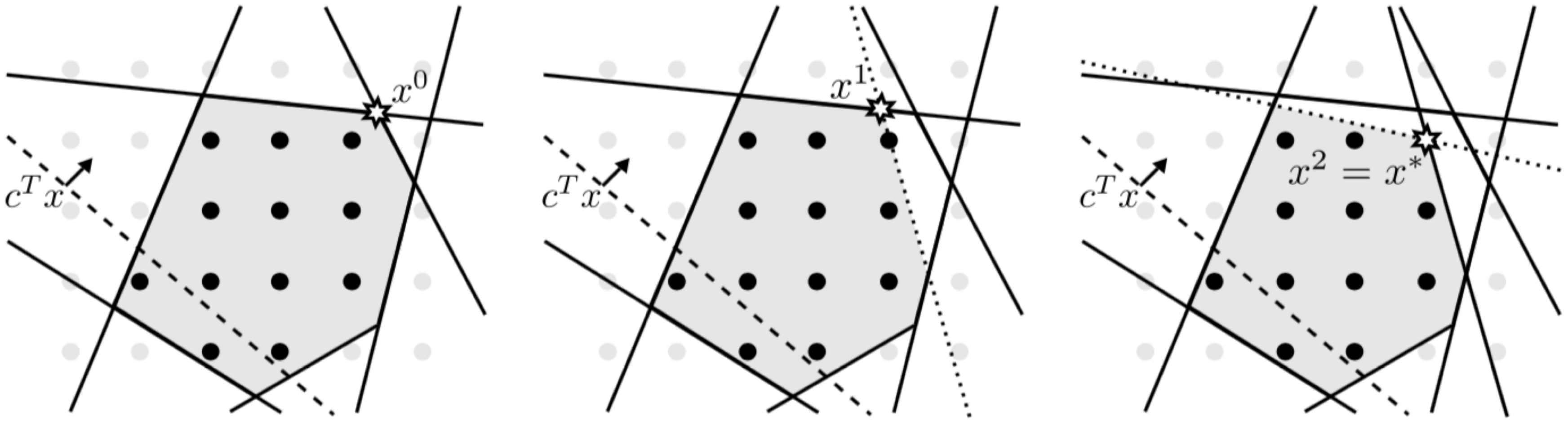


Figure 2.2: The cutting plane method solves an ILP problem. The gray area denotes the polyhedron described by the constraints of the current relaxed problem. The dashed line denotes the objective function $c^T x$ which increases in the direction in which the arrow points. The integer feasible solutions are colored black. Solving the LP relaxation, we obtain the relaxed solution x^0. After adding a cutting plane (dotted line), we obtain a new relaxed solution x^1. Finally, after adding a second cutting plane, we obtain solution x^2 which has integer value and is thus the optimal solution x^* of the ILP. A cutting plane method solves only the LP relaxation of an ILP, but here, since the optimal solution has integer value, the solution of the LP relaxation is also the solution of the ILP.

Inken Wohlers. Exact algorithms for pairwise protein structure alignment. PhD thesis, VU University Amsterdam, 2012

Cutting Plane Method

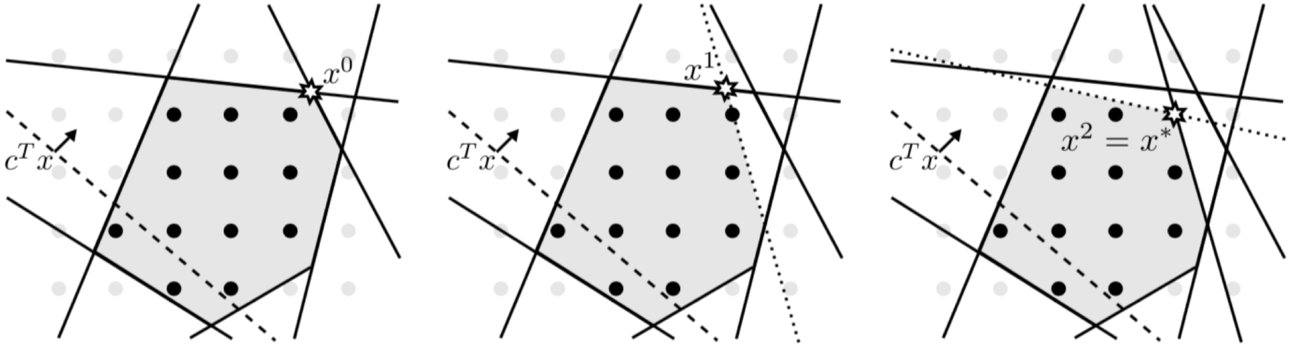


Figure 2.2: The cutting plane method solves an ILP problem. The gray area denotes the polyhedron described by the constraints of the current relaxed problem. The dashed line denotes the objective function $c^T x$ which increases in the direction in which the arrow points. The integer feasible solutions are colored black. Solving the LP relaxation, we obtain the relaxed solution x^0. After adding a cutting plane (dotted line), we obtain a new relaxed solution x^1. Finally, after adding a second cutting plane, we obtain solution x^2 which has integer value and is thus the optimal solution x^* of the ILP. A cutting plane method solves only the LP relaxation of an ILP, but here, since the optimal solution has integer value, the solution of the LP relaxation is also the solution of the ILP.

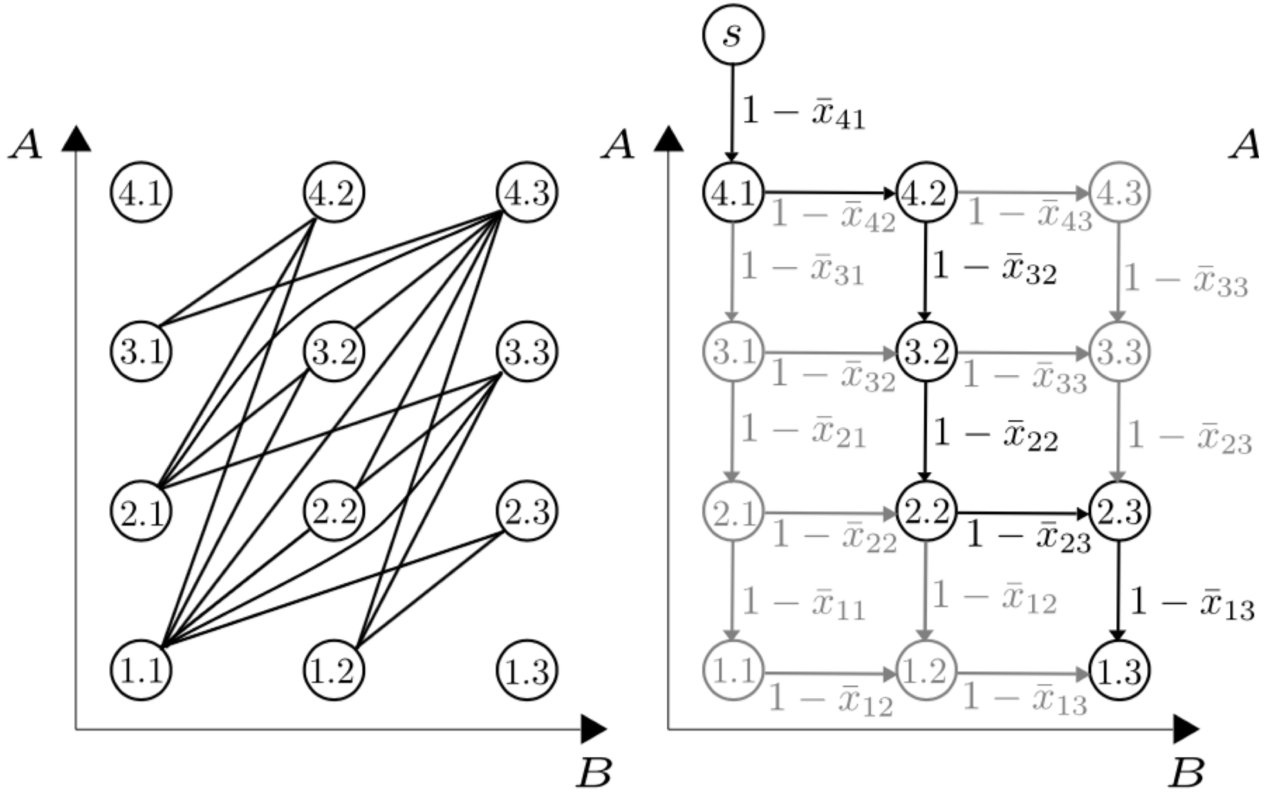
Algorithm 2 Solving an LP problem using the cutting plane method.

P // `The original problem`
P^t // `The relaxed problem in iteration` t
$t \leftarrow 0$ // `Iteration`
while `True` **do**
  Compute optimal solution x^t for P^t
  if x^t feasible for P **then**
    return x^t
  else
    Find a cutting plane $a_i x \leq b_i$ that all solutions of P satisfy, but not x^t
    $P^{t+1} \leftarrow P^t$ with additional constraint $a_i x \leq b_i$
    $t \leftarrow t + 1$
  end if
end while

Branch & Cut: Solving an ILP

- Whiteboard

Cut Separation in CMO



Inken Wohlers. Exact algorithms for pairwise protein structure alignment. PhD thesis, VU University Amsterdam, 2012

Figure 3.8: Example of the graphs in which we use shortest path computations to detect violated constraints. Left: The alignment graph $G = (V, E)$. Center: The graph $G' = (V', E')$ in which we identify a violated constraint (3.8). The shortest decreasing path is colored black, it is $C = \{4.1, 4.2, 3.2, 2.2, 2.3, 1.3\}$. If for the this path $\sum_{i.k \in C} \bar{x}_{ik} > 1$ holds, we identified a violated constraint (3.8).