Selected Topics on Combinatorial Optimization

Professor Dr. Petra Mutzel

Lehrstuhl für Algorithm Engineering

Fakultät für Informatik

TU Dortmund

PART 2: Overview of Projects

Vienna Graduate School on Computational Optimization

November 5/6, 2018

Outline Current Projects

1. **Rational Drug design (DFG SPP Algorithms for Big Data)**
   - Maximum Common Subgraph Problem for Molecule Graphs
   - Structural Clustering of Sets of Graphs
2. **Weisfeiler-Lehman for Data Analysis**
3. **Explorative Data Analysis**
4. **Adiabatic Quantum Computing**

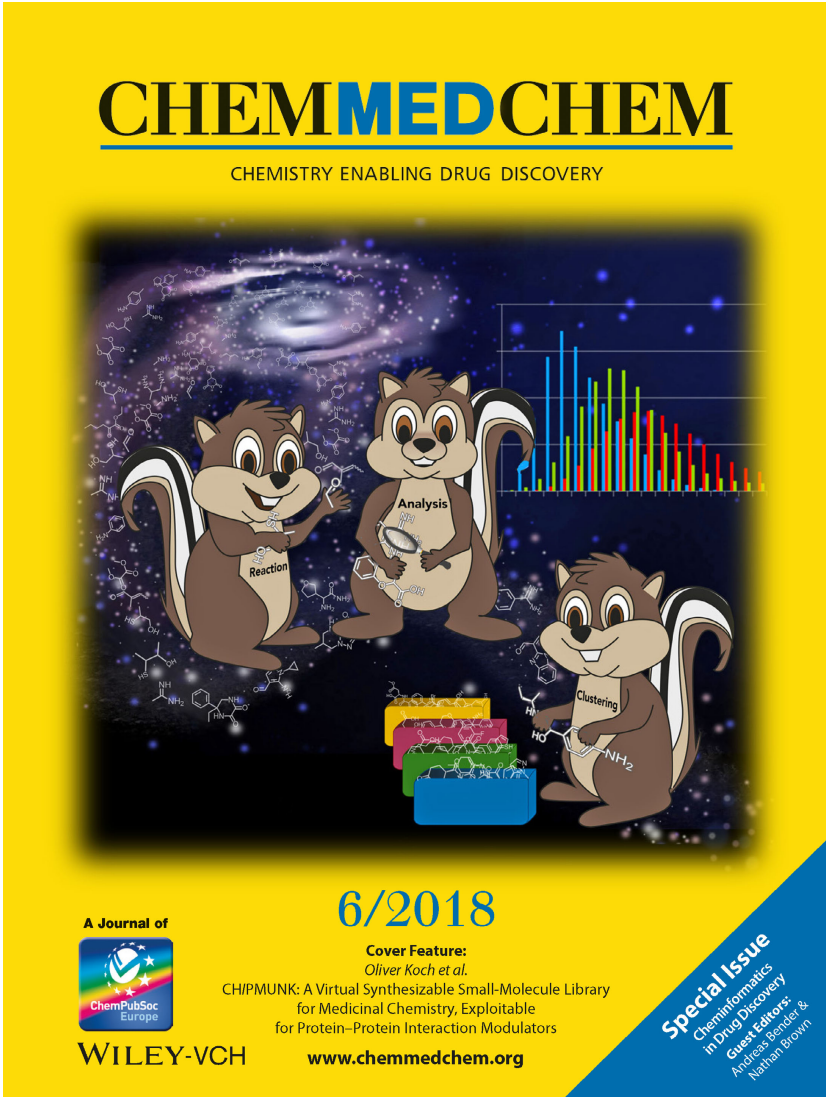
Rational Drug Design

Project: Graph-based Methods for Rational Drug Design: GraBaDrug

DFG SPP Algorithms for Big Data

jointly with Dr. Oliver Koch (Medical Chemistry)

Data Analysis: Collect, Analyse, Evaluate



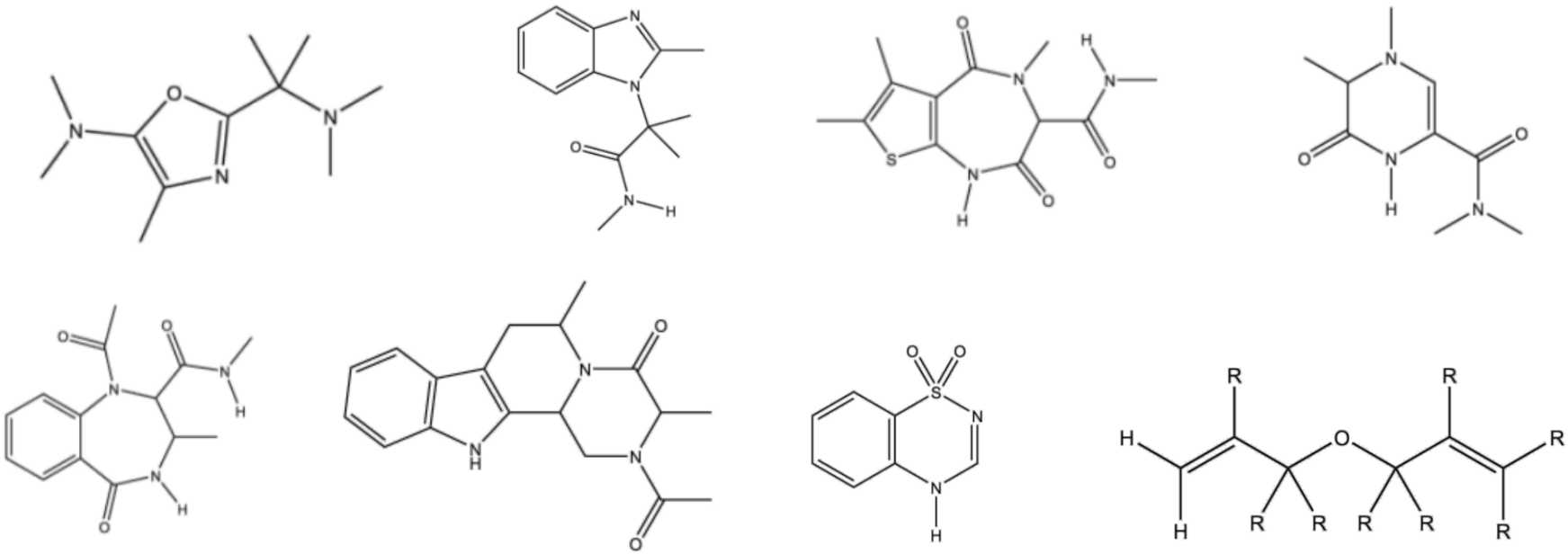
Humbeck, Weigang, Schäfer, Mutzel, Koch: CHIPMUNK: A Virtual Synthesizable Small-Molecule Library for Medicinal Chemistry, ChemMedChem 2018 + Cover Feature

Maximum Common Subgraph for Molecule Graphs

Techniques for big data

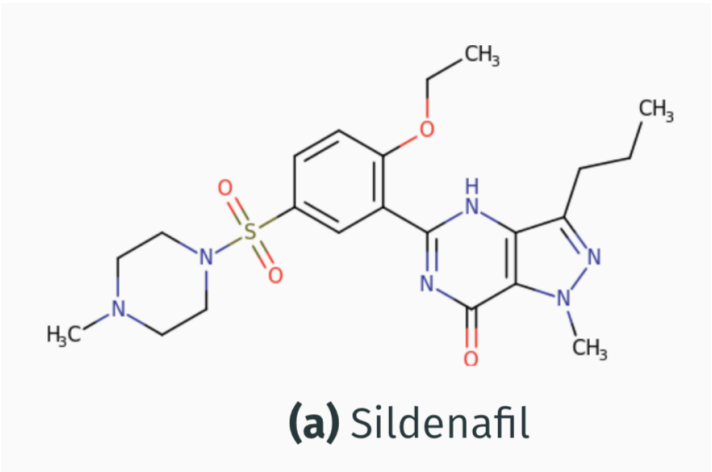
- Utilization of the structure of the input instances
- Analysis of the given problem from practice
- Problem decomposition
- Approximation
- Randomization
- Parallelisation
- Fixed-Parameter Algorithms
- ILP-Modelling

Properties of Molecule Graphs

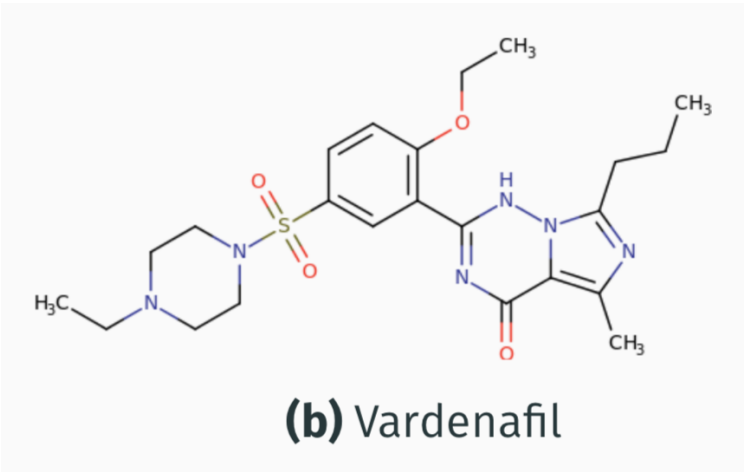


- almost always planar, often outerplanar
- bounded tree width
- bounded degree
- have vertex and edge labels (e.g. activity attributes)

Analysis of the Chemical Problem



(a) Sildenafil

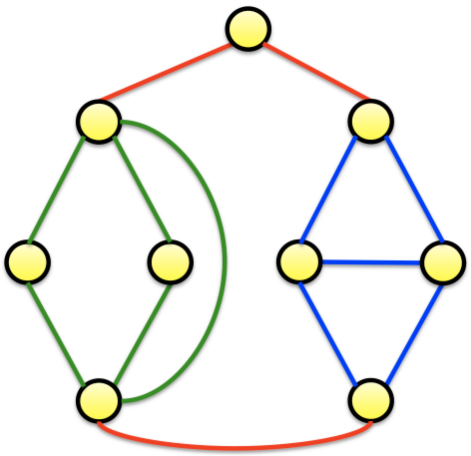


(b) Vardenafil

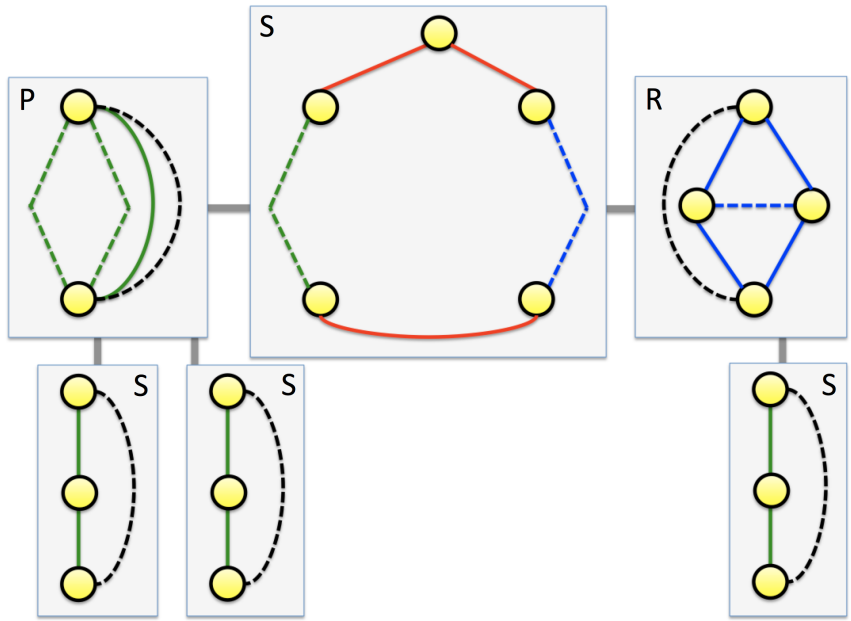
- Rings and bridges shall be preserved
- very important in Cheminformatics

→ Block-and-bridge preserving MCS → simpler

Block/Bridge MCS [Eur. J. Combinatorics 2018]



G



SPQR-tree of G

- NP-completeness of MCS for biconnected partial 2-trees with almost all vertices of degree bounded by 3
- pol. Algorithm for Block/Bridge MCS for partial 2-trees
- previously pol. algorithms for trees and outerplanar G
- Idea: BC-tree and SPQR-tree decomposition

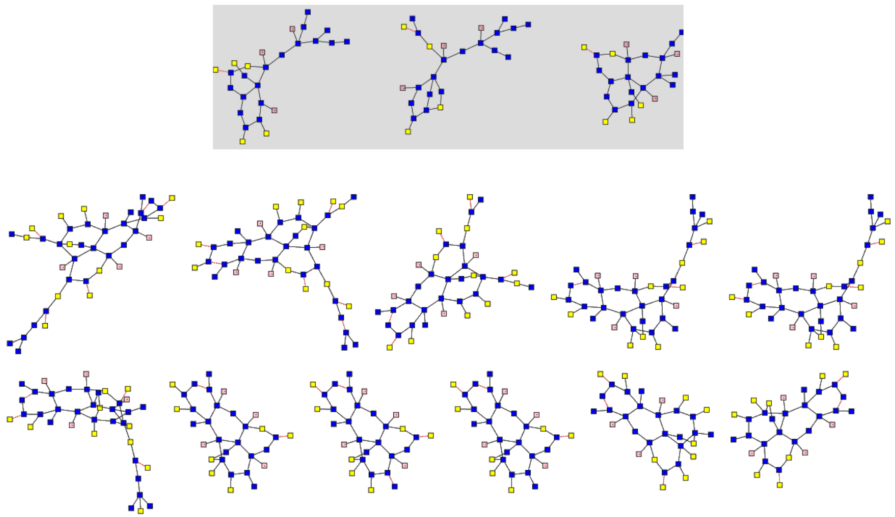
Kriege, Kurpicz, Mutzel 2018, Gutwenger, Mutzel 2000, Kriege, Mutzel 2014

Structural Clustering of Sets of Graphs

Techniques for Big Data

- Utilization of the structure of the input instances
- Analysis of the given problem from practice
- Problem decomposition
- Approximation
- Randomisation
- Parallelisation
- Fixed-Parameter Algorithms
- ILP-Modelling

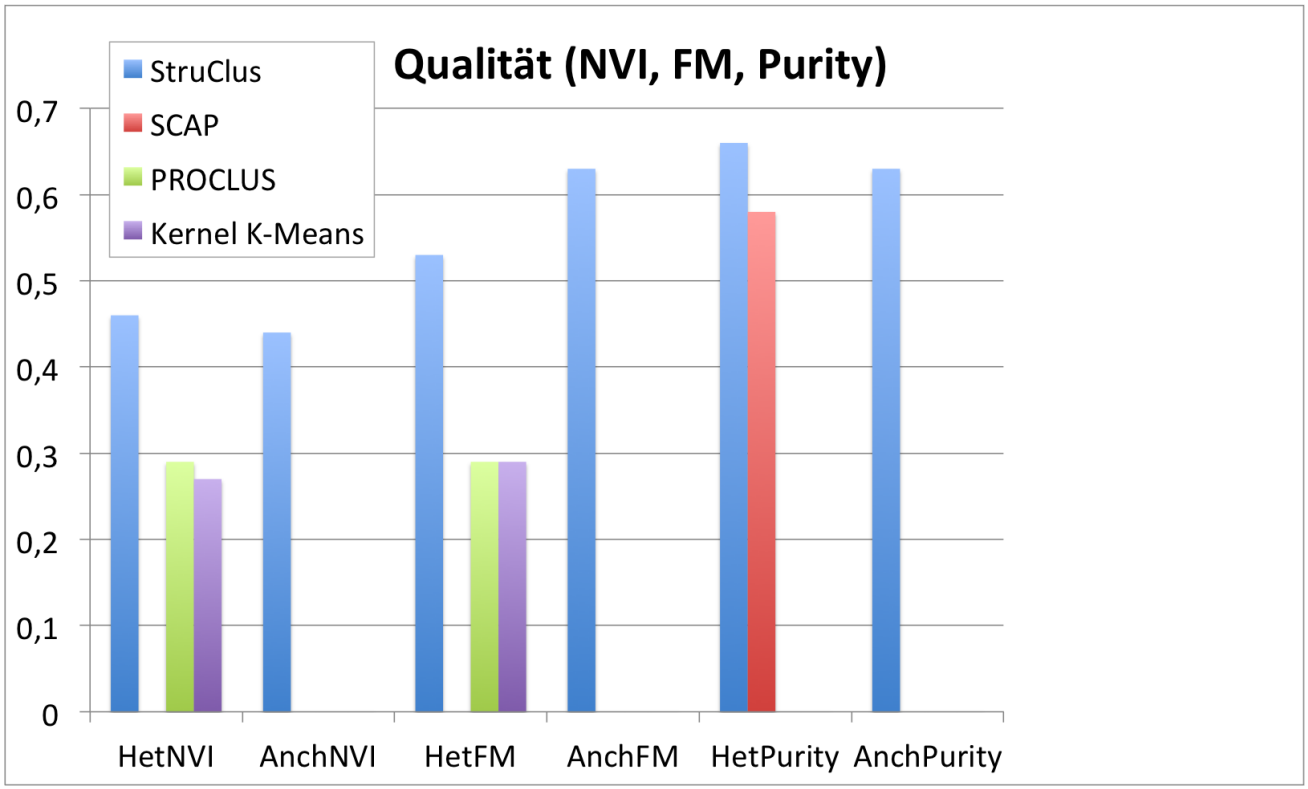
Structural Clustering (StruClus) [ADMA 2017]



- structural clustering of large sets of graphs with attributes
- sets of representatives for clusters → interpretability
- based on common subgraphs
- new error-bounded sampling strategy for *support counting*
- linear running time → scalable
- parallelisable → very fast in practice

Mutzel, Schäfer: International Conference on Advanced Data Mining and Applications 2017

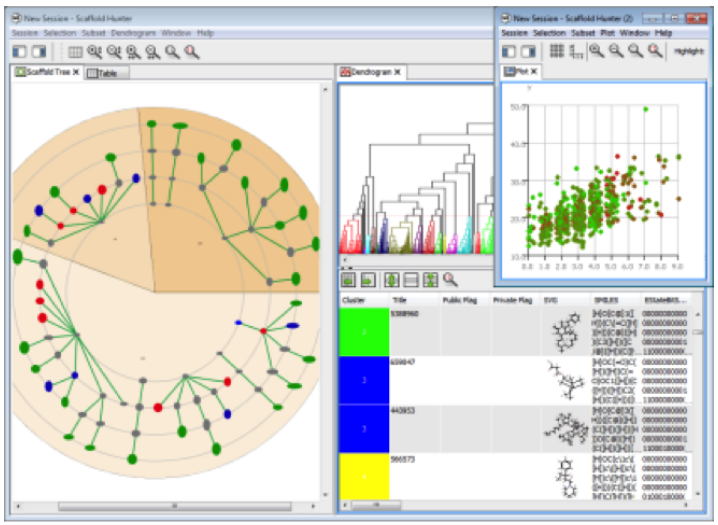
Evaluation of the StruClus Algorithm

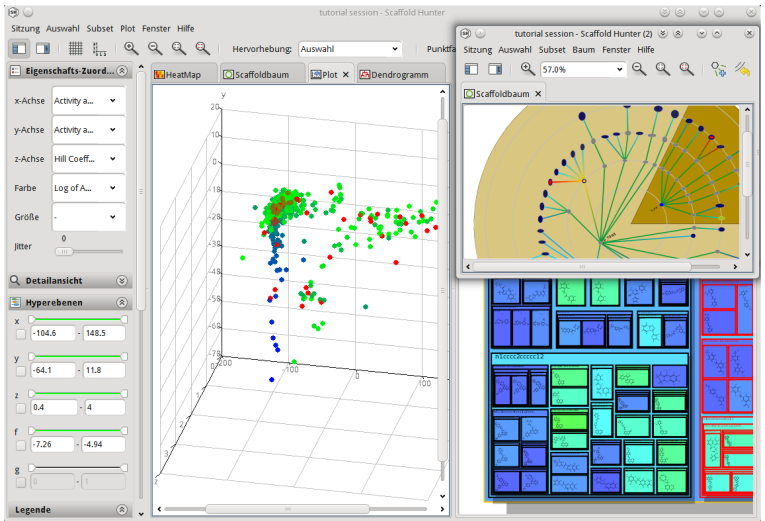


- Quality of the clustering for 3 molecule data bases:
- Heterocyclic: 10 000 (39), AnchorQuery: 65 700 (11)
- only StruClus: ChemDB (5 Mio. in 19 h)

Rational Drug Design: GraBaDrug

DFG SPP Algorithms for Big Data





- Explorative analysis of molecule data bases ← Scaffold Hunter
- Search for similar molecule structures ← Graph similarity, clustering
- Creation of virtual molecule data bases for drug design
  ← CHIPMUNK: 95 Mio. molecules with $\leq$ 700 atoms, 90 attributes

Nature Chem. Biol. 2009: Wetzel, Klein, Renner, Rauh, Oprea, Mutzel, Waldmann
ChemMedChem 2018: Humbeck, Weigang, Schäfer, Mutzel Koch + Cover Feature, u.v.m.

Weisfeiler-Lehman for Data Analysis

Weisfeiler-Lehman for Data Analysis

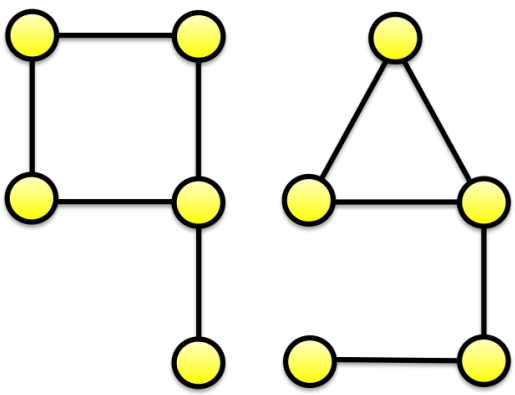
DFG SFB 876:
Providing Information by Resource-Constrained Data Analysis

Weisfeiler-Lehman Algorithm

- Coloring algorithm with the goal of vertex classification

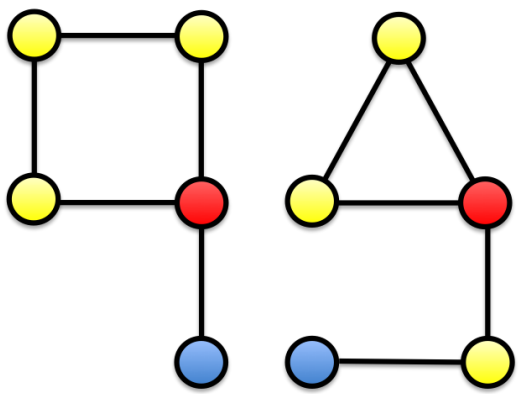
WL-Algorithm

- **Initial:** All vertices v have the same color.
- **Iteration:** Further separation of identically colored vertex sets based on color histograms of neighbors.



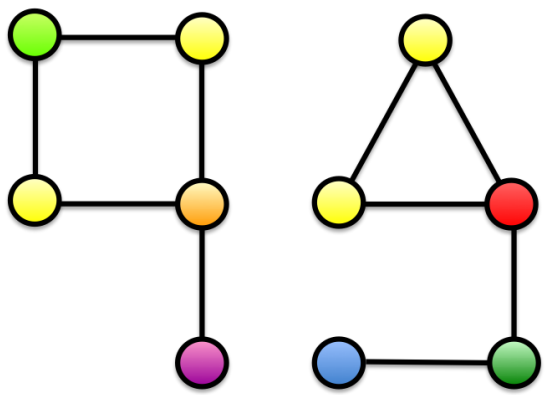
G H

Initialization



G H

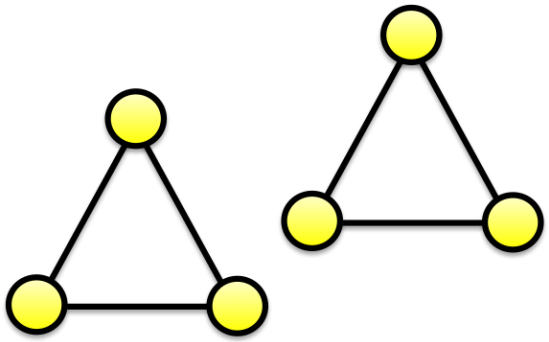
1. Iteration



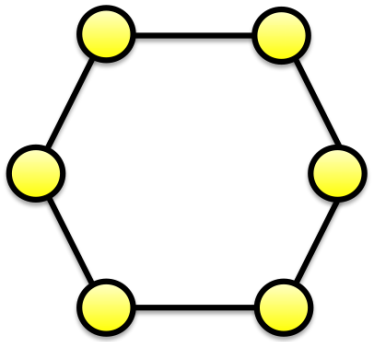
G H

2. Iteration

Weisfeiler-Lehman Algorithm



G

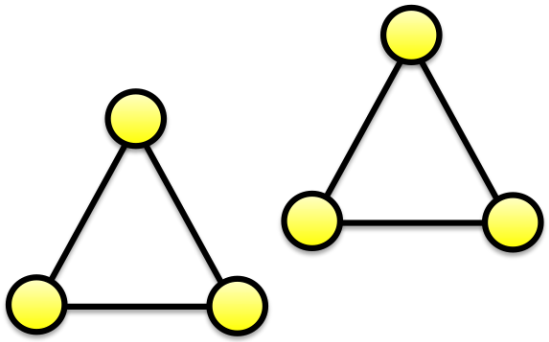


H

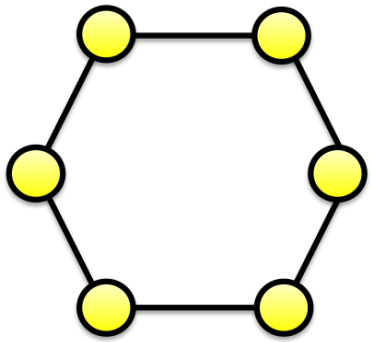
Properties of WL

- If G and H have different colors $\Longrightarrow G \not\simeq H$

Weisfeiler-Lehman Algorithm



G

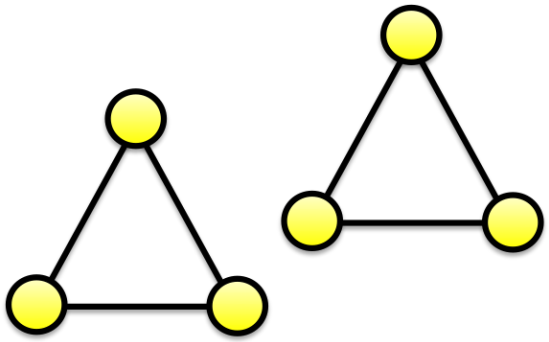


H

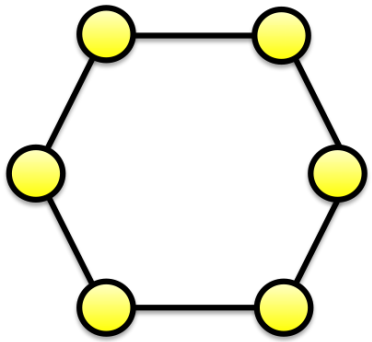
Properties of WL

- If G and H have different colors $\Longrightarrow G \not\simeq H$
- WL can identify all trees, i.e. non-isomorphic trees get different colors

Weisfeiler-Lehman Algorithm



G

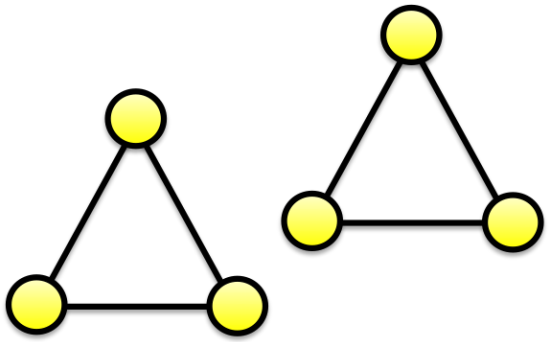


H

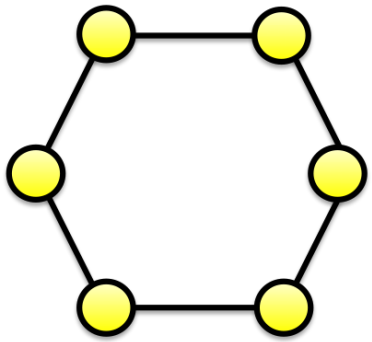
Properties of WL

- If G and H have different colors $\Longrightarrow G \not\simeq H$
- WL can identify all trees, i.e. non-isomorphic trees get different colors
- random graphs G will be identified correctly with high probability

Weisfeiler-Lehman Algorithm



G

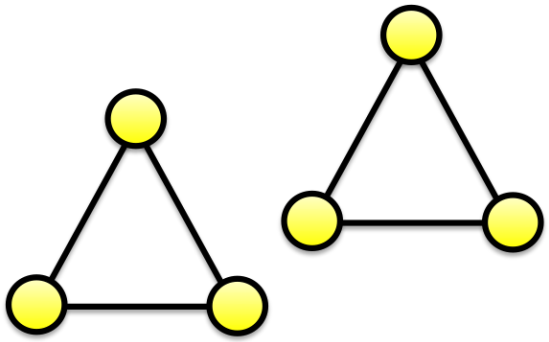


H

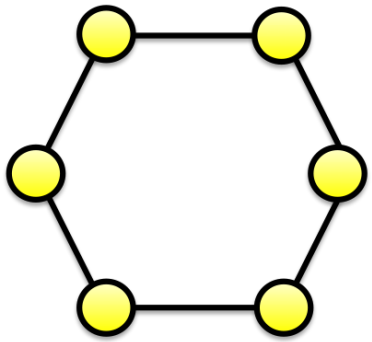
Properties of WL

- If G and H have different colors $\Longrightarrow G \not\simeq H$
- WL can identify all trees, i.e. non-isomorphic trees get different colors
- random graphs G will be identified correctly with high probability
- running time: $O((|V| + |E|) \log |V|)$

Weisfeiler-Lehman Algorithm



G



H

Properties of WL

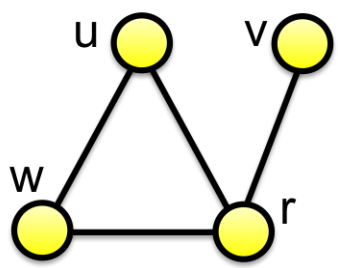
- If G and H have different colors $\Longrightarrow G \not\simeq H$
- WL can identify all trees, i.e. non-isomorphic trees get different colors
- random graphs G will be identified correctly with high probability
- running time: $O((|V| + |E|) \log |V|)$
- cannot separate regular graphs (same degree) $\rightarrow$ same color

k-dimensional Weisfeiler-Lehman Algorithm

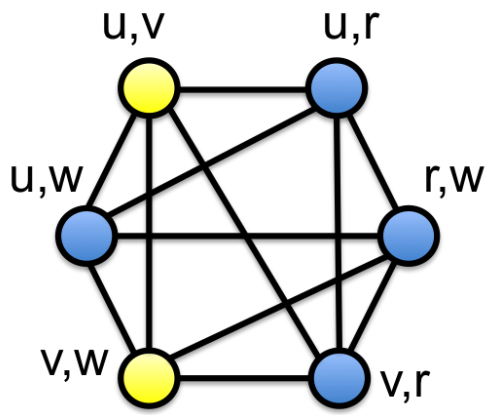
Two k-vertex sets are neighbors if they differ in only one element. [1]

k-WL Algorithm

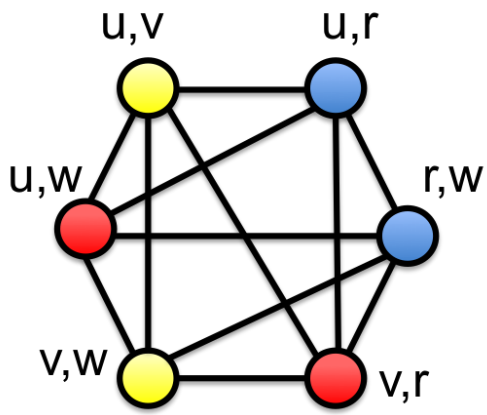
- **Initial:** k-sets U, W get the same color if $G[U] \simeq G[W]$.
- **Iteration:** Two identically colored k-sets U und W get different colors if there exists a color c so that U and W have a different set of neighbors of color c.



G



Initialization 2-WL



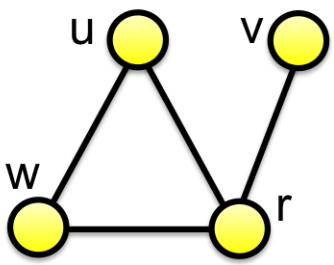
1. Iteration 2-WL

[1] defined on k-tuples, for visualization reasons here for k-sets

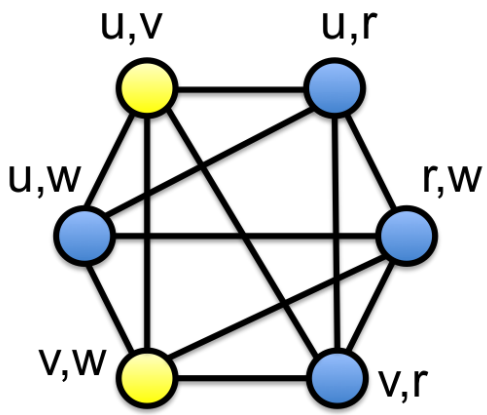
k-WL Algorithm

Properties of the k-WL

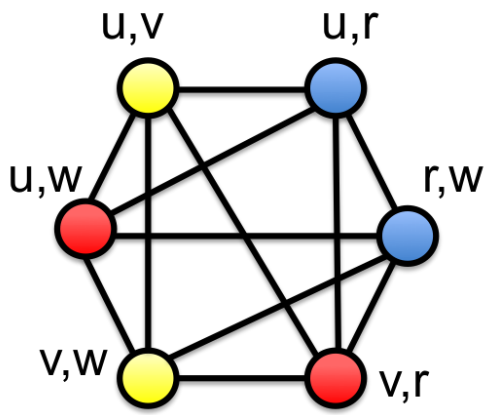
- k-WL ($k > 1$) is stronger than 1-WL
- exact for large enough k (identifies each graph)
- graph isomorphism approach by Babai uses $k = O(\log n)$
- there exist graph classes for which $k = \theta(n)$ is necessary
- running time: $O(k^2|V|^{k+1} \log |V|)$



G



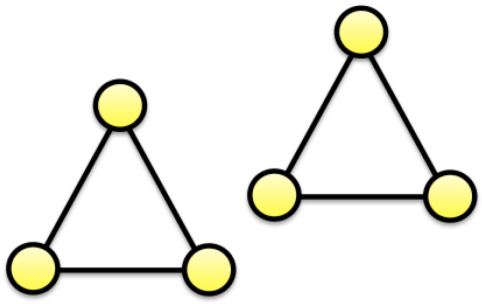
Initialization 2-WL



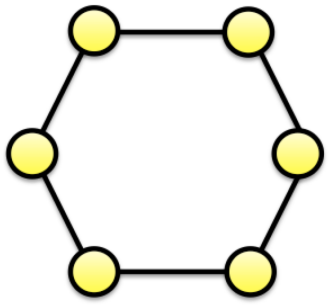
1. Iteration 2-WL

k-WL Algorithm

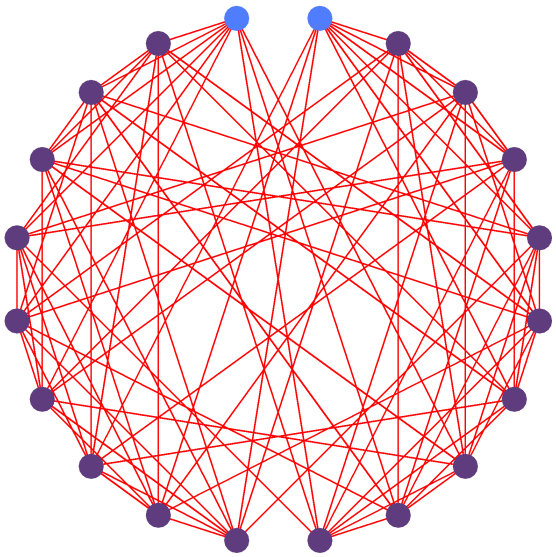
k-WL can separate regular graphs: $k = 3$

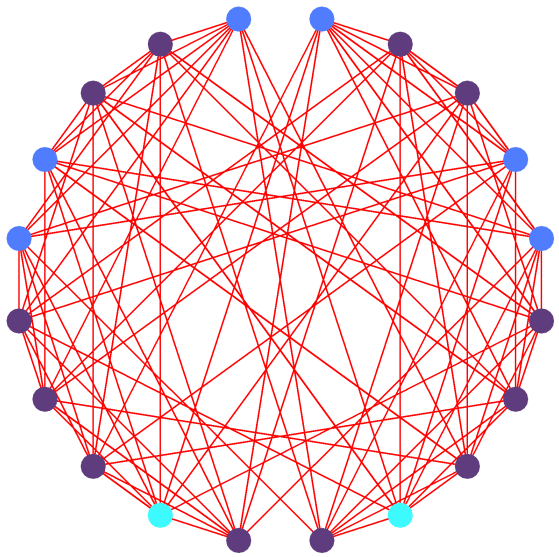


G



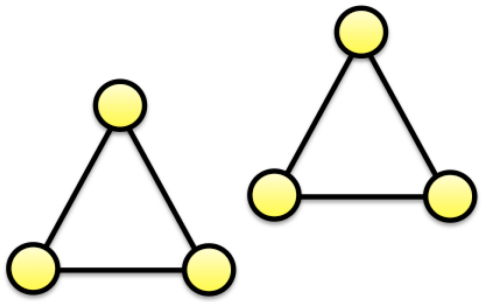
H



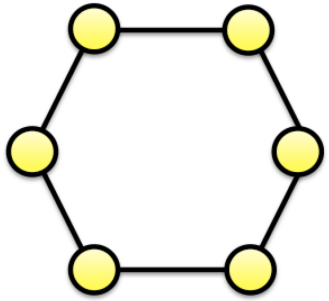


k-WL Algorithm

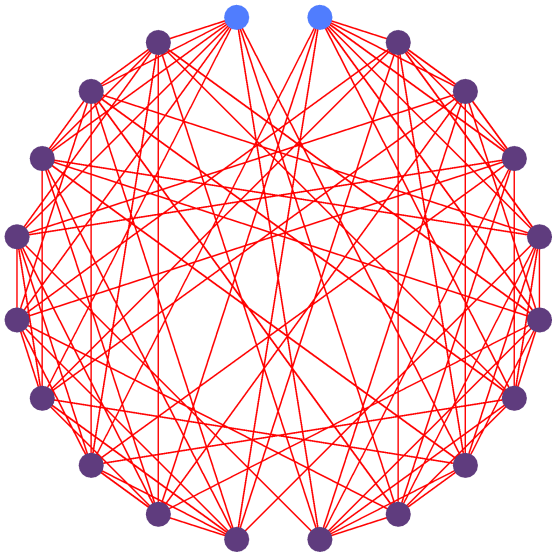
k-WL can separate regular graphs: $k = 3$

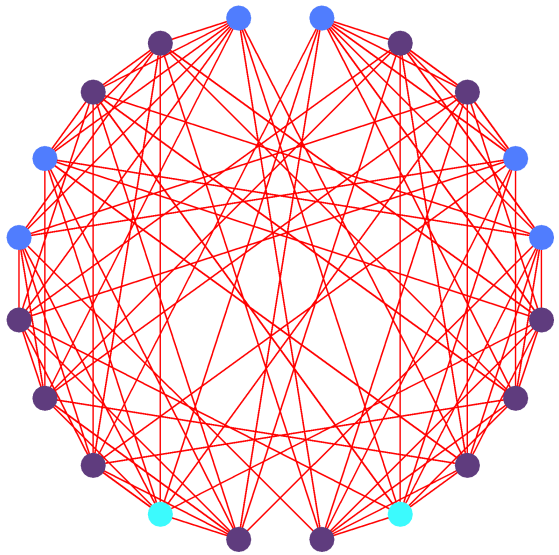


G



H





$\rightarrow$ strong for $k \geq 2$, but also very slow

Local k-WL Algorithm [ICDM 2017]

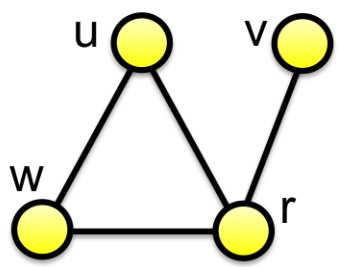
Techniques for Big Data

- Utilization of the structure of the input instances
- Analysis of the given problem from practice
- Problem decomposition
- Approximation
- Randomization
- Parallelisation
- Fixed-Parameter Algorithms
- ILP-Modelling

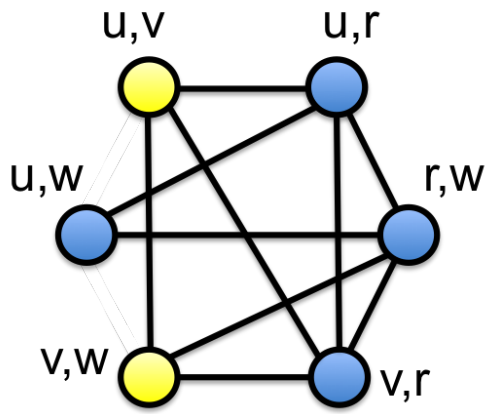
Local k-WL Algorithm (k-LWL) [ICDM 2017]

Idea: Define neighbors of the k-sets depending on graph structure

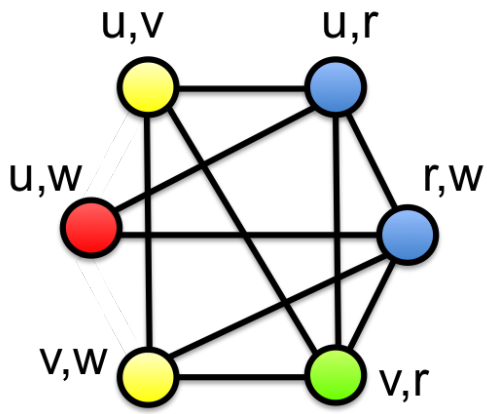
- Two k-sets U und W are neighbors, if they differ in only one element and there exists at least one edge from a vertex in U (resp. W) to a vertex in W (resp. U).
- $\rightarrow$ takes sparsity of the original graph into account
- $\rightarrow$ considers local and global graph properties



G



Initialization 2-LWL



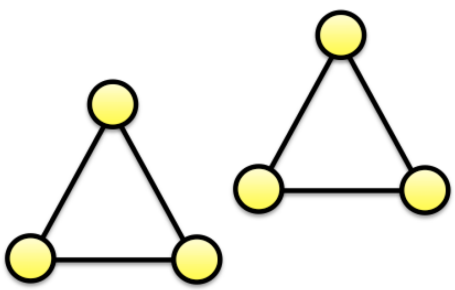
1. Iteration 2-LWL

Morris, Kersting, Mutzel 2017

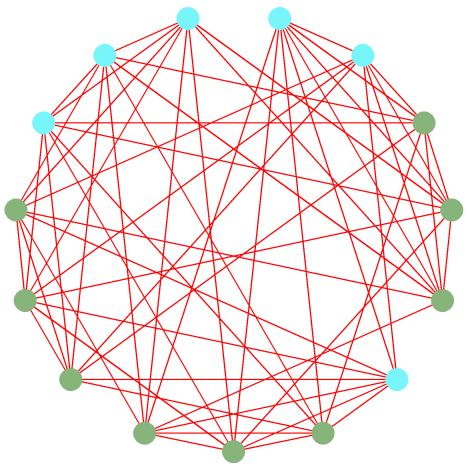
Comparison: Local k-LWL vs. k-WL

Running time

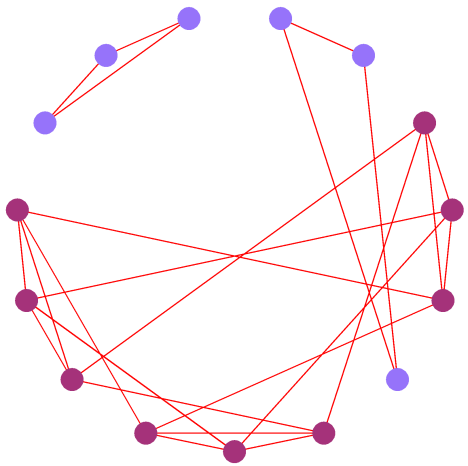
- k-sets of the k-LWL have much less neighbors
- much faster than k-WL



G



2-WL



2-LWL

Morris, Kersting, Mutzel 2017

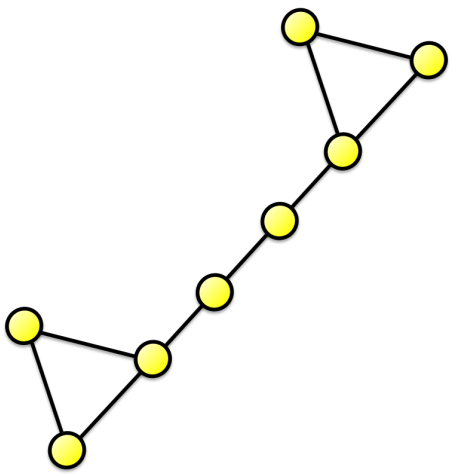
Comparison: Local k-LWL vs. k-WL

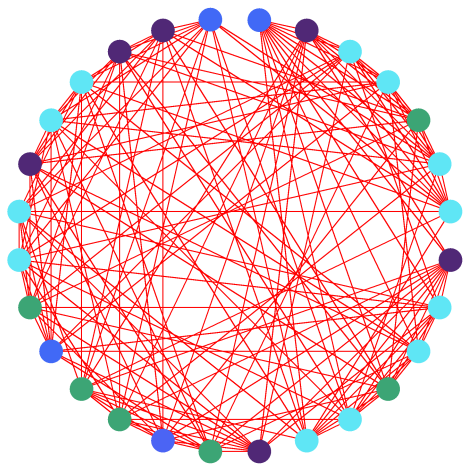
Separation Strength: Conjecture (for connected G)

Comparison: Local k-LWL vs. k-WL

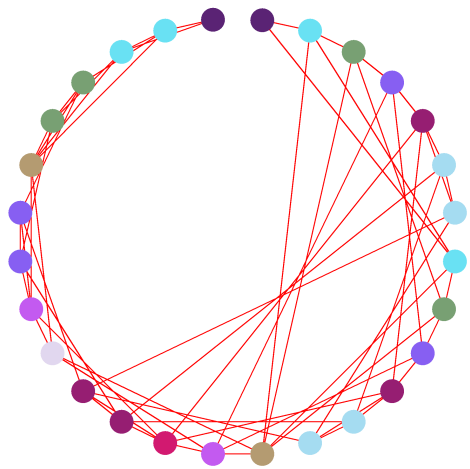
Separation Strength: Conjecture (for connected G)

- Local k-LWL refines at least as much as k-WL.





2-WL: 5 color classes
$12, 6, 6, 3, 1$

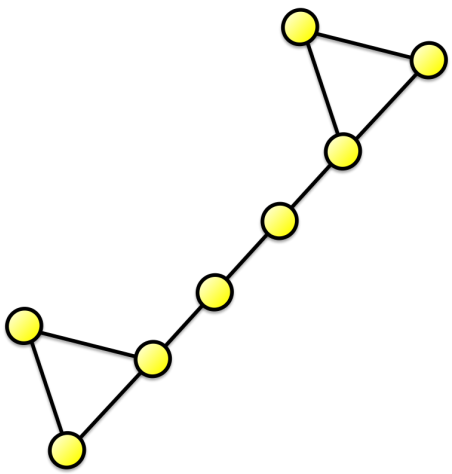


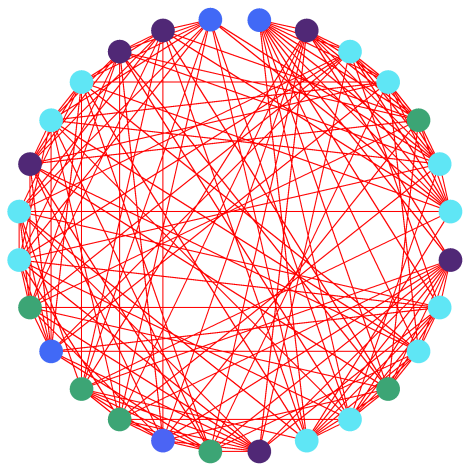
2-LWL: 10 color classes
$4, 4, 4, 4, 2, 4, 2, 2, 1, 1$

Comparison: Local k-LWL vs. k-WL

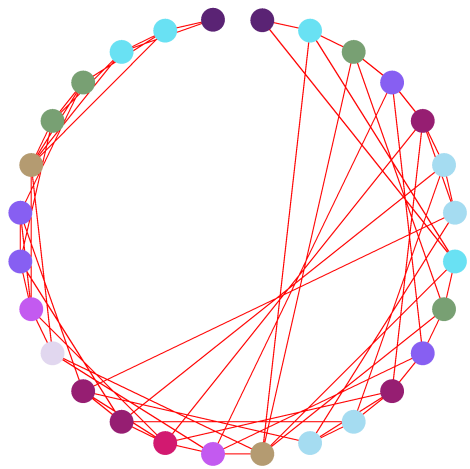
Separation Strength: Conjecture (for connected G)

- Local k-LWL refines at least as much as k-WL.
- If k-WL can separate two graphs, then also k-LWL.





2-WL: 5 color classes
$12, 6, 6, 3, 1$

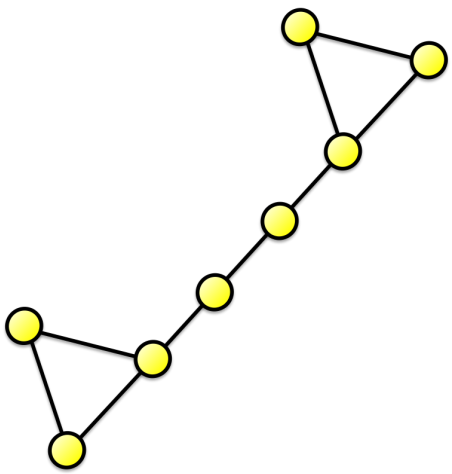


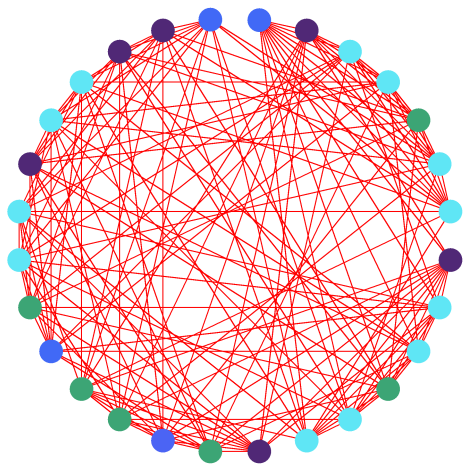
2-LWL: 10 color classes
$4, 4, 4, 4, 2, 4, 2, 2, 1, 1$

Comparison: Local k-LWL vs. k-WL

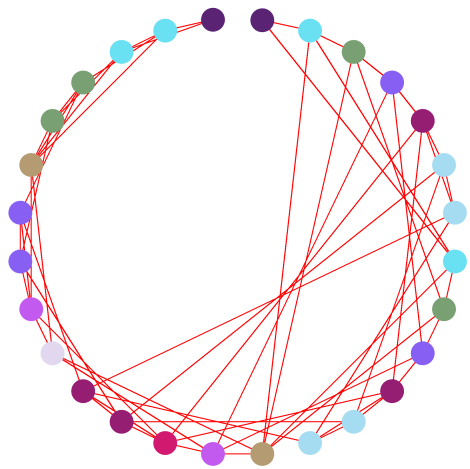
Separation Strength: Conjecture (for connected G)

- Local k-LWL refines at least as much as k-WL.
- If k-WL can separate two graphs, then also k-LWL.
- Local k-LWL is stronger than k-WL.





2-WL: 5 color classes
$12, 6, 6, 3, 1$

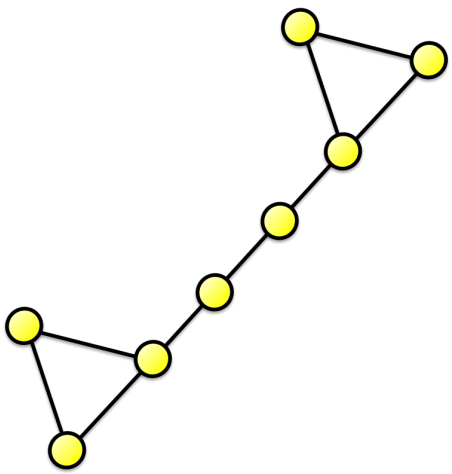


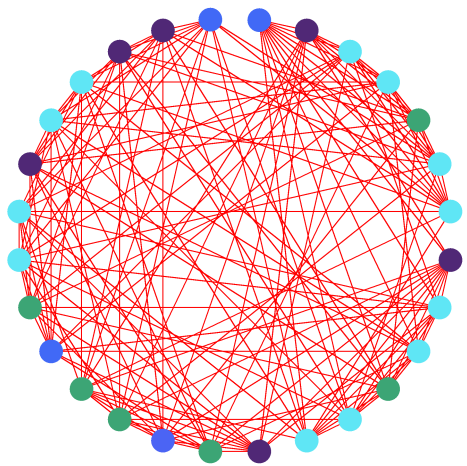
2-LWL: 10 color classes
$4, 4, 4, 4, 2, 4, 2, 2, 1, 1$

Comparison: Local k-LWL vs. k-WL

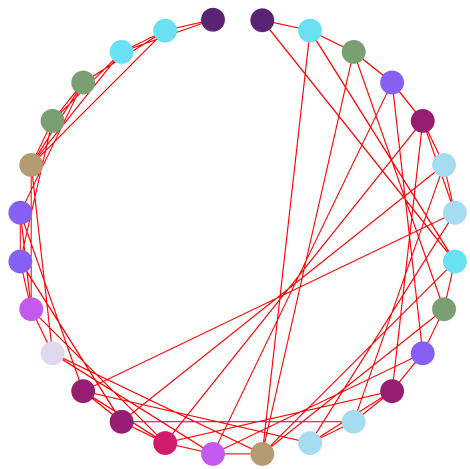
Separation Strength: Conjecture (for connected G)

- Local k-LWL refines at least as much as k-WL.
- If k-WL can separate two graphs, then also k-LWL.
- Local k-LWL is stronger than k-WL.
- Sherali-Adams Relaxation of k-LWL is stronger than that of k-WL.



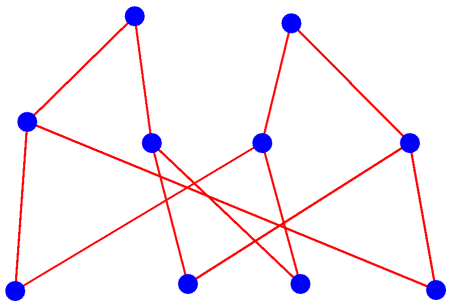


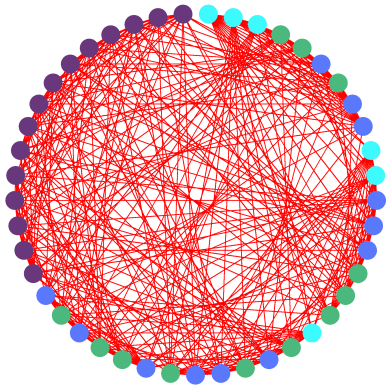
2-WL: 5 color classes
$12, 6, 6, 3, 1$



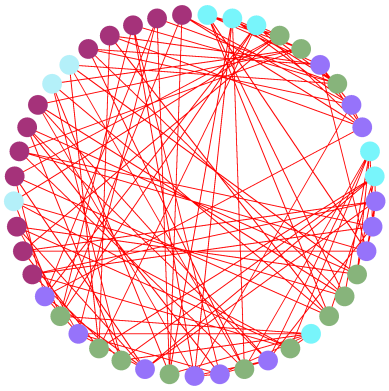
2-LWL: 10 color classes
$4, 4, 4, 4, 2, 4, 2, 2, 1, 1$

Cai, Führer, Immerman - Graphs for lower bound

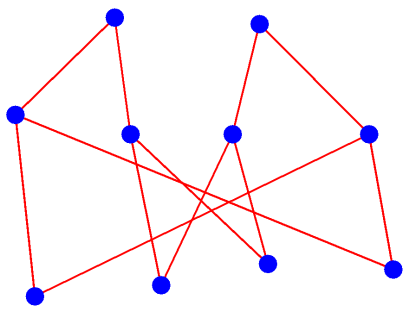


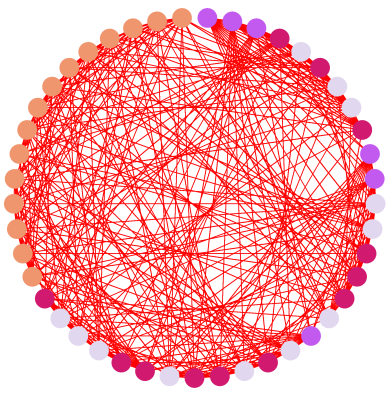


2-WL: 4 color classes

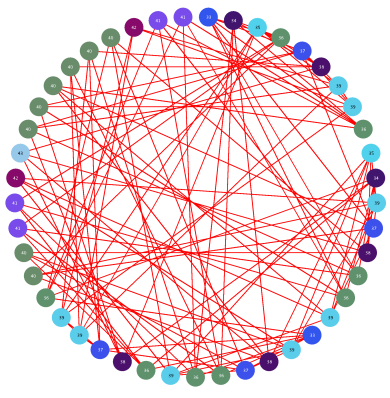


2-LWL: 5 color classes





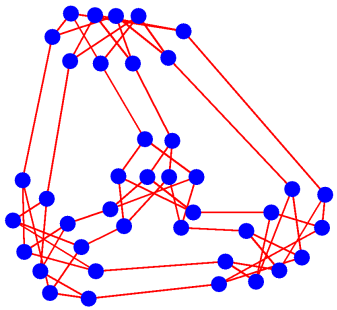
2-WL: 4 color classes

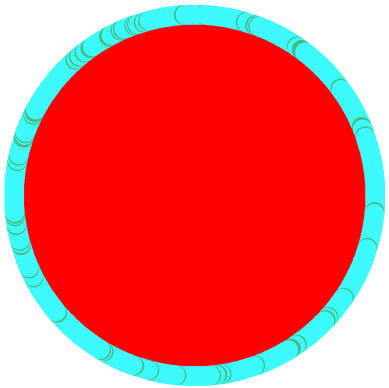


2-LWL: 11 color classes

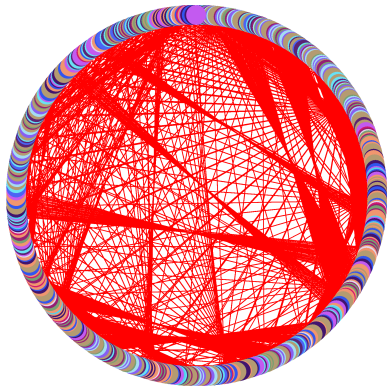
$\rightarrow$2-WL cannot separate graphs, but 2-LWL

Immerman, Grohe - Graphs for lower bound



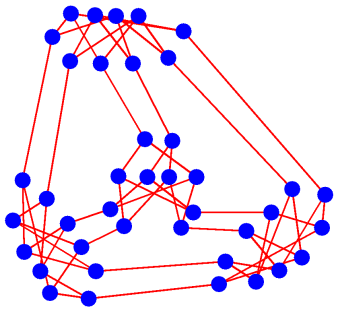


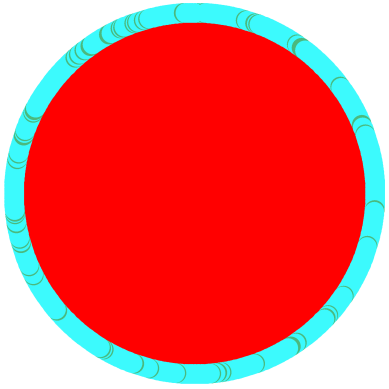
2-WL: 2 color classes



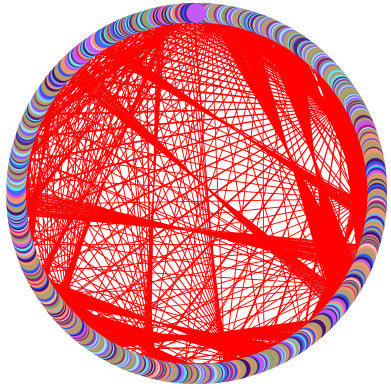
2-LWL: 15 color classes

Immerman, Grohe - Graphs for lower bound

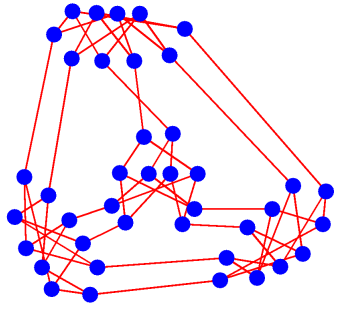


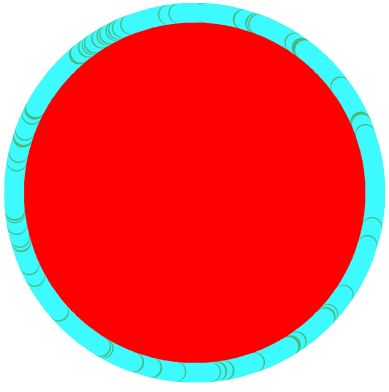


2-WL: 2 color classes

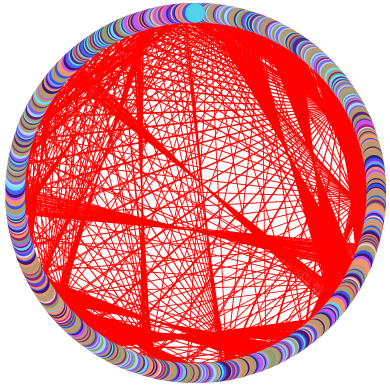


2-LWL: 15 color classes





2-WL: 2 color classes



2-LWL: 15 color classes

$\rightarrow$ 2-LWL refines more but unfortunately does not separate

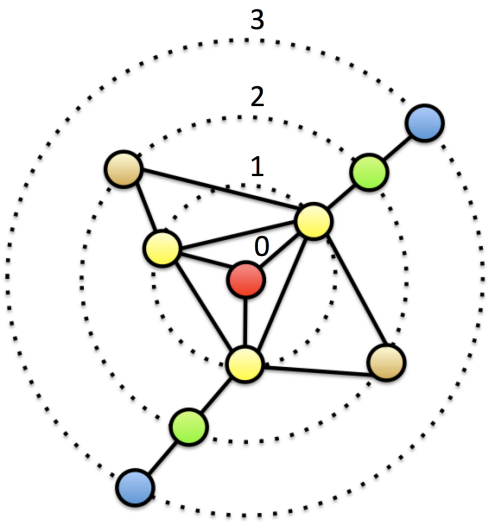
Local k-LWL Algorithm [ICDM 2017]

Idea: Increase scalability by sampling

- Sample a subset of the k-sets
- Explore the h-neighborhood around these sets
- Run the local k-LWL on each of the h-neighborhoods

Lemma

These k-sets get the same color as the k-LWL after h rounds.

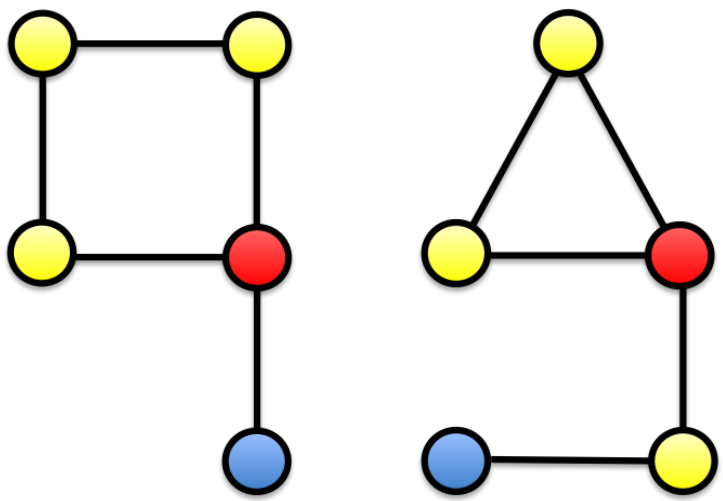


Morris, Kersting, Mutzel 2017

WL, k-WL, k-LWL for Data Analysis

Idea: Combination of WL with similarity measures

- Construct feature vector for each graph
- e.g. after each round: sort the vertices according to colors, vector gets information of number of vertices with color c
- append these (possibly weighted) vectors to each other

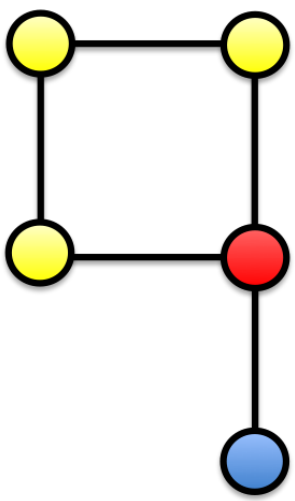


$\Phi(G) = (3, 1, 1)$ $\Phi(H) = (3, 1, 1)$

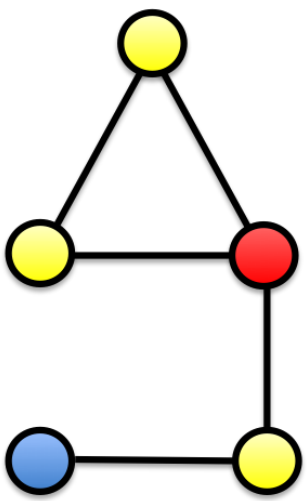
WL, k-WL, k-LWL for Data Analysis

Idea: Combination of WL with similarity measures

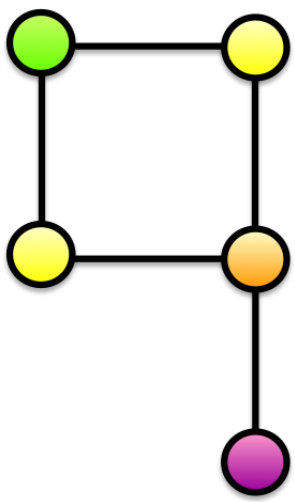
- Construct feature vector for each graph
- e.g. after each round: sort the vertices according to colors, vector gets information of number of vertices with color c
- append these (possibly weighted) vectors to each other



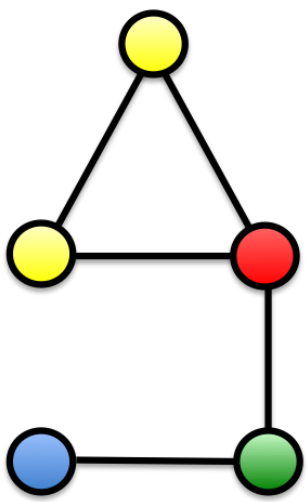
$\Phi(G) = (3, 1, 1)$



$\Phi(H) = (3, 1, 1)$



$\Phi(G) = (3, 1, 1, 2, 0, 0, 1, 1, 1, 0)$

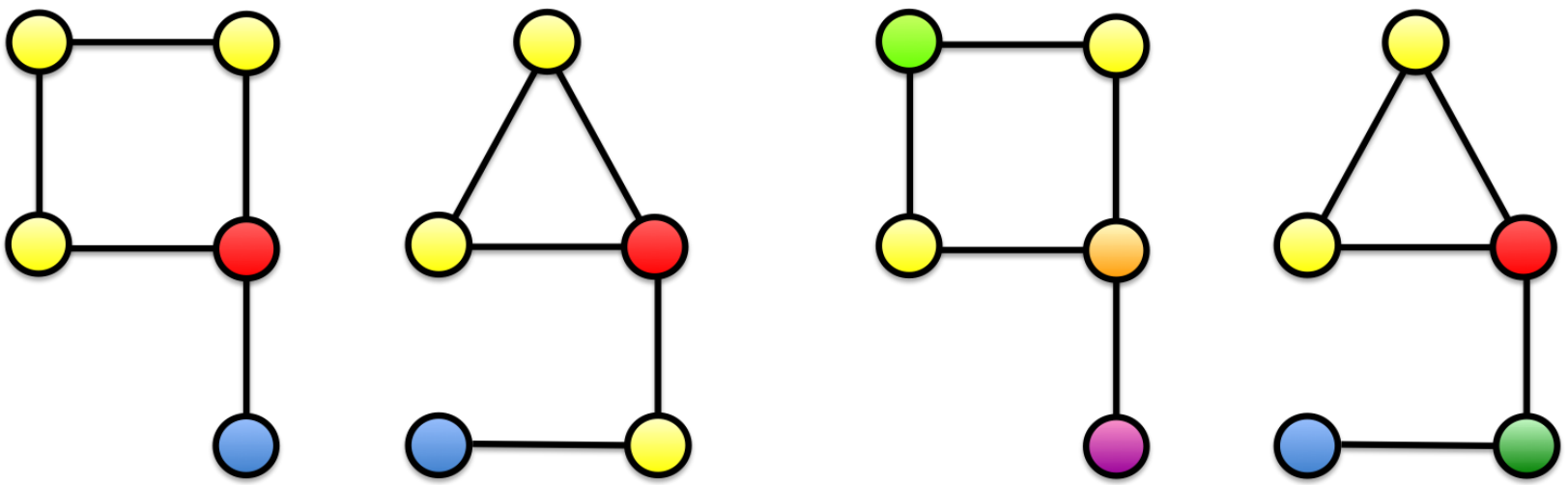


$\Phi(H) = (3, 1, 1, 2, 1, 1, 0, 0, 0, 1)$

WL, k-WL, k-LWL for Data Analysis

Idea: Combination of WL with similarity measures

- Construct feature vector for each graph
- e.g. after each round: sort the vertices according to colors, vector gets information of number of vertices with color c
- append these (possibly weighted) vectors to each other

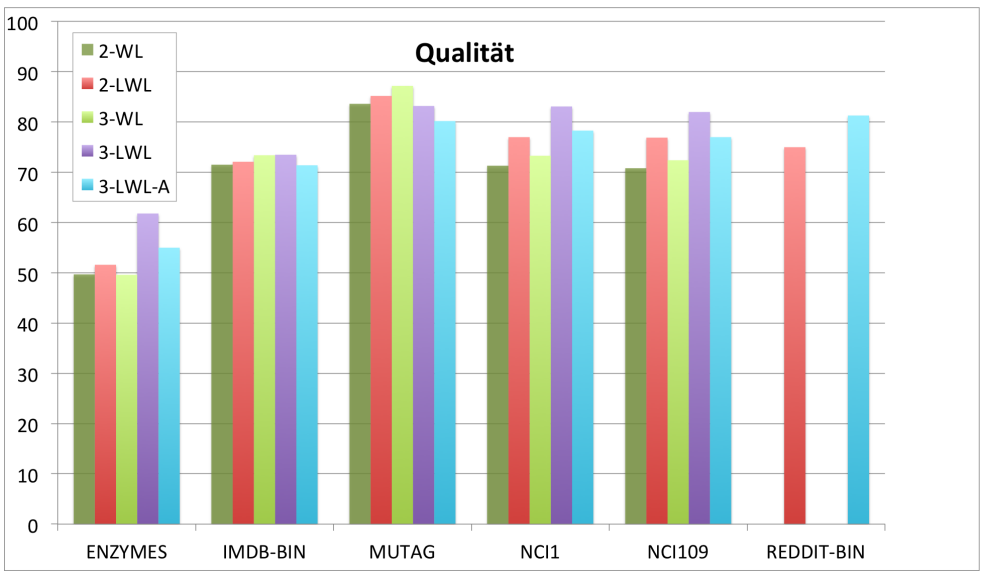


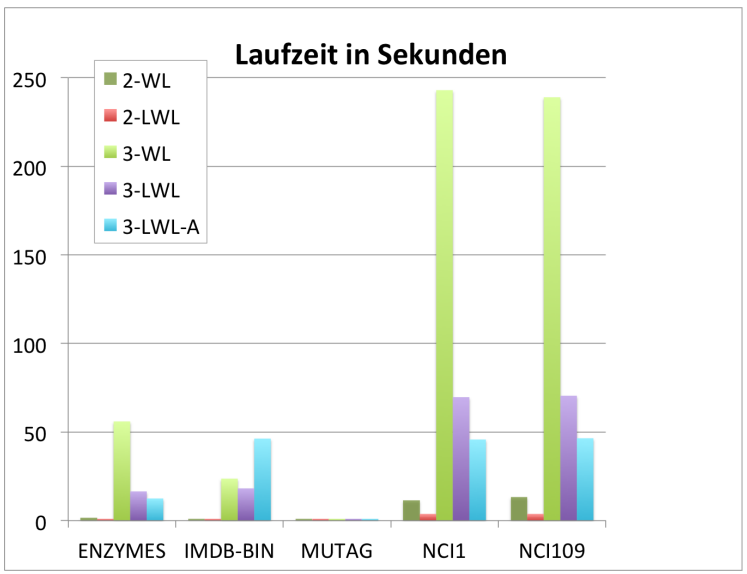
$\Phi(G) = (3, 1, 1)$ $\Phi(H) = (3, 1, 1)$ $\Phi(G) = (3, 1, 1, 2, 0, 0, 1, 1, 1, 0)$ $\Phi(H) = (3, 1, 1, 2, 1, 1, 0, 0, 0, 1)$

$\rightarrow$ Similarity measure based on graph kernel, Jaccard-Coefficient, ...

Evaluation of the k-LWL (graph kernel)

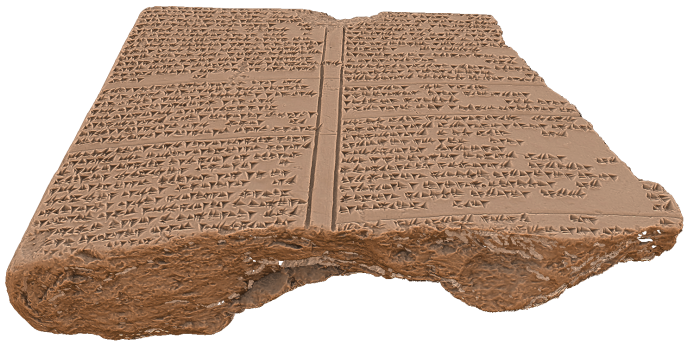
- Protein interaction networks (ENZYMES)
- Molecule data bases (MUTAG)
- Cancer data sets (NCI1, NCI109)
- Social networks (IMDB-BIN, REDDIT-BIN)

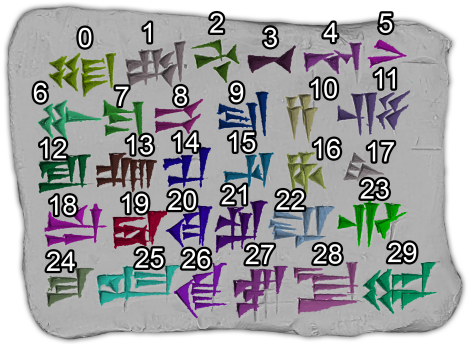


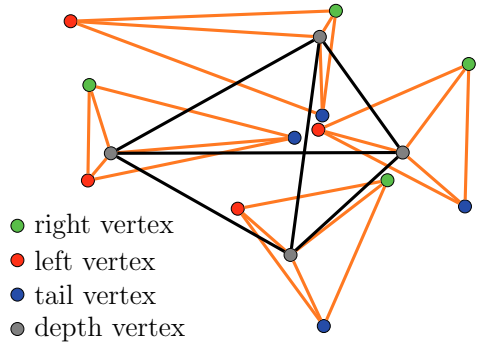


Morris, Kersting, Mutzel 2017

Recognition of Cuneiform Characters







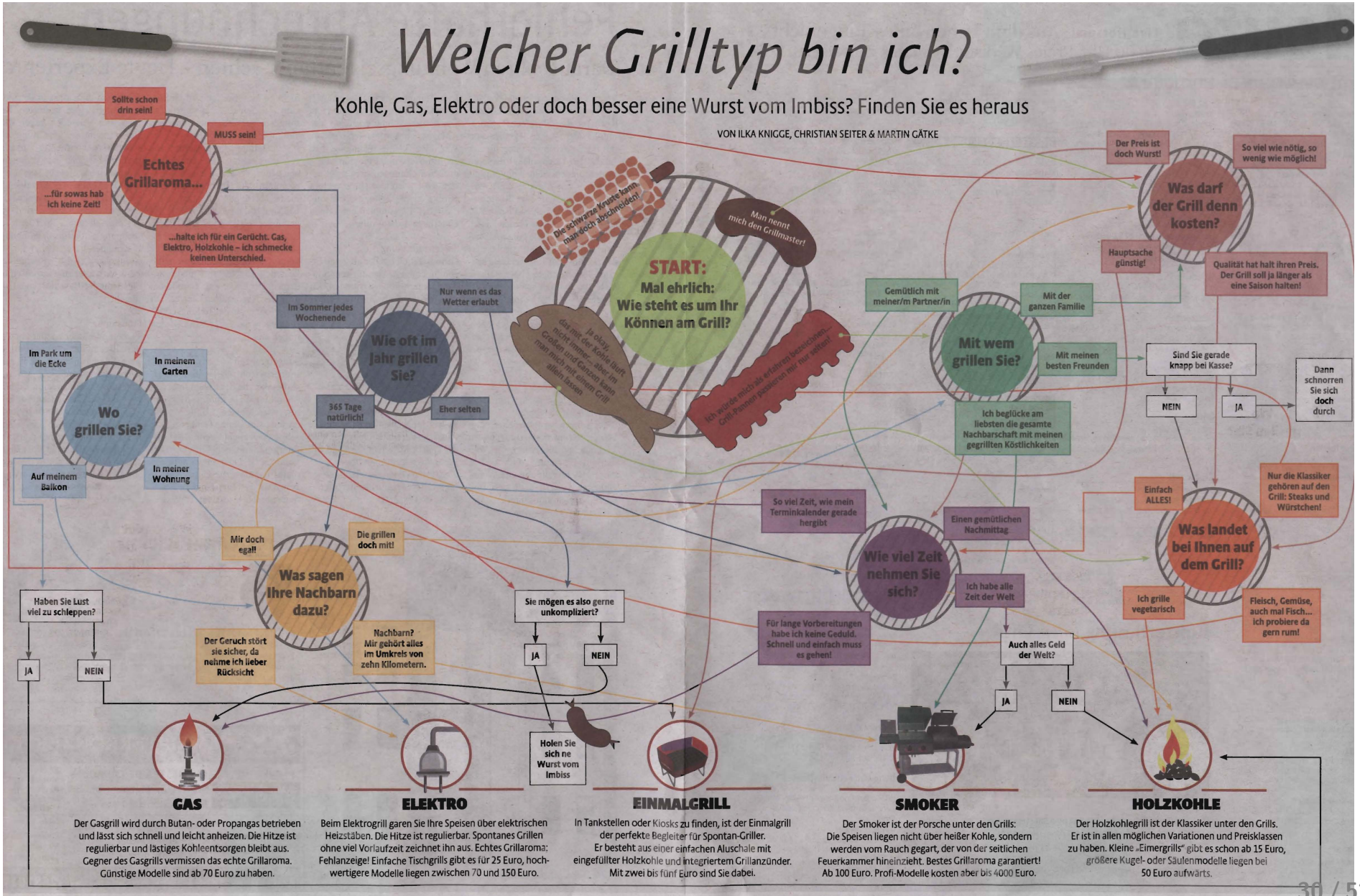
- 500.000 digitized cuneiform fragments (LS7, TU Dortmund)
- group of wedge signs corresponds to a character
- so far 1000 different characters known
- only a very small set of cuneiform characters have been classified
- Goal: Recognition of Cuneiform characters for supporting classic Altphilologists

mit Kriege, Fey, Fisseler, Weichert: COST, Proc. MLR 2018
DFG SFB 876: Providing Information by Resource-Constrained Data Analysis

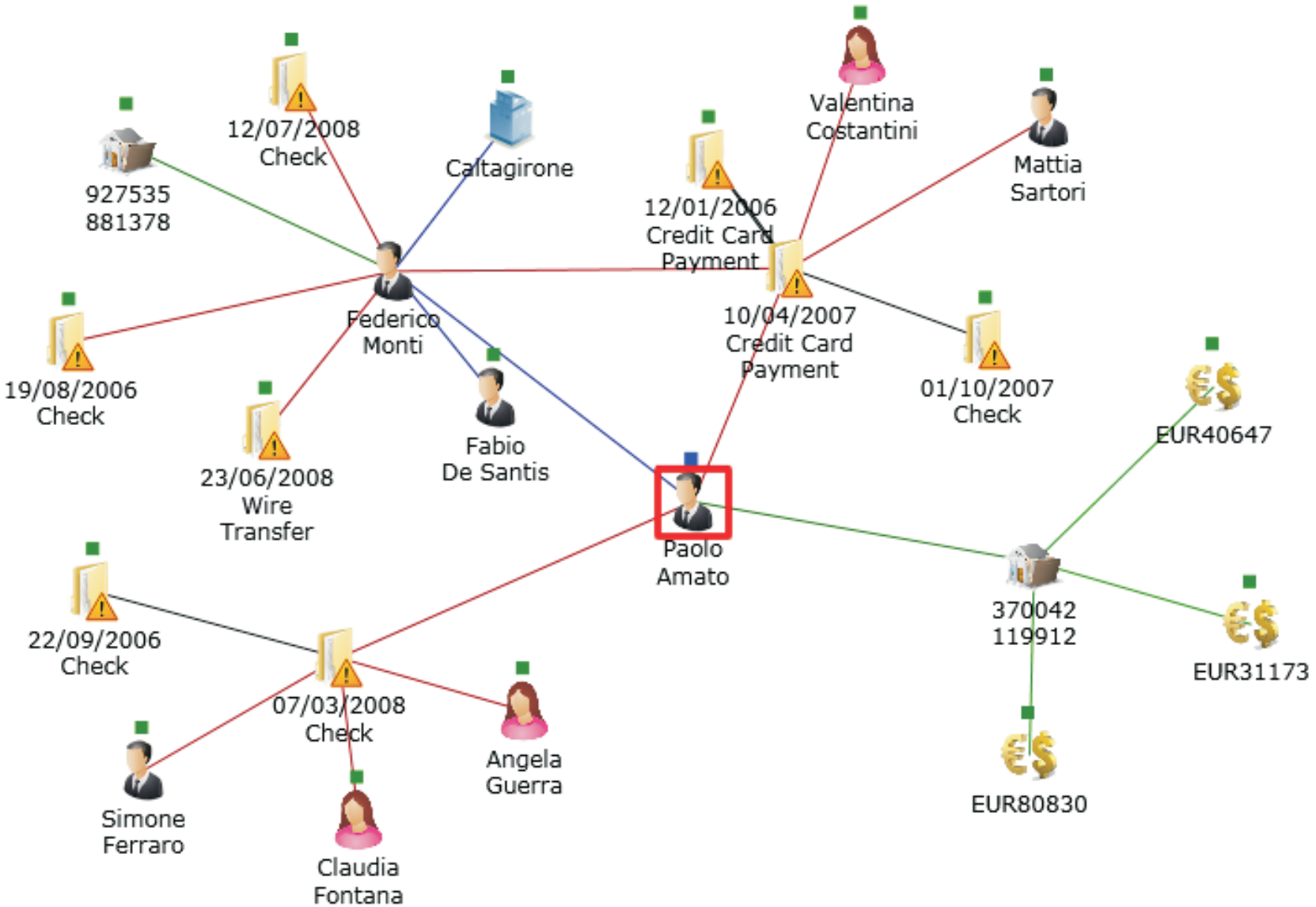
Explorative Data Analysis

Explorative Data Analysis

Which type of grill shall I buy?

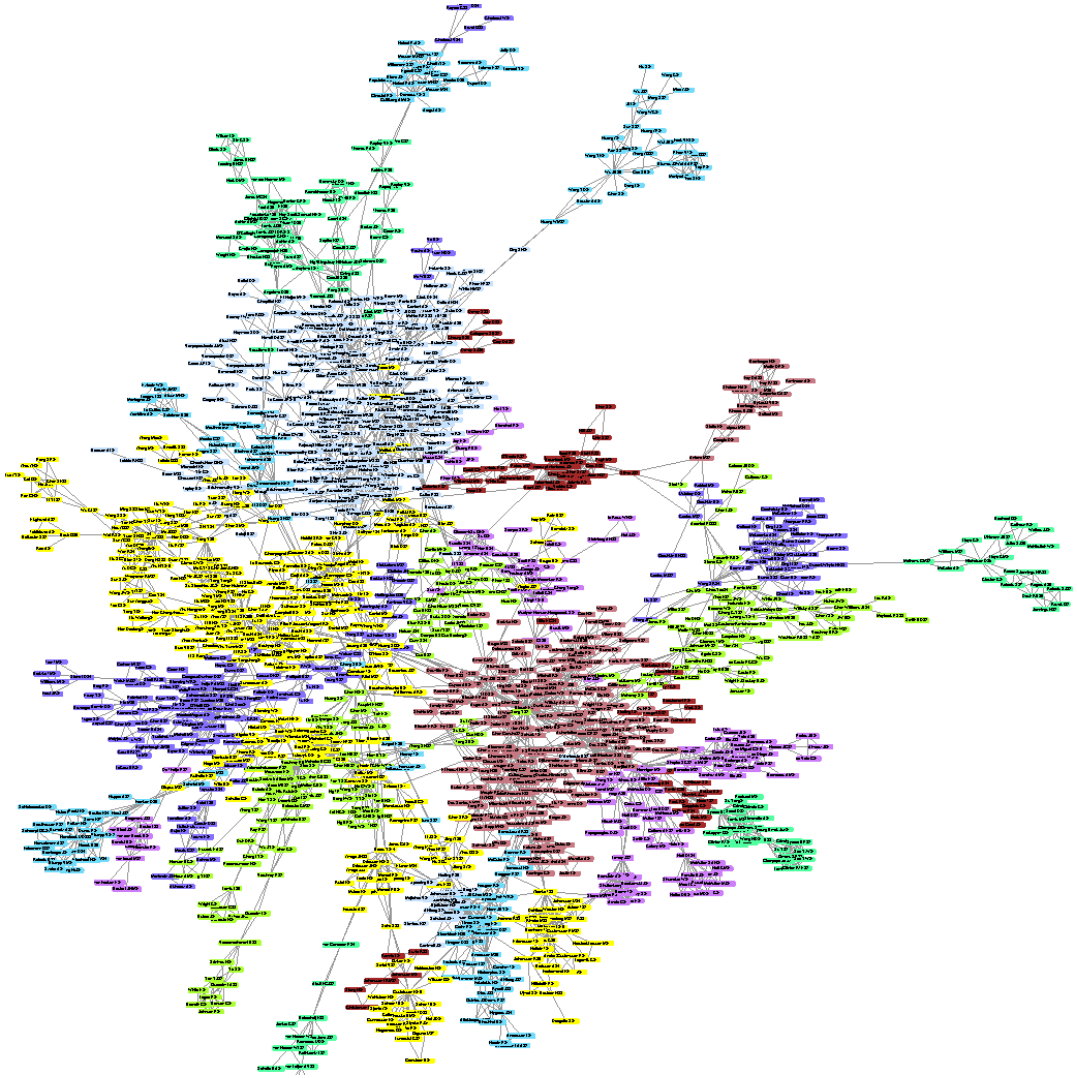


Social Network Analysis



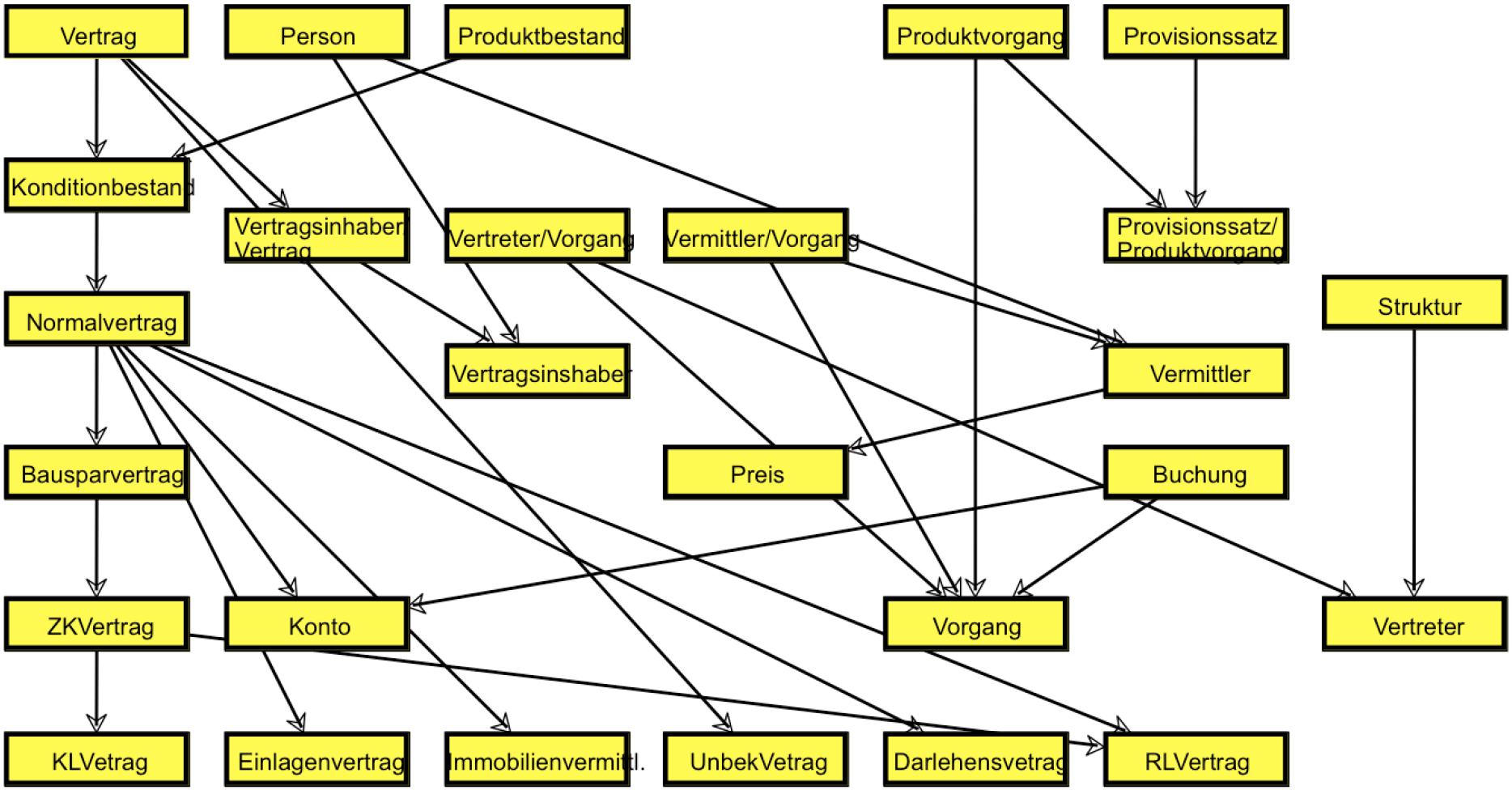
Source: Didimo, Liotta, Palladino, Montecchiani 2011

Co-author Network (Information FUSION)

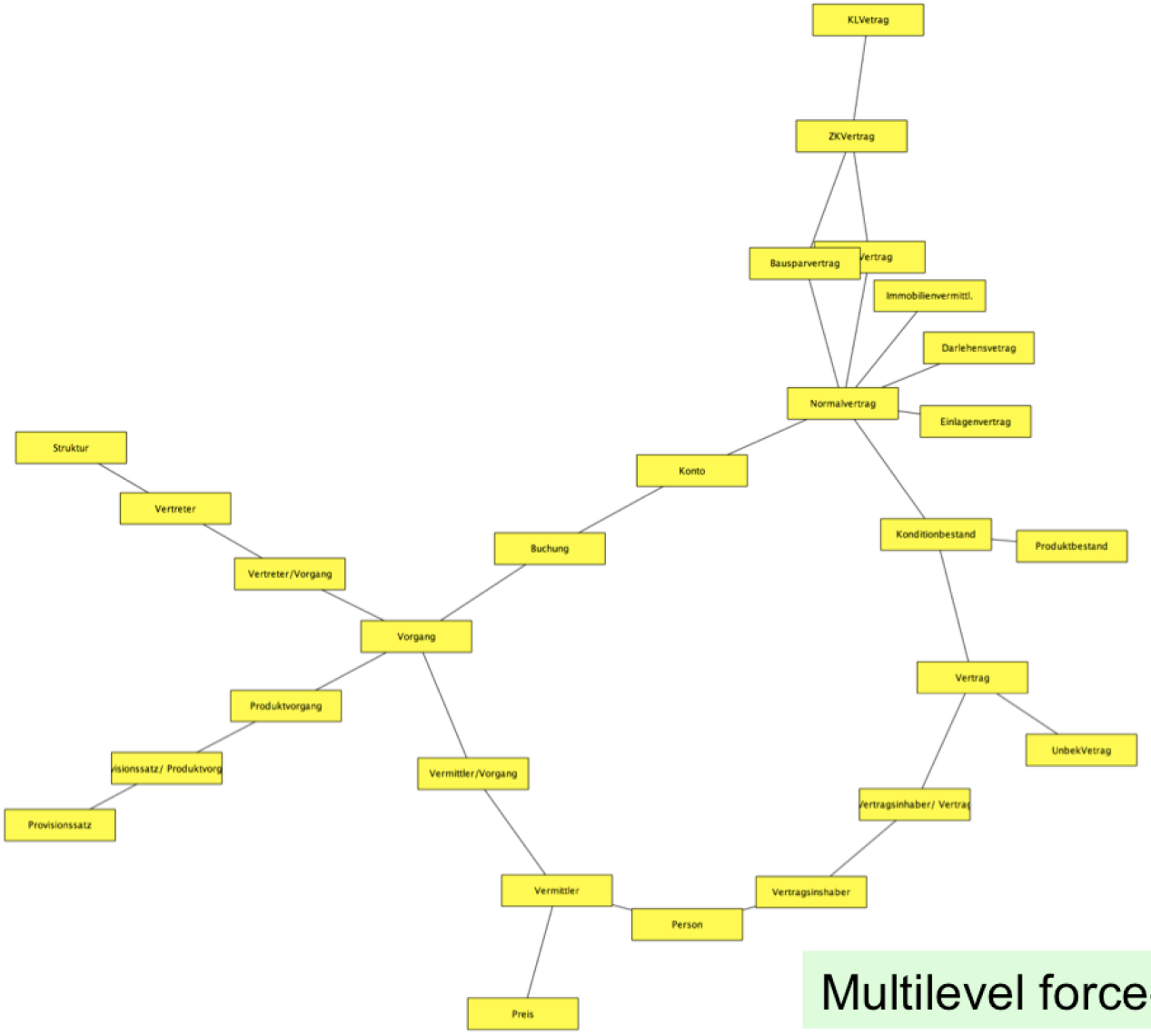


Source: Johansson, Martenson, Svenson 2011, layout by Pajek

Data Base Model (Original)

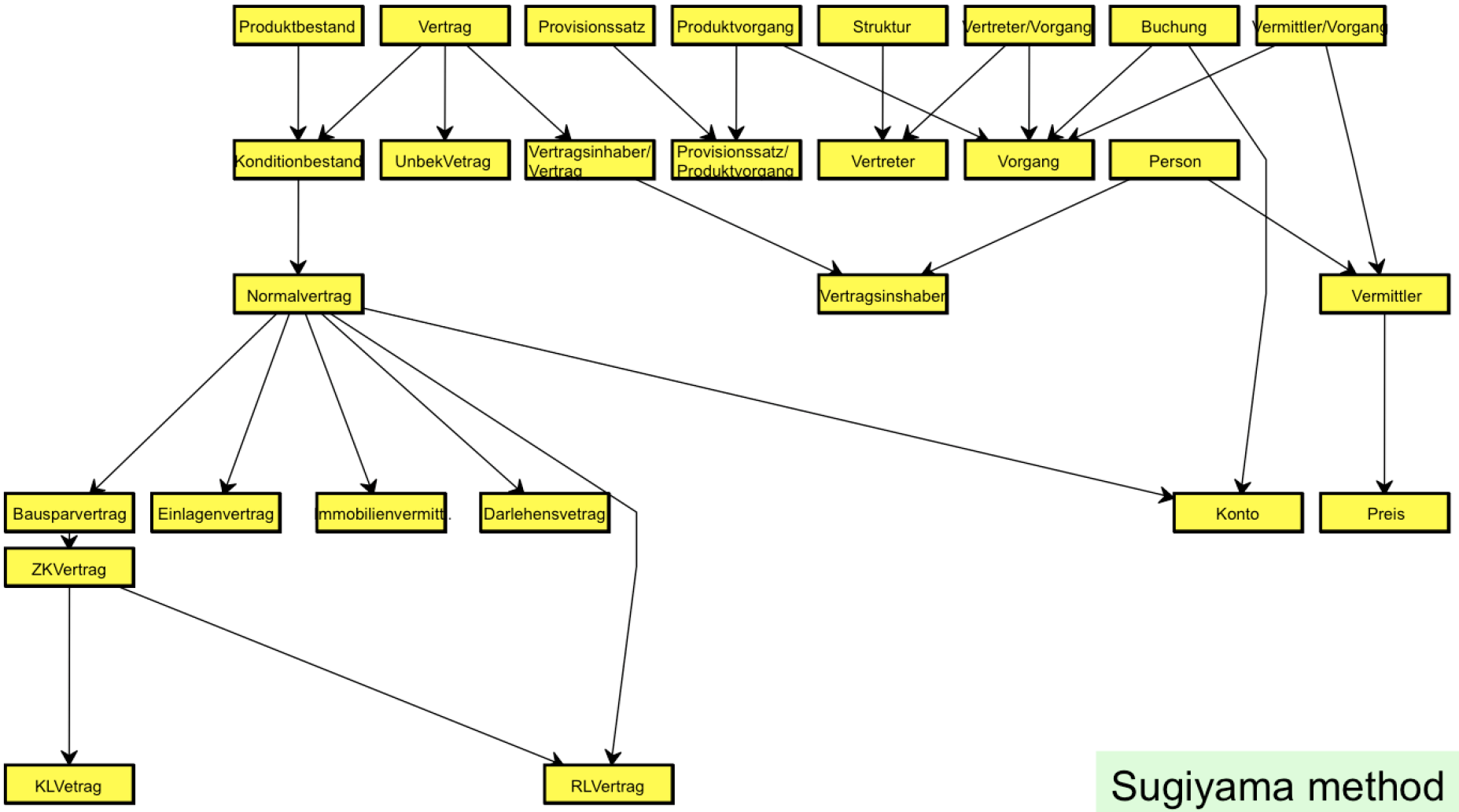


Data Base Model (Force-Directed)

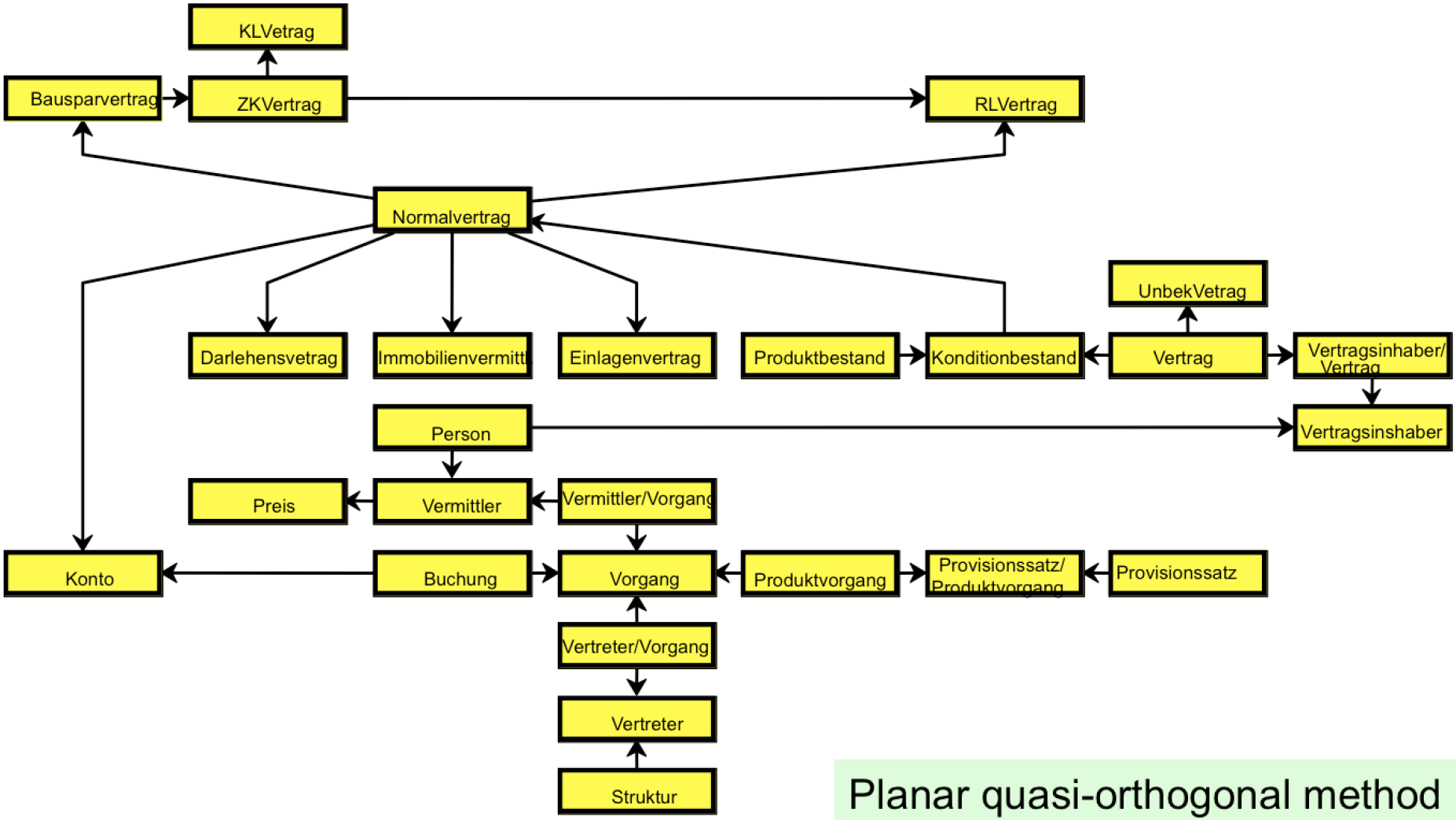


Multilevel force-directed FMMM

Data Base Model (Sugiyama)

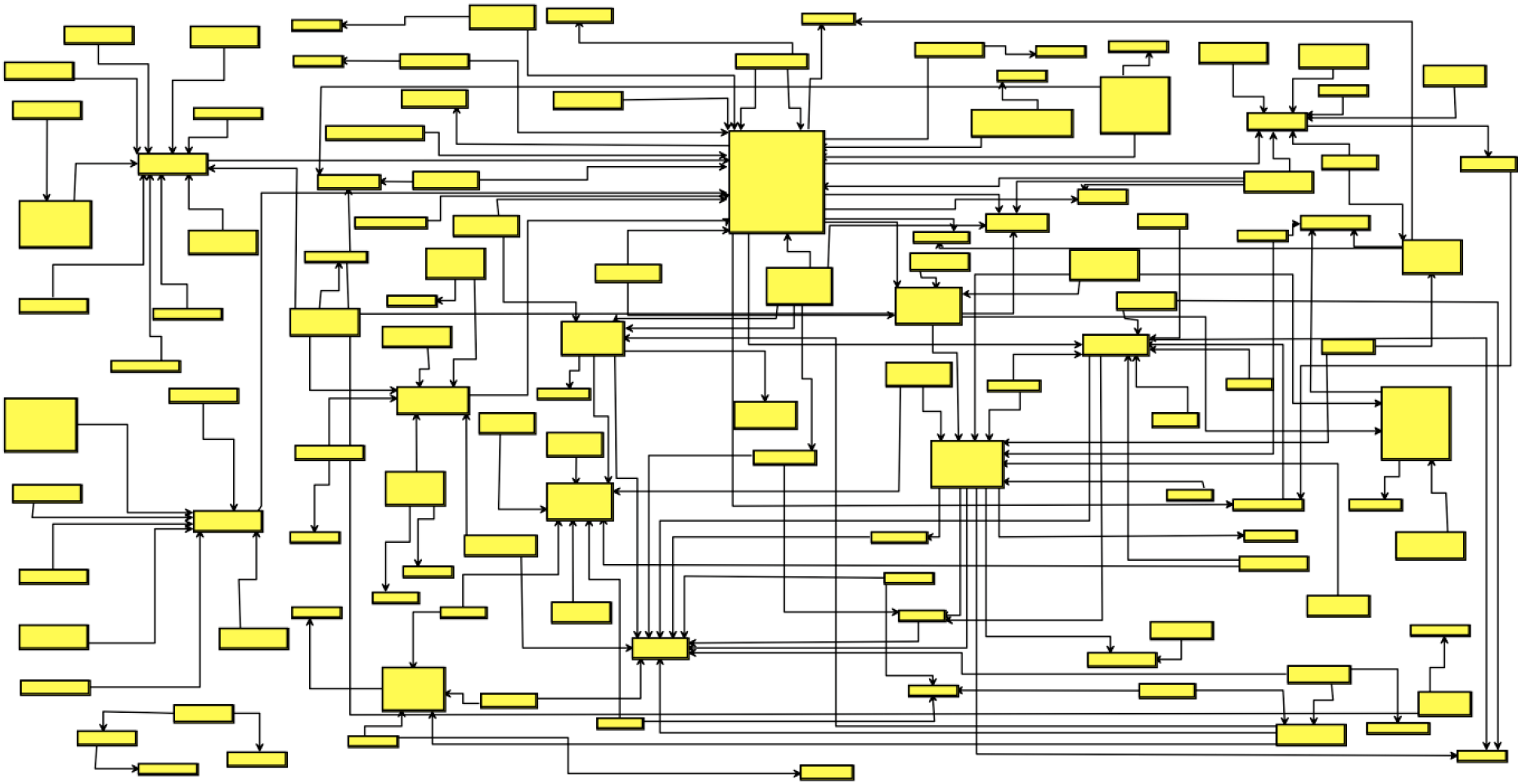


Data Base Model (Planar Quasi-Orthogonal)

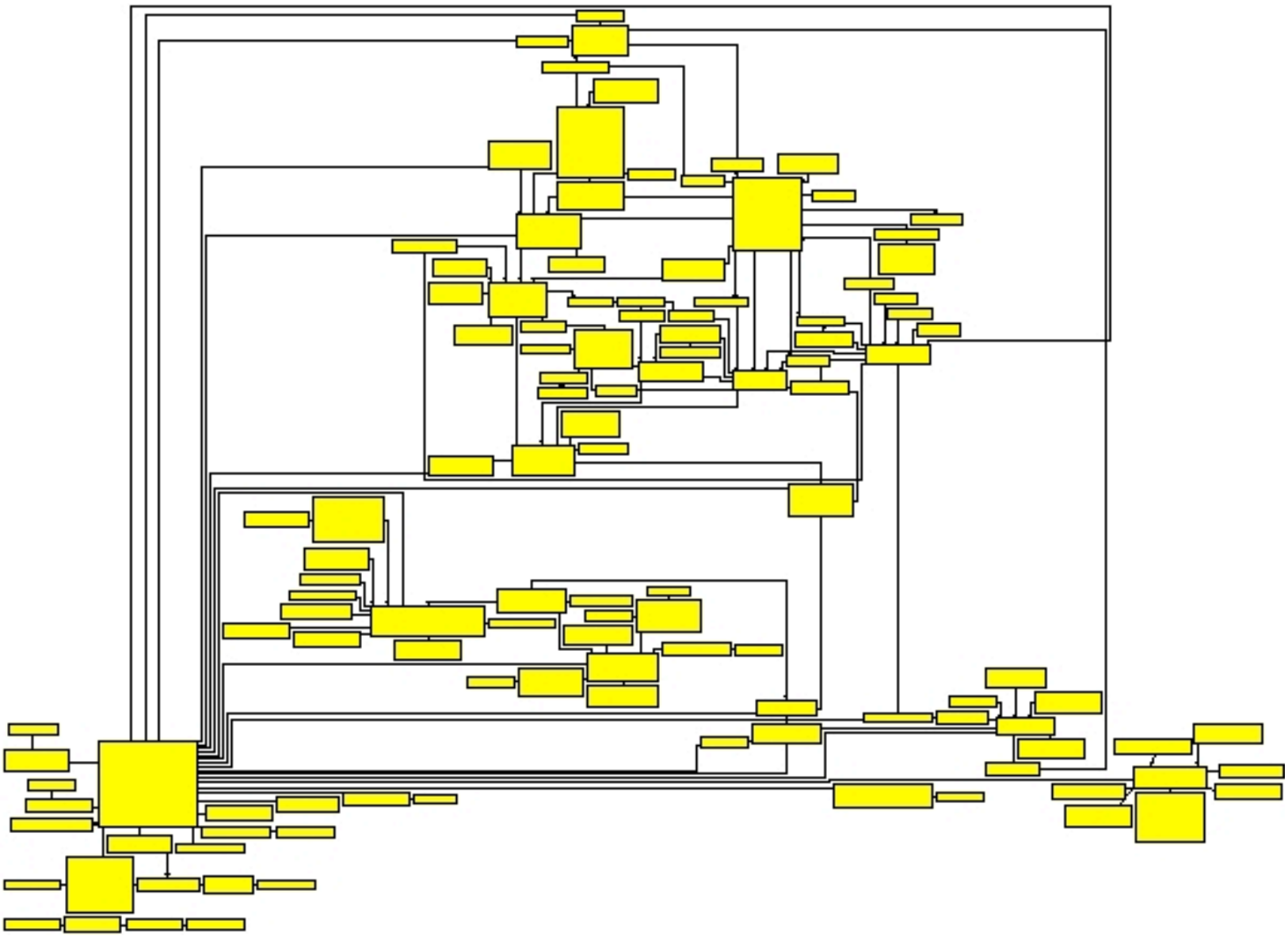


Planar quasi-orthogonal method

Data Base Model by Insurance Company (Original)



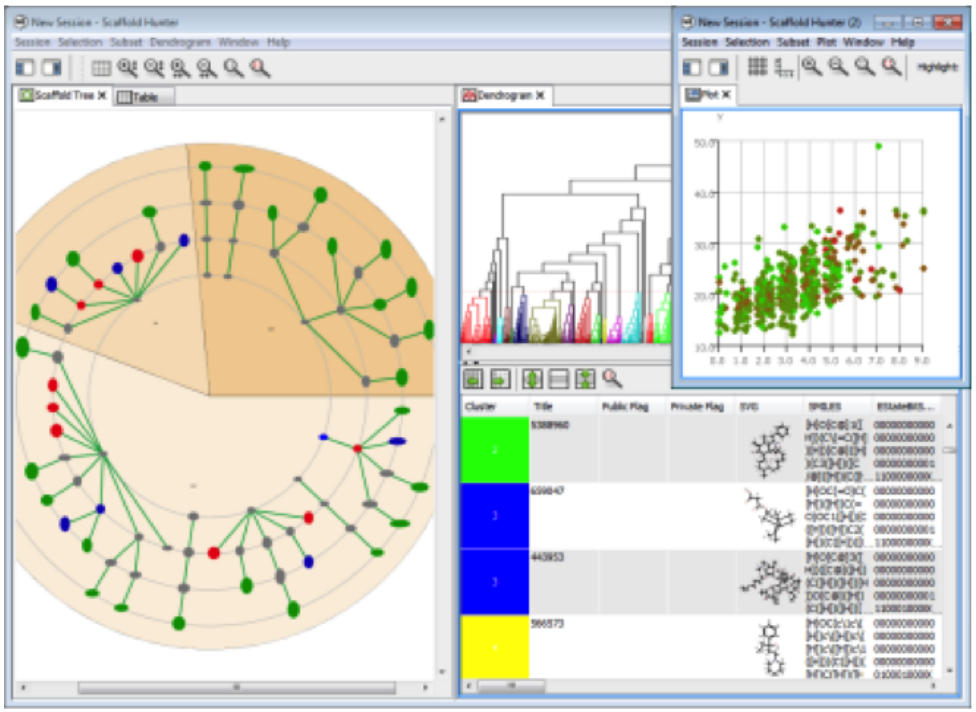
Data Base Model by Insurance Company (Layout)



Optimization Topics in Graph Drawing

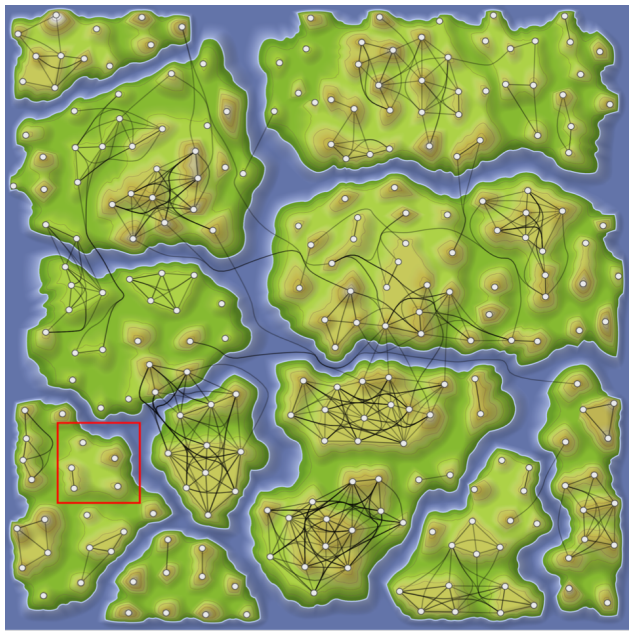
- Crossing minimization
- Layered crossing minimization
- (Planar) bend minimization
- Compaction for (planar) orthogonal drawings
- Vertex and edge labelling
- Embedding optimization
- ...

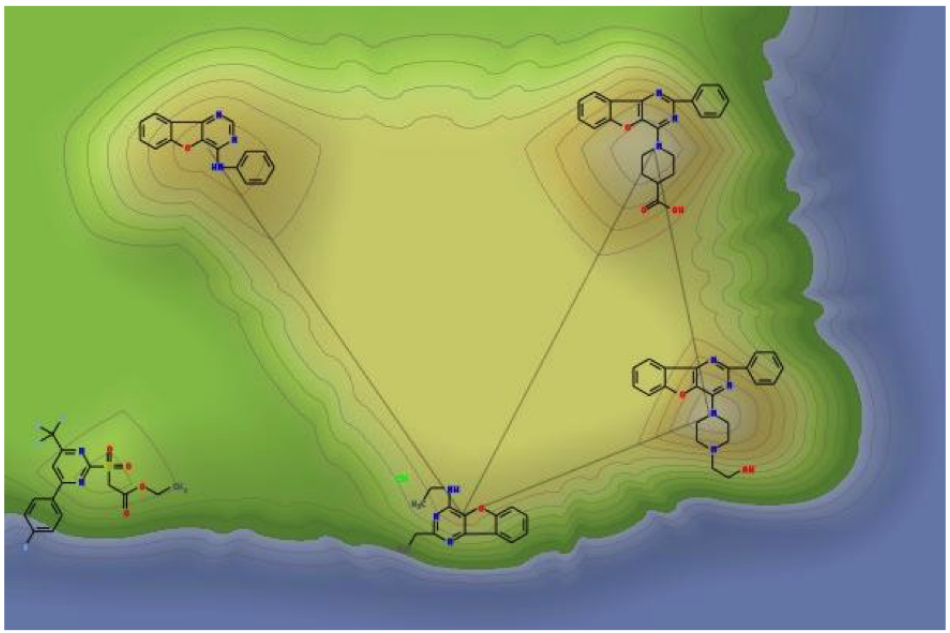
Explorative Data Analysis — Impressions



Scaffold Hunter: Visualization of chemical molecule data bases

Map based Layout of Molecule Data

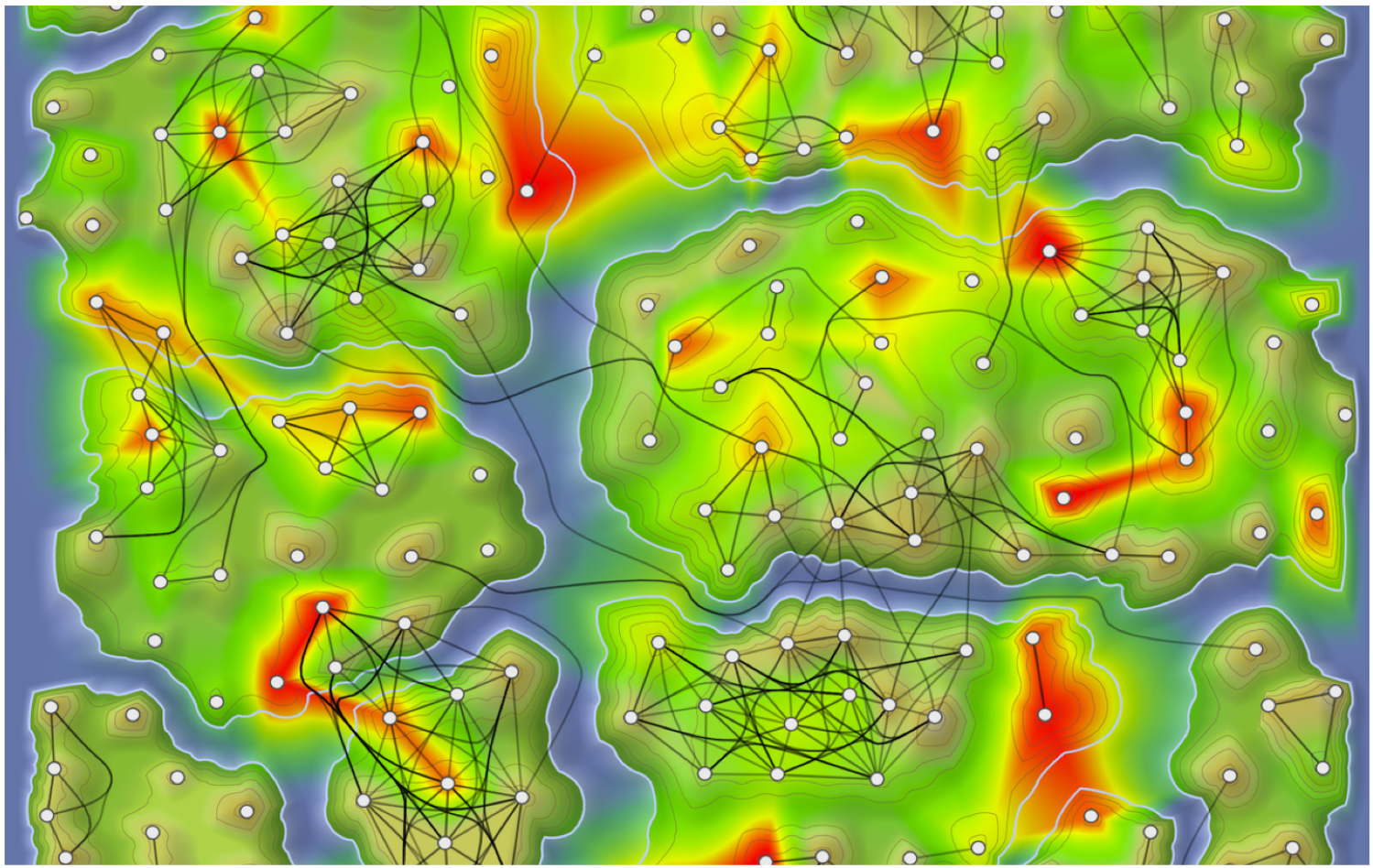




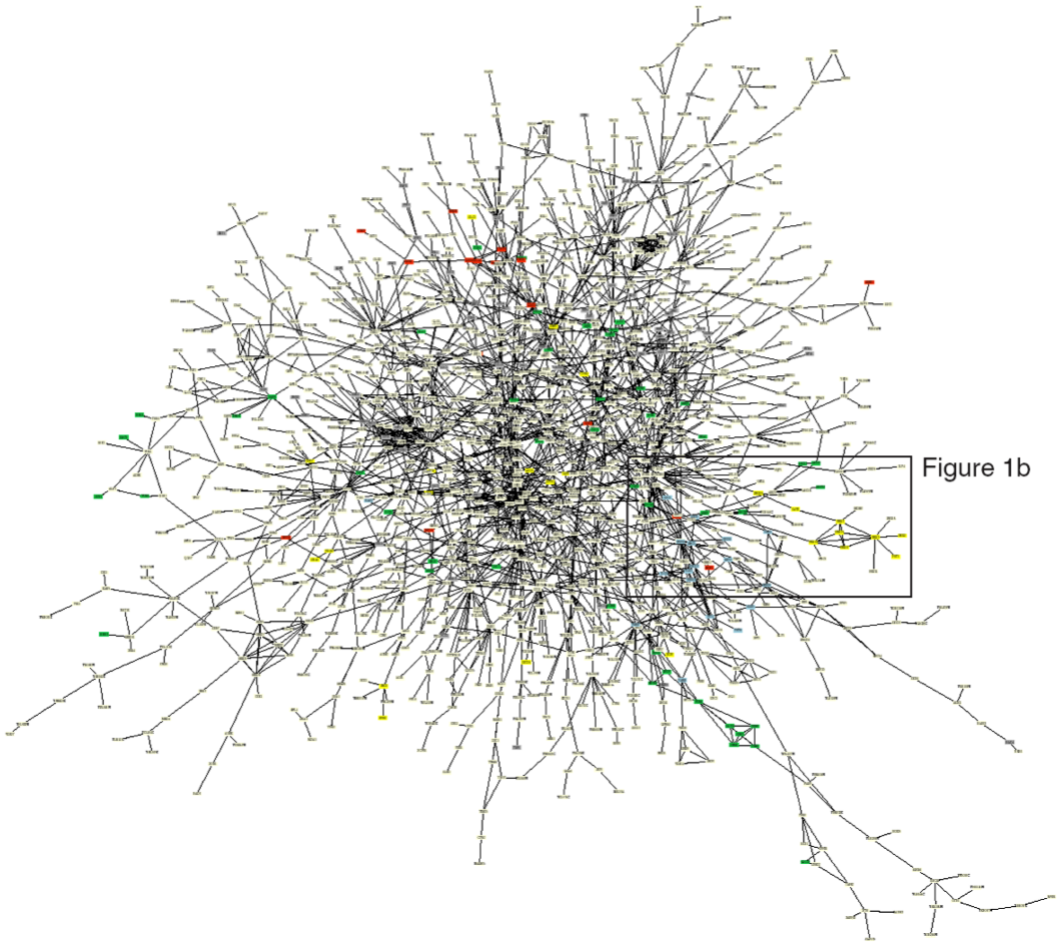
- Irwin 2009: "In principle, one would like to be able to organize and browse large chemical datasets [...] as easily as one can today browse maps on the internet." [Nat. Chem. Biol.]
- Cluster hierarchy based on structural similarity (height level)

Gronemann, Jünger, Kriege, Mutzel 2013

Map based Layout of Molecule Data

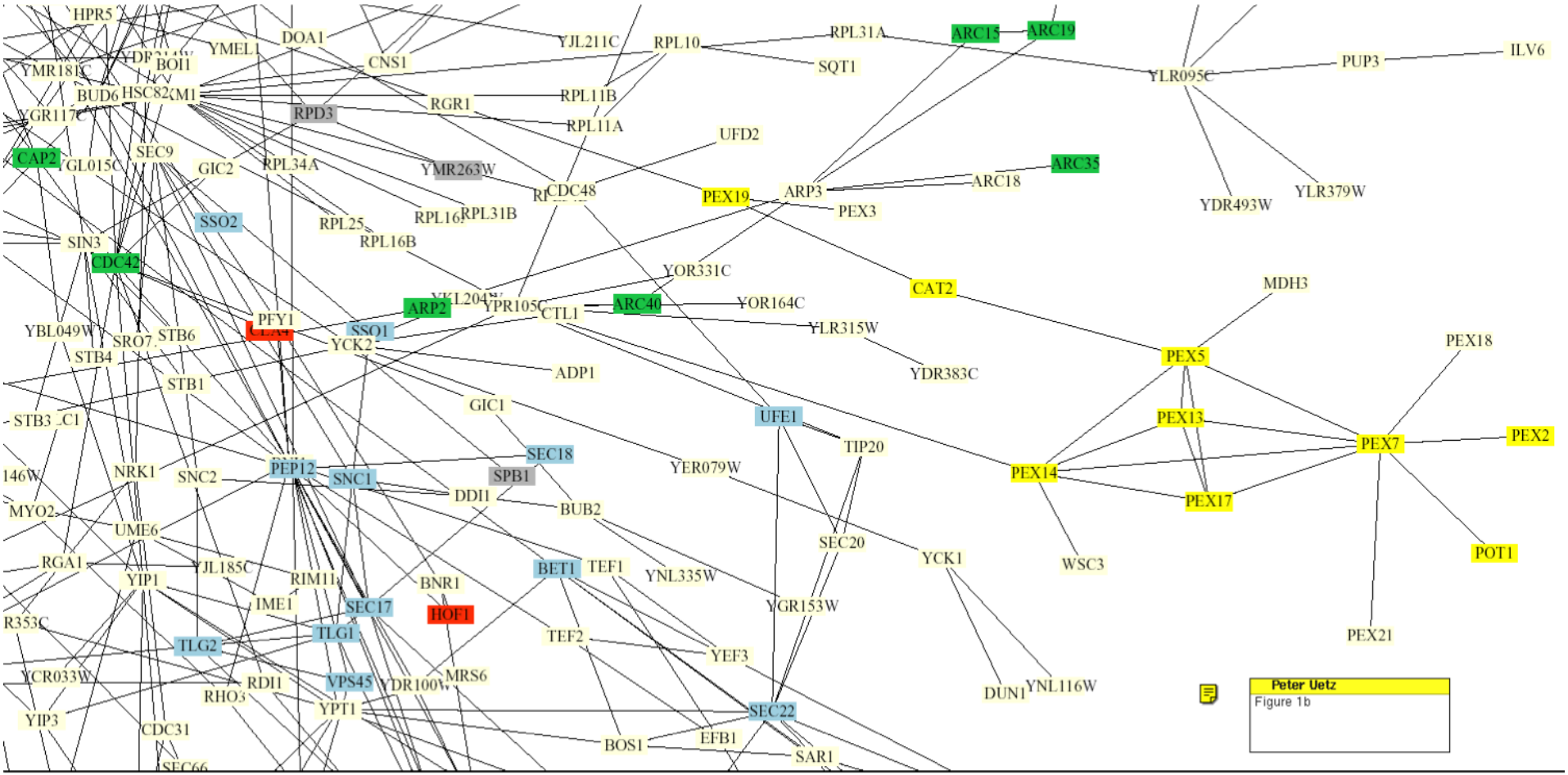


Protein Interaction Network in Yeast

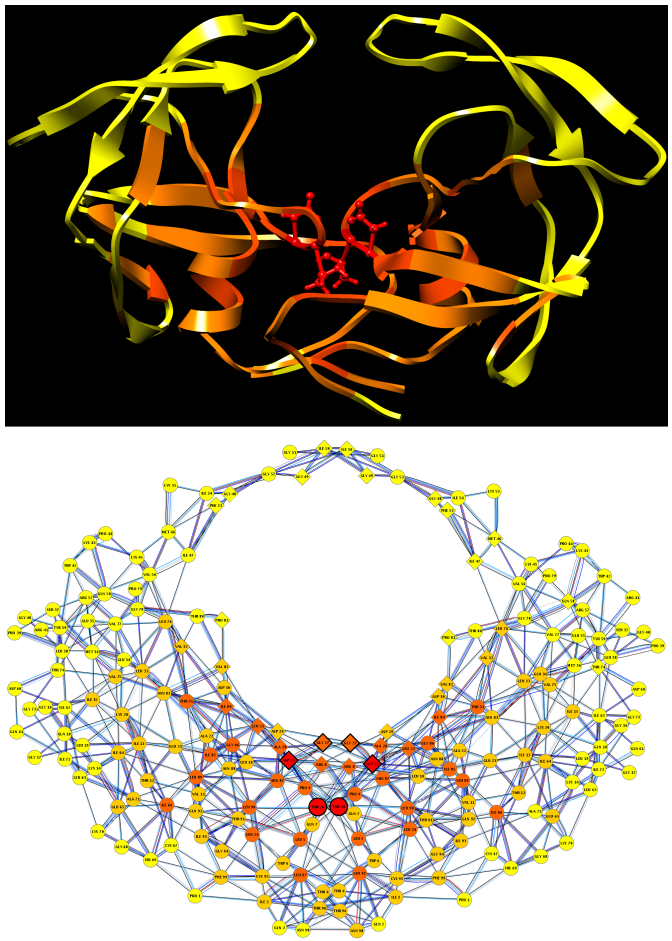


- Schwikowski, Uetz, Fields [Nature Biotechnology 2000]
- drawn with our graph drawing software AGD (MPI Informatik)
- 1548 proteins with 2358 interactions

Protein Interaction Network in Yeast

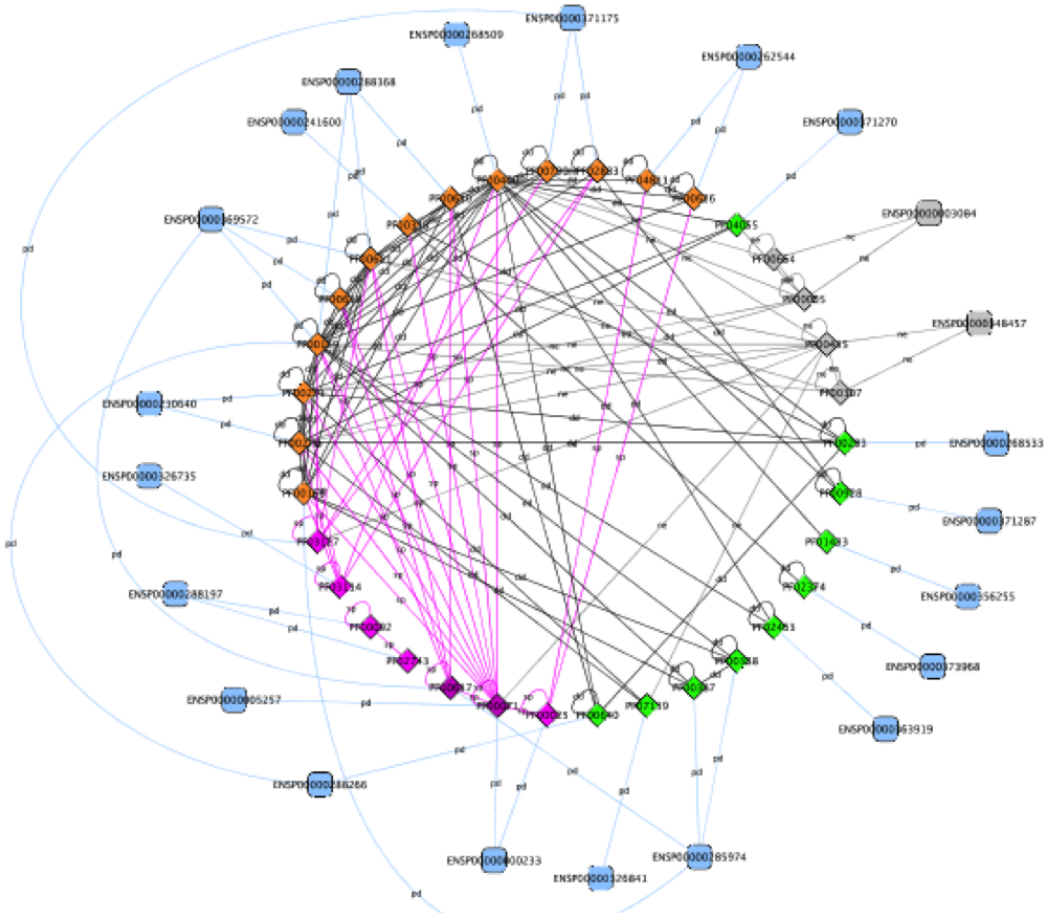


Protease with Centrality Analysis



Cooperation with AG Albrecht (MPI Informatik, TU Graz), Cytoscape 2011

Protein-Domain Interaction Network

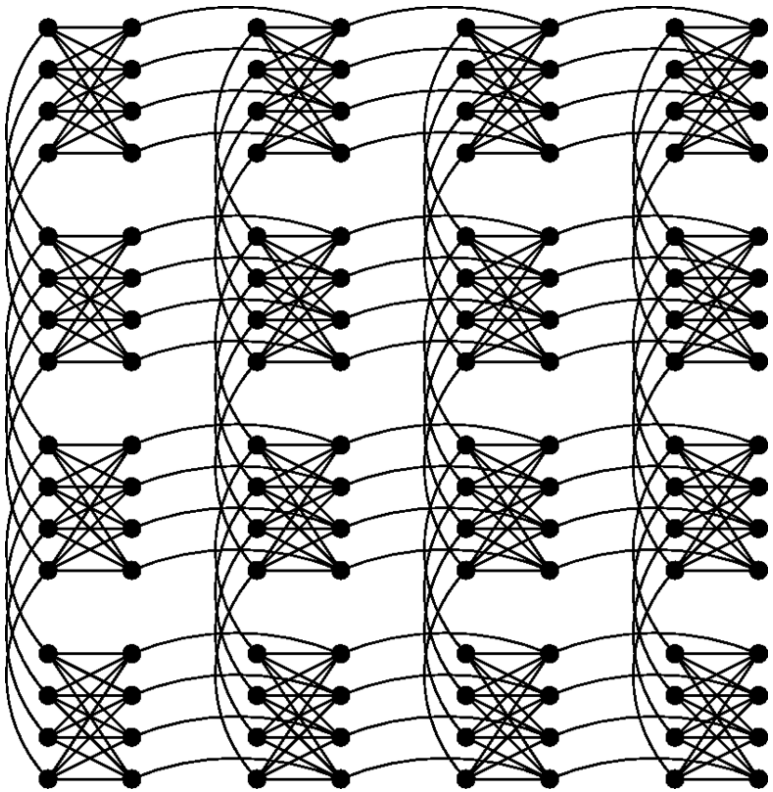


Cooperation with AG Albrecht (MPI Informatik, TU Graz), Cytoscape 2011

Adiabatic Quantum Computing

Structure of Adiabatic Quantum Computer by D-Wave:

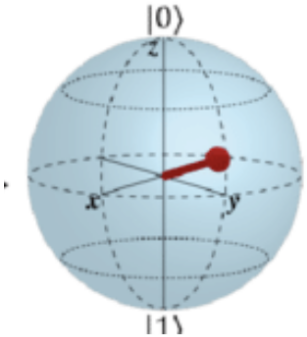
4x4 Chimera grid

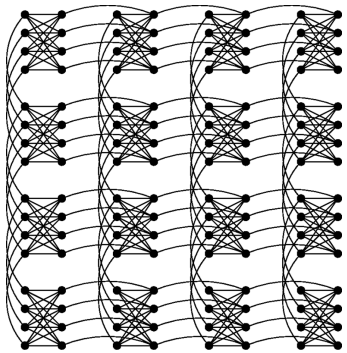


D-Wave 2017: 2000Q with 2048 Qubits (16x16 Chimera grid)

mit Jünger (Köln), Reinelt (Heidelberg), Rinaldi (Rom); Stollenwerk, Lobe (DLR), Kaibel (Magdeburg); McGeoch (D-Wave)

Adiabatic Quantum Computing





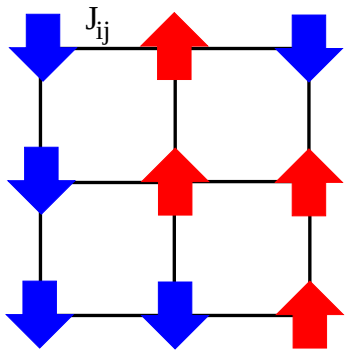
Idee: Adiabatic Quantum Computing

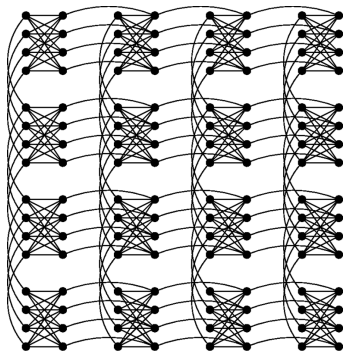
- construct system (A) with (unknown) ground state, which corresponds to the solution of my problem
- construct system (B) whose ground state can be prepared and measured experimentally
- transfer system (B) slowly to system (A) and measure the ground state ← Adiabatic Theorem

D-Wave: 2017: 2000Q with 2048 Qubits (16x16 Chimera grid)

Quelle:
https://www.researchgate.net/publication321133310_Molecular_Spin_Qudits_for_Quantum_Algorithms/figures?lo=1

Quantum Annealing





Definition (Ising Spin Model Problem (ISM))

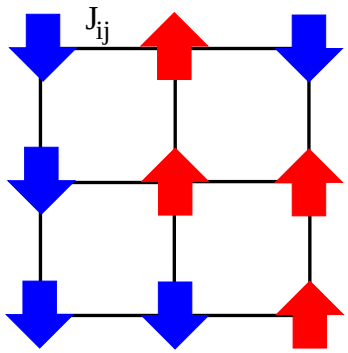
- **Given:** $G = (V, E)$ with weights h_i $(i \in V)$ and J_{ij} $((i,j) \in E)$
- **Find:** Values for the spin variables $s_i \in \{-1, +1\}$, $i \in V$, so that

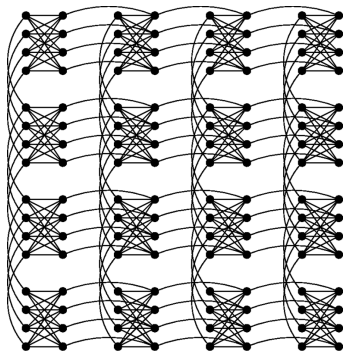
$$E(s) = \sum_{i \in V} h_i s_i + \sum_{(i,j) \in E} J_{ij} s_i s_j$$

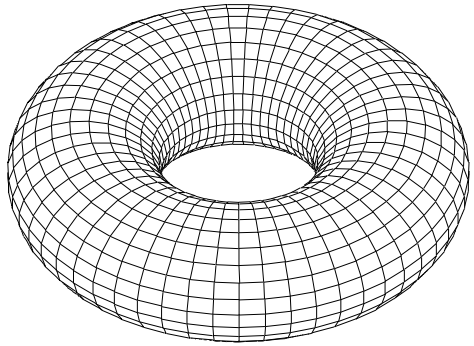
is minimized. $\leftarrow$ ground state

(ISM) equivalent to QUBO, MAX W2SAT, MAX CUT

Quantum Annealing







Definition (Ising Spin Model Problem (ISM))

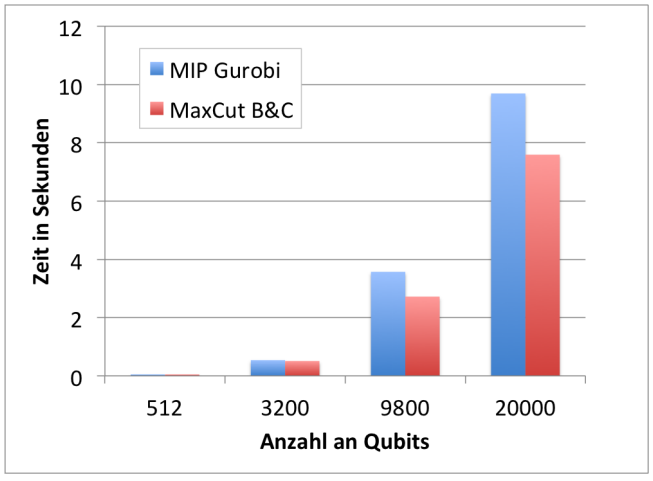
- **Given:** $G = (V, E)$ with weights h_i $(i \in V)$ and J_{ij} $((i,j) \in E)$
- **Find:** Values for the spin variables $s_i \in \{-1, +1\}$, $i \in V$, so that

$$E(s) = \sum_{i \in V} h_i s_i + \sum_{(i,j) \in E} J_{ij} s_i s_j$$

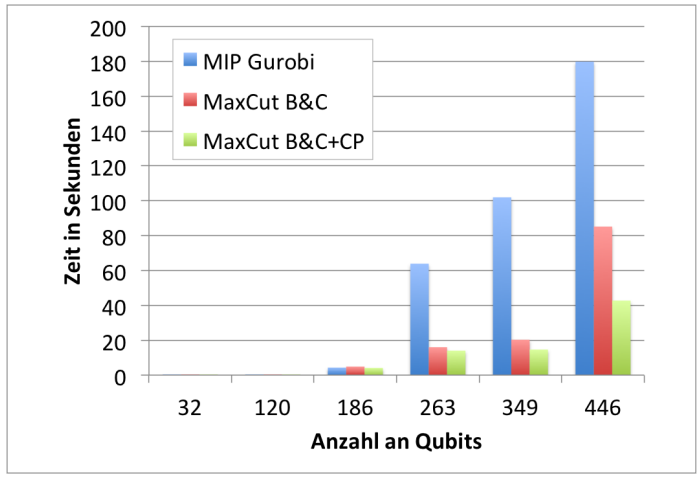
is minimized. $\leftarrow$ ground state

(ISM) equivalent to QUBO, MAX W2SAT, MAX CUT

Experimental Evaluation (exact solution)



random QUBO instances

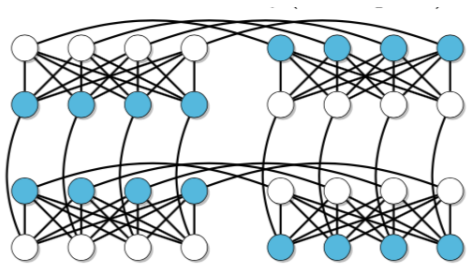


random ISM instances

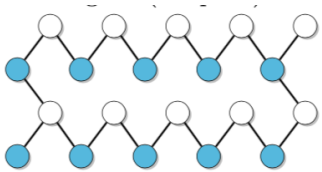
- D-Wave Quantum Annealer (C_8 with 500 Qubits) vs. QP CPLEX [McGeoch,Wang 2013]
- Experiments with MIP-solver CPLEX [Dash/Puget 2015]
- Our MaxCut Branch-and-Cut vs. strengthened MaxCut B&C by cutting planes $\leftarrow$ provable exact solutions

mit Jünger (Köln), Reinelt (Heidelberg), Rinaldi (Rom)

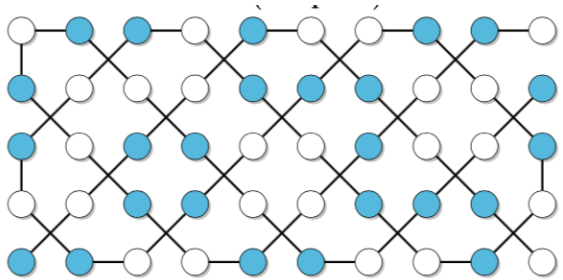
Future Research



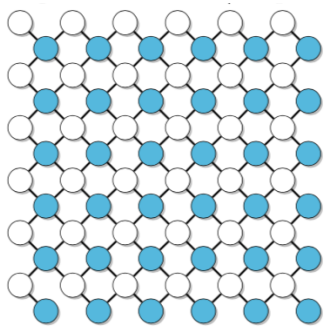
DWave 2048 Qubit



Rigetti 20 Qubit



IBM 50 Qubit



Google Bristlecone 72 Qubit

- Quality comparisons with D-Wave (via DLR to D-Wave at NASA)
- Practical problems: MAX SAT for general graphs; must be embedded into Chimera grids ← graph-minor embedding problems
- similar for true Quantum computers by Rigetti, IBM, Google
- Fixed-Parameter-Tractable Algorithms for Qubit instances

mit Jünger (Köln), Reinelt (Heidelberg), Rinaldi (Rom); Stollenwerk, Lobe (DLR), Kaibel (Magdeburg); McGeoch (D-Wave)
Image sources: Goodrich, Horton, Sullivan 2018

Plan (possible topics) for this Lecture Series

- Data streaming algorithms, e.g., for the matching problem in graphs
- Temporal graphs, e.g., for quickest path problems or multicriteria path problems
- Dynamic graph algorithms, e.g., shortest path problems
- Weisfeiler-Lehman algorithms for graph classification
- Graph similarity: maximum common subgraphs
- Parallel/distributed graph algorithms
- Graph clustering (clustering of a set of graphs and clustering of vertices of a graph)
- Graph and network visualization
- Crossing minimization in graphs
- Fixed-parameter algorithms, e.g., for max-cut on 1-planar graphs
- External memory algorithms and data structures
- Anything else you are interested in: max-cut problem for quantum computing, ILP-models for graph coloring, network design problems