Seminar of Machine Learning and Modeling
Prague, 11 October 2012

Stochastic Graph Algorithms: Clique Covering and Clustering

David Chalupa
Slovak University of Technology
Institute of Applied Informatics
Faculty of Informatics and Information Technologies

STU
FIIT

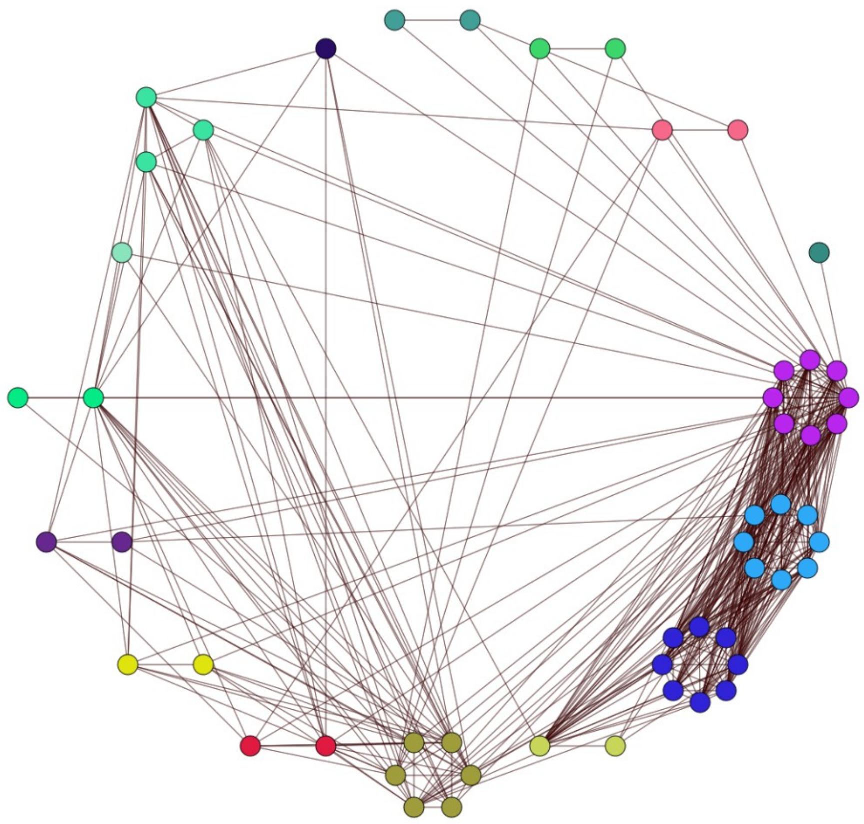
Outline of the Talk

- problems: theory and applications
- concepts of solving for the studied problems
  - algorithmic strategies for the clique covering problem (CCP) and graph clustering
  - analytical vs. experimental methodology of evaluation
- current results
  - an order-based representation for CCP and order-based algorithms: IG and RLS
  - multicriteria construction procedures (MCPs) for graph clustering
- conclusions, discussion, references

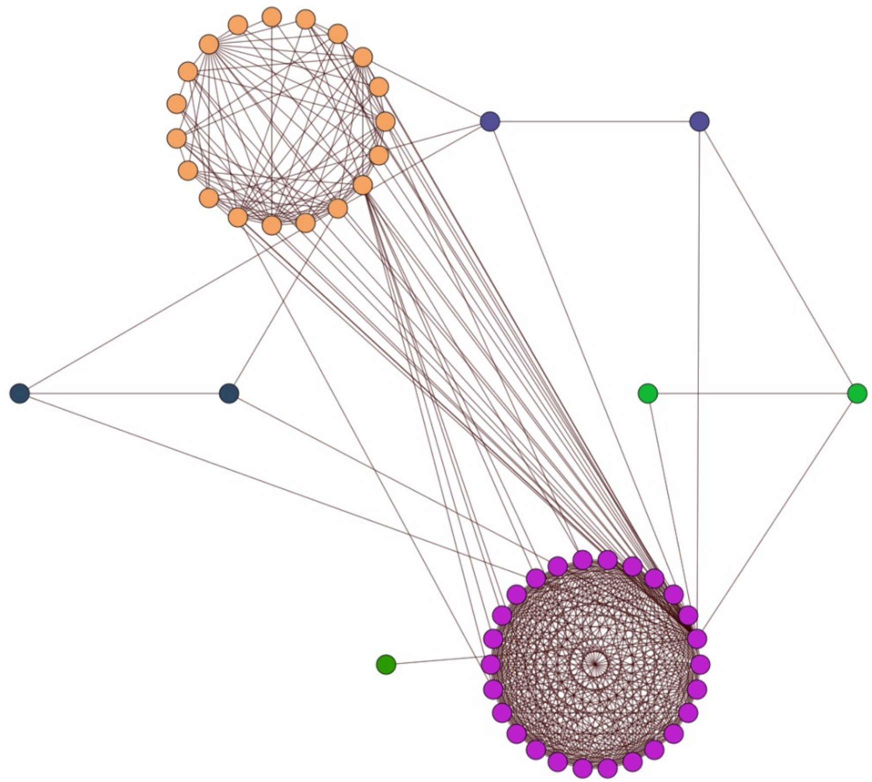
Clique Covering and Graph Clustering Problems

Problems: Clique Covering and Graph Clustering

- visual illustration on a small social network



clique covering



graph clustering

Motivation

- computational hardness
  - *clique covering* is NP-hard [Karp, 1972]
  - *graph clustering* is difficult even to define, many meaningful quality measures are NP-complete [Schaeffer, 2007]
- *real-world applications* of this type of problems
  - data mining [Sun et al., 2008] and web mining [Tang et al., 2011]
  - social network analysis [Chalupa, 2011a], social media marketing [Schaeffer, 2007]

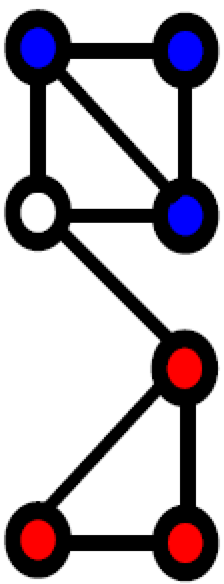
Motivation

- research citation network analysis [Sun et al., 2008]
- protein interaction in bioinformatics [Gao et al., 2009]
- gene-activation dependencies in bioinformatics [Boyer et al. 2005]
- analysis of terrorist organization networks [Patillo et al., 2012]
- infectious diseases epidemiology [Rothenberg et al., 1996]
- scheduling and timetabling [Burke et al., 2007]
- frequency assignment in mobile radio networks [Smith et al., 1998]
- and even more...

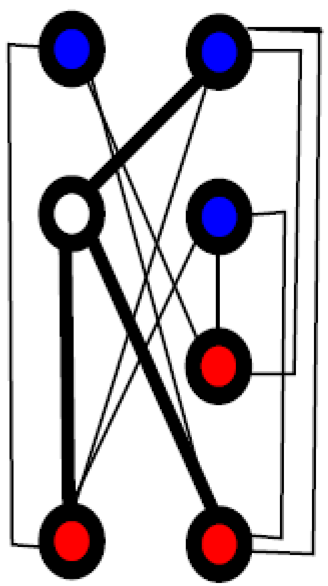
Clique Covering and Graph Coloring

- (vertex) clique covering problem (CCP)
  - „*inverse graph coloring*“
  - reduction from one problem to another [Karp, 1972]: let H = G‘ (complementary graph); then coloring of G‘ corresponds to clique covering of H and vice versa
  - *clique covering number*: $\vartheta(G)$, *chromatic number*: $\chi(G)$, $\vartheta(G) = \chi(G')$
  - coloring is *inapproximable* within $O(|V|^{1-\varepsilon})$ for any $\varepsilon > 0$ unless P = NP [Zuckerman, 2007]; the same holds probably also for the CCP
  - however, the problems are still *not the same*

Relationship Between Clique Covering and Coloring Problems



G

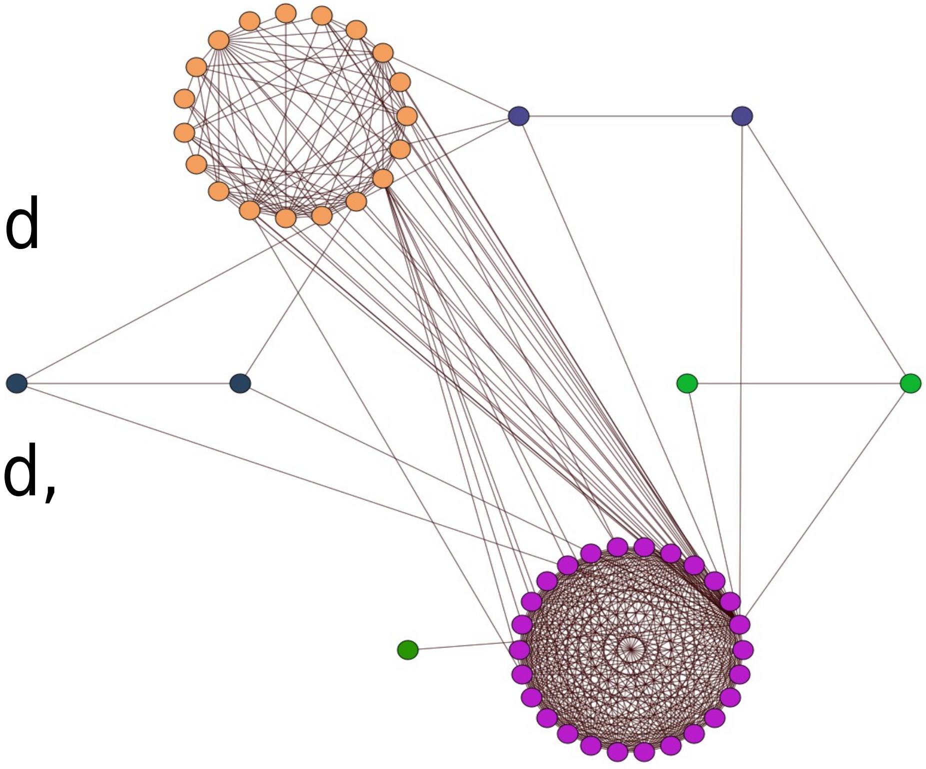


$\overline{G}$

- $\overline{G}$ – graph coloring
  - to choose a color, we have to scan the neighbors
  - we simply use a graph coloring heuristic on G
- G – clique covering
  - to choose a color, it is not enough to scan neighbors (without an additional information)

Graph Clustering

- a set of related decomposition problems
  - the aim is to decompose the graph into groups of "*similar*" vertices
  - "similarity" can be measured using *density*, *connectivity*, *centrality*, *distribution*, etc.
  - it is still not generally agreed, what is a *"good clustering"* [Schaeffer, 2007]



Concepts of Solving for Clique Covering and Graph Clustering

Concepts of Solving for Clique Covering and Graph Clustering

- clique covering (CCP)
  - *classical coloring heuristics* ([Brélaz, 1979]) - fast, quality strongly depends on the structure of the graph
  - *k-fixed local search and evolutionary algorithms* ([Galinier and Hao, 1999], [Titiloye and Crispin, 2011]) – solid quality of results, slow convergence, very inefficient if *k* is highly overestimated
  - *non-k-fixed stochastic algorithms* are less common ([Culberson and Luo, 1996])

Concepts of Solving for Clique Covering and Graph Clustering

- graph clustering
  - *hierarchical methods* ([Girvan and Newman, 2002]) – dendrogram-based, a popular metric is a betweenness of an edge
  - *centrality-based methods* ([Kaufman and Rouseeuw, 1990]) – typically *k-medoids*, using vertices as central points and optimizing their choice
  - *local search and evolutionary algorithms* ([Schaeffer, 2007])

Efficiency Issues

- analytical view
  - *classical techniques* of analysis and complexity
  - analytical techniques for *evolutionary algorithms*
- experimental view
  - *benchmarking* – quite a lot of data (DIMACS, network analysis benchmarks, real-world networks, etc.)
  - clique covering – easy evaluation and comparison, $\vartheta(G)$ is a particular number
  - graph clustering – not so straightforward, comparison to manually created solutions

Evaluation Techniques for Stochastic Graph Algorithms

- analytical techniques
  - a combination of *classical graph-theoretical approach* and *evolutionary algorithm analysis*
  - the choice of analytical method depends on the studied issue
- experimental techniques
  - optimality, success rate, statistical significance, etc.
  - *“When, we do not know, how to analyze...“*

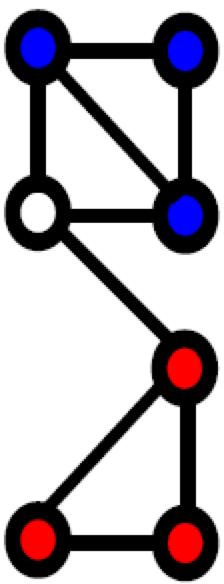
Analytical Techniques for Evolutionary Algorithms [Neumann and Witt, 2010]

- fitness-based landscape partitions
  - the search space is divided into *m* partitions, where the last one contains *only the optimum*
  - *probability of augmentation* – a lower bound on the probability that the algorithm jumps from partition *i* to *i+1* (p_i)
  - *waiting time* – the number of iterations, until the algorithm jumps to a higher partition (from geometric distribution, its expectation is $1/p_i$)
  - *expected time complexity* – the sum of waiting times, until partition *m* is reached

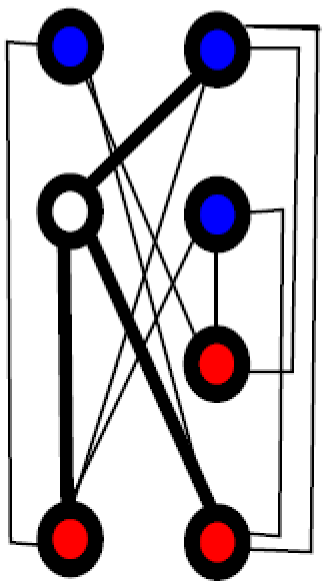
An Order-based Representation for CCP [Chalupa, 2012]

An Order-based Representation for CCP

- genotype-phenotype mapping based approach
  - greedy graph coloring [Welsh and Powell, 1967] can be used
  - the key issue is efficiency for real-world graphs
- $\overline{G}$ – graph coloring
  - to choose a color, we have to scan the neighbors
- G – clique covering
  - to choose a color, it is not enough to scan neighbors (without an additional information)



G



$\overline{G}$

Greedy Clique Covering (GCC)

Greedy Clique Covering

Input: graph $G = [V, E]$

permutation $P = [P_1, P_2, ..., P_{|V|}]$ of vertices in V

Output: clique covering S of G

for $c = 1..|V|$

  $sizes(c) = 0$

for $i = 1..|V|$

  $j = P_i$

  $c = find\_equal(\Gamma(v_j, c), sizes(c))$

  $V_c = V_c \cup \{v_j\}$

return $S = \{V_1, V_2, ..., V_k\}$

Optimality / Suboptimality Issues in GCC

- the basic issue in GCC – optimality
  - *Theorem:* For an arbitrary graph $G = [V,E]$, there is a permutation, for which the greedy clique covering will produce the optimal solution with $\vartheta(G)$ cliques.
  - *Proof:* Let $S = \{V_1, V_2, \ldots, V_{\vartheta(G)}\}$ be the optimal solution to the CCP. Then, the optimal permutation P can be constructed in the way that the vertices from the same classes are next to each other in P, i.e. $P = [V_{s1}, V_{s2}, \ldots, V_{s\vartheta(G)}]$, where $s_1, s_2, \ldots, s_{\vartheta(G)}$ is an arbitrary permutation of integers from 1 to $\vartheta(G)$. Since vertices of each of the subpermutations form the correct cliques, this permutation will surely lead to the optimal clique covering. QED.

Efficiency Issues in GCC

- GCC
  - computational complexity $O(|E(G)|)$
  - space complexity $O(|V|)$
- greedy graph coloring
  - computational complexity $O(|E(G')|)$
  - space complexity $O(|V|^2)$
- GCC is more efficient for sparse graphs
  - with current implementation techniques, GCC is faster than greedy coloring for graphs with density less than ca. *4/21*

Stochastic Order-based Approach to CCP: Iterated Greedy (IG) Algorithm

Block-based Mutation

- block-based properties of the representation
  - the identified cliques represent blocks of the solution
  - by reordering but (internally) preserving these blocks, the solution can be equally good or even superior to the previous one, similarly to the coloring problem [Culberson and Luo, 1996]
  - thus, although IG reminds one of random optimization, the fitness behaves similarly to local search
- reorderings of permutations
  - random order, reverse order

Iterated Greedy Algorithm with GCC

The IG heuristic for the CCP

Input: graph $G = [V, E]$
Output: clique covering S of G

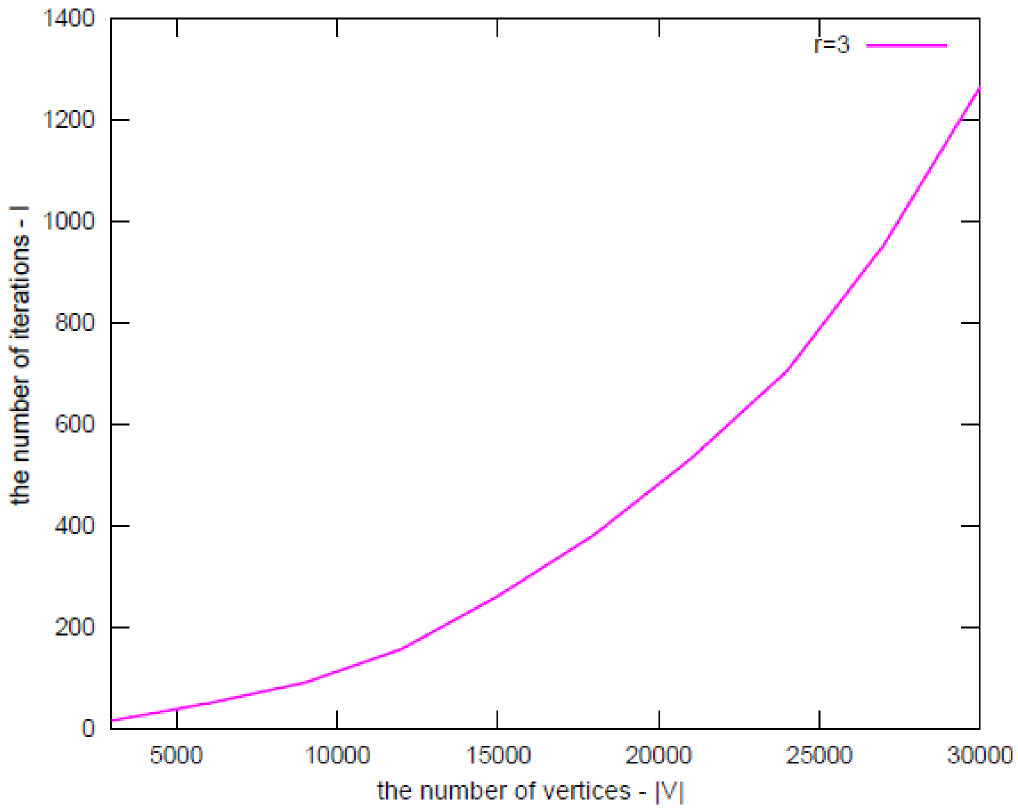
$P = random\_permutation(1, 2, ..., |V|)$
for $i = 1..I_{max}$
    $\{V_1, V_2, ..., V_k\} = greedy\_clique\_covering(G, P)$
    if $\vartheta^*(G)$ is known and $k = \vartheta^*(G)$
        return $S = \{V_1, V_2, ..., V_k\}$
    with p_{rev} probability
        $P = [V_k, V_{k-1}, ..., V_1]$
    else
        $P = random\_permutation(V_1, V_2, ..., V_k)$
return $S = \{V_1, V_2, ..., V_k\}$

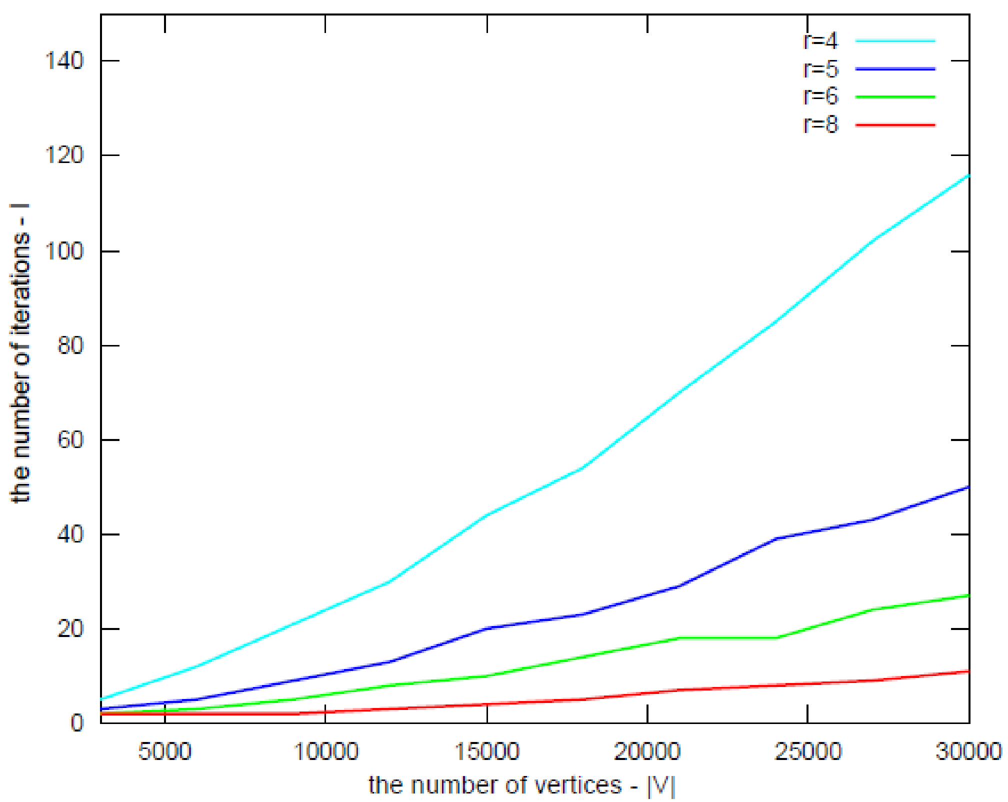
IG on Graphs with Planted Cliques

- a simple model of “clustered“ graphs
  - $\vartheta(G)$ embedded cliques of constant size r
  - probability p_{out} of generating an edge between two cliques
  - complements of k-colorable graphs in the coloring problem [Culberson and Luo, 1996]
- the key questions
  - How hard is it to find the right solution with $\vartheta(G)$ cliques?
  - How much time does IG need to find them?

Running time of IG on Sparse Graphs with Planted Cliques

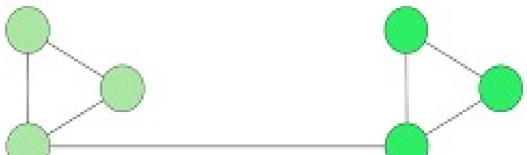
- empirical study of the performance of IG
  - $|V| = 3000 \Leftrightarrow 30000$, $r = 3 \Leftrightarrow 8$, $p_{out} = 10^{-3}$
  - p_{out} is small ➔ results indicate polynomial performance

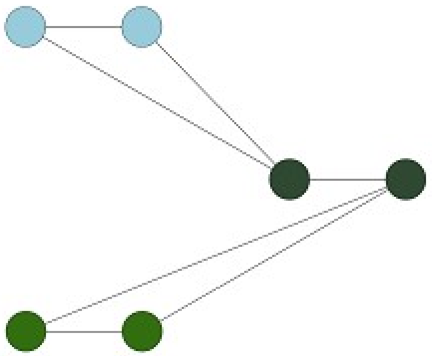


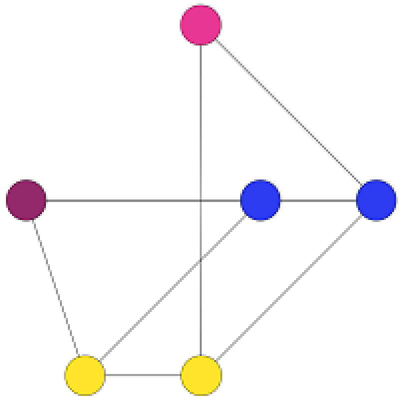


Analytical View on the Behavior of IG on Graphs with Planted Cliques

- overestimation by GCC
  - suppose that we re-represent the permutation $[v_1, v_2, \ldots, v_{|V|}]$ as $[[v_1, v_2], [v_2, v_3], \ldots, [v_{|V|-1}, v_{|V|}]]$
  - there are two ways, how GCC overestimates
    1. an inter-clique edge between two cliques precedes all intra-clique edges from the cliques it connects
    2. an inter-clique couple $[v_i, v_{i+1}]$ without an edge precedes a vertex adjacent to both v_i and v_{i+1}, which is in the same clique as v_{i+1}, but the First Fit strategy will falsely put in the same clique as v_i







Analytical View on the Behavior of IG on Graphs with Planted Cliques

- overestimation in sparse biclique graphs
  - complements of bipartite graphs
  - *Theorem:* Let G = [V,E] be a graph with planted cliques for $\vartheta = |V|/r = 2$ and $|E|_{out} < r$. Then, for each clique covering generated by GCC, a random reordering of its cliques will lead to the optimum with probability at least $1/[|V|/r+r-1]$.
  - *Proof:* By induction from small cases, evaluated exhaustively. An important implication of the property that $|E|_{out} < r$ is that there is a clique inside one of the expected ones.
  - *Consequence*: On these graphs, IG finds optimal clique covering in $O(|V|^3)$ time.

Analytical View on the Behavior of IG on Graphs with Planted Cliques

- generalization of the previous result
  - *Theorem:* Let $G = [V,E]$ be a graph with planted cliques $K_{r,1}, K_{r,2}, \ldots, K_{r,|V|/r}$. Suppose that $S_i = \{V_{1,i}, V_{2,i}, \ldots, V_{ki,i}\}$ is the current clique covering at the *i*-th iteration of IG. Furthermore, suppose that at each iteration *i*, there are *j* and *m*, such that there is a clique $G(V_{ki,j}) \in S_i$, which is a subgraph of some expected clique $K_{r,m}$ ($G(V_{ki,j}) \neq K_{r,m}$). Then, IG with GCC and random reorderings will converge to the optimal solution in $O(|V|^4)$ time.
  - *Proof:* A sketch: At each iteration, there is a clique $G(V_{ki,j})$ that is a subgraph of some of the expected cliques. This implies an $O(|V|)$ waiting time for an augmentation. The structure of the graph also implies that the number of fitness levels is $O(|V|)$. Overall, this implies an $O(|V|^4)$ upper bound.

Experimental Evaluation

- three algorithms
  - BRE - Brélaz's coloring heuristic
  - SAT-GCC – saturation-based GCC (permutation is determined greedily)
  - IG-GCC – iterated greedy with GCC (permutation is evolved)
  - best results are highlighted with bold

G	BRE	SAT-GCC	IG-GCC
Erdős-Rényi uniform random graphs			
$unif1000\_0.1$	299	310	**243**
$unif5000\_0.1$	1241	1288	**1066**
$unif10000\_0.1$	2326	2389	**2025**
$unif20000\_0.01$	7640	7817	**6387**
Leighton graphs from DIMACS instances.			
$le450\_15a$	85	89	**80**
$le450\_15b$	92	90	**82**
$le450\_15c$	68	74	**57**
$le450\_15d$	73	73	**57**
$le450\_25a$	**91**	92	**91**
$le450\_25b$	81	82	**80**
$le450\_25c$	61	59	**54**
$le450\_25d$	60	59	**51**
Social graphs			
$soc2000$	**1471**	1473	**1471**
$soc10000$	6619	6633	**6618**
$soc20000$	12770	12804	**12764**

Current Research

- analysis of order-based algorithms
  - IG – it seems that on one hand, IG is very efficient for graphs with planted cliques, as well as real world data
  - however, there are graphs, where IG performs really badly
  - RLS – another interesting algorithm, using vertex-based mutations, instead of block-based
  - seems more robust but not so efficient in practice

Multicriteria Construction Procedures (MCPs) for Graph Clustering [Chalupa and Pospíchal, 2012]

Multicriteria Construction Procedures

- constructive algorithms for graph clustering
  - a mapping of a permutation of vertices to a graph clustering

Algorithm 1: A General Framework for an MCP

A General Framework for an MCP

Input: graph $G = [V, E]$
permutation $P = [P_1, P_2, \ldots, P_{|V|}]$ of vertices
Output: a clustering S of G

for $i = 1..|V|$
    $j = P_i$
    $c = find\_cluster(v_j)$
    $V_c = V_c \cup \{v_j\}$
    $update\_auxiliary\_data(V_c)$
return $S = \{V_1, V_2, \ldots, V_k\}$

Criteria for Graph Clustering

1. Each vertex is clustered and the clusters are non overlapping.
2. The clusters are more dense than the whole graph:
   $\forall i = 1..k \; d(G(Vi)) > d(G)$, where d is the density.
3. The relative connectivity of a vertex to be newly added to the cluster must be higher than its relative connectivity to the residual, currently non-clustered subgraph:
   $w_c / |V_{c,i}| > \delta_r / [|V_r|-1]$
   where $V_{c,i}$ is the set of vertices in cluster c at the iteration i of the MCP,
   w_c is the number edges, brought into the cluster by the vertex to be newly added and
   $|V_r|$ and δ_r are the number of vertices and the degree of the newly added vertex in the subgraph containing only the currently non-clustered vertices.

Criteria for Graph Clustering

4. If there are more candidate clusters, the one with highest connectivity is taken:
$c = \arg_c \max w_c / |V_{c,i}|$
where for the cluster c, $w_c / |V_{c,i}|$ must be a feasible value, according to the previous rule.
5. The vertex to be newly added must bring at least as many edges, as is the current average intra-cluster degree in the particular cluster, while a small tolerance τ may be sometimes allowed:
$w_c + \tau \geq 2|E_{c,i}| / |V_{c,i}|$,
where $|E_{c,i}|$ is the number of edges in $G(V_{c,i})$.

Multicriteria Construction Procedure Based on Density and Connectivity (MCP-DC)

- MCP-DC implements the previous 5 criteria as follows
  - local density needed in criterion 2 is fulfilled if:
    $d(G)\,|V_{c,i}|\,(|V_{c,i}|+1) - 2|E_{c,i}| - 2w_c < 0$
  - the local connectivity in criterion 3 is fulfilled if the following holds:
    $|V_{c,i}| - w_c\,[|V_r| - 1] / \delta_r < 0$
  - the maximization of the connectivity in criterion 4, i.e. the ratio $w_c / |V_{c,i}|$, can be implemented simultaneously with criterion 3, since the necessary values are calculated in the verification of criterion 3
  - the criterion 5 yields the following condition, where $\tau \geq 0$ is a parameter of tolerance for the intra-cluster degree of the newly added vertex:
    $2|E_{c,i}| / |V_{c,i}| - \tau - w_c \leq 0$

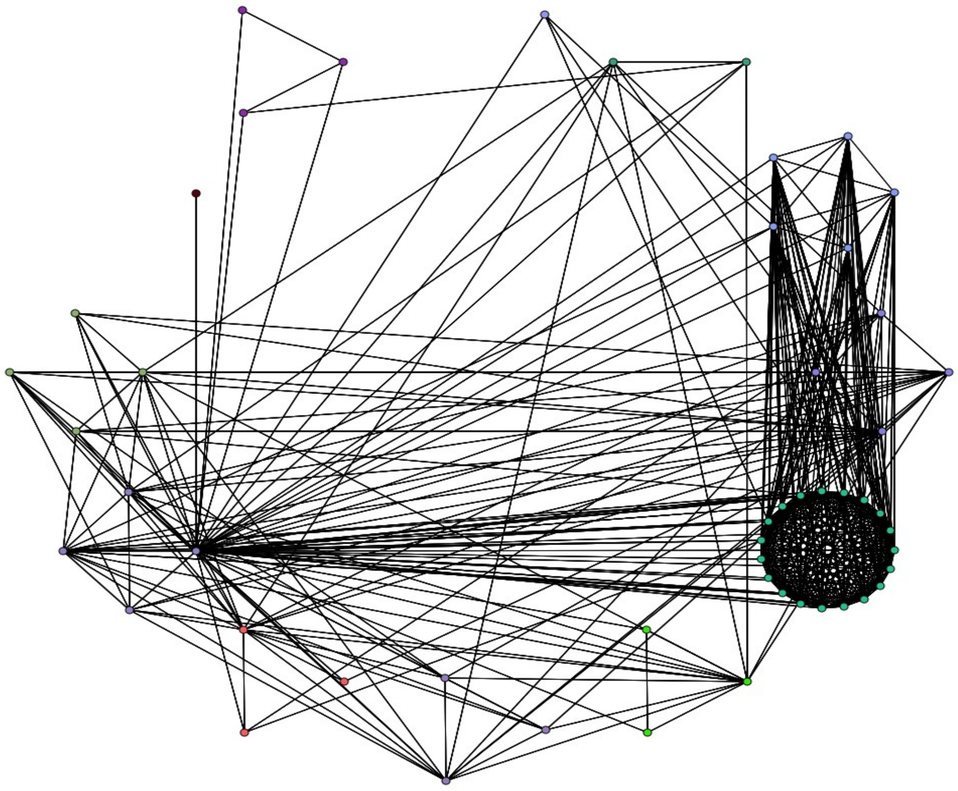
Multicriteria Construction Procedure Based on Density and Connectivity (MCP-DC)

- the advantage of this implementation of criteria in MCP-DC is that the complexity is favorable for sparse graphs
- *Theorem.* MCP-DC can be implemented to run in $O(\delta|V|) = O(|E|)$ time.
- *Proof.* $|V_{c,i}|$ and $|E_{c,i}|$ can be trivially recalculated in *O(1)* time per iteration. The previous formulations of the MCP-DC criteria can be implemented by iterative subtracting of a constant (in the cases of criteria 2 and 5) or the ratio $[|V_r| - 1] / \delta_r$ (in the case of criterion 3) from the respective values. Explicit storage of values w_c yields the same for criterion 4. Restoration of the former values after subtraction can be done by simulating the inverse process. All these operations need $O(\delta)$ average time per iteration, thus, they lead to an $O(\delta|V|) = O(|E|)$ running time of MCP-DC. QED.

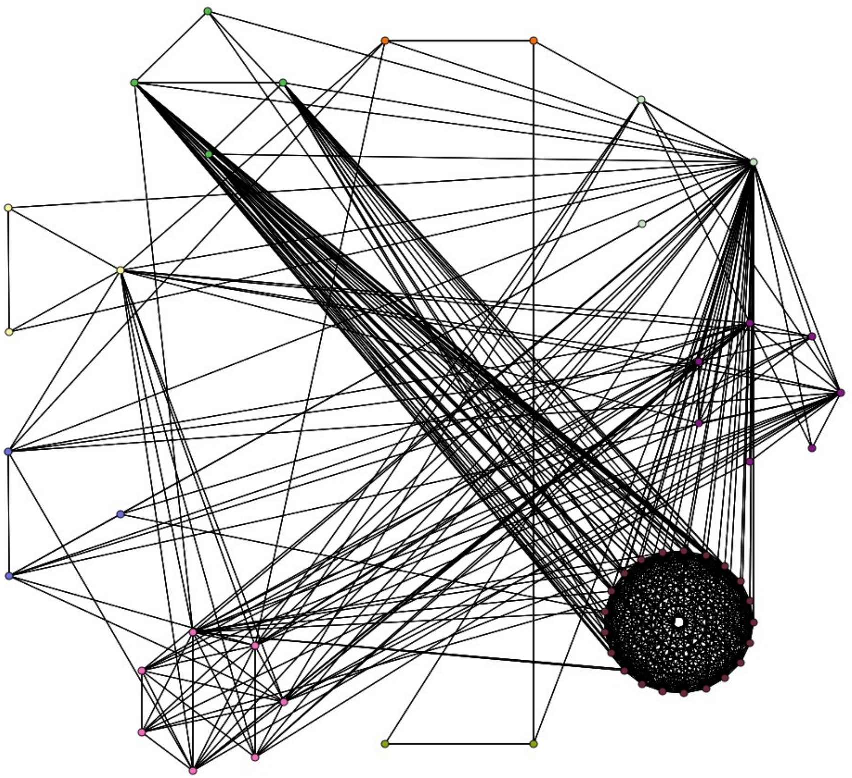
Metaheuristic Optimization for MCPs

- a simple local search algorithm
  - we begin with a random permutation of vertices and use an MCP to construct a clustering
  - *mutation:* at each iteration, we try a single random vertex exchange in the permutation and evaluate the new number of clusters using the MCP
  - *acceptance of mutation:* we accept if the new clustering has *at most as many clusters* as the current one
  - *stopping criterion:* maximum of s_{max} iterations without improvement

The Emergence of a Good Clustering

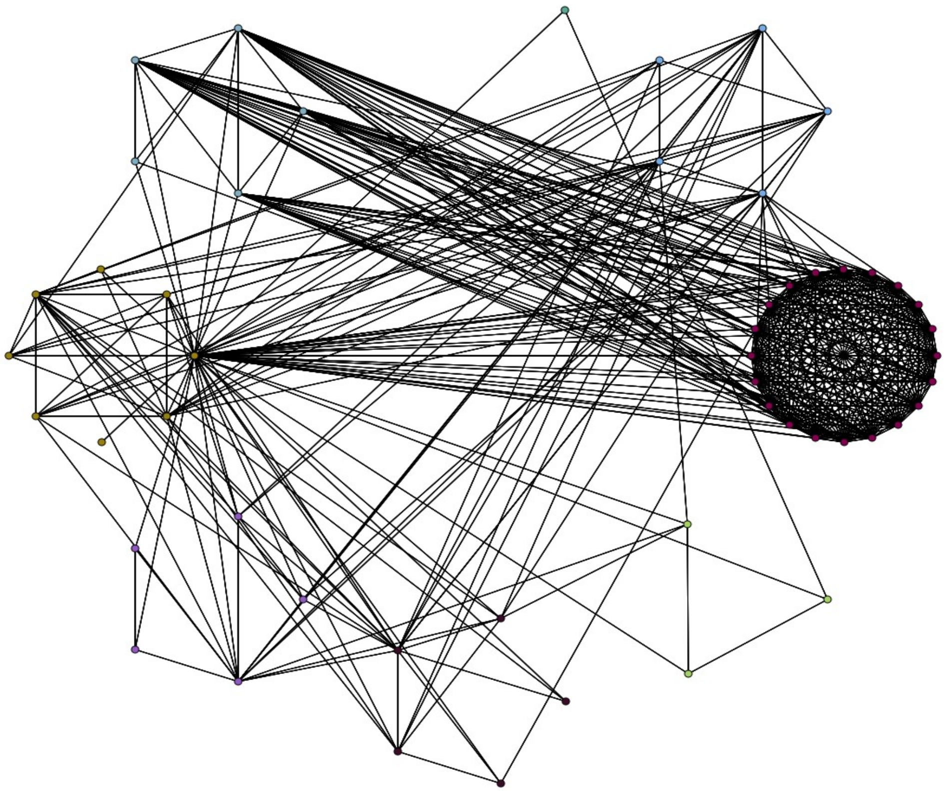


0 iterations

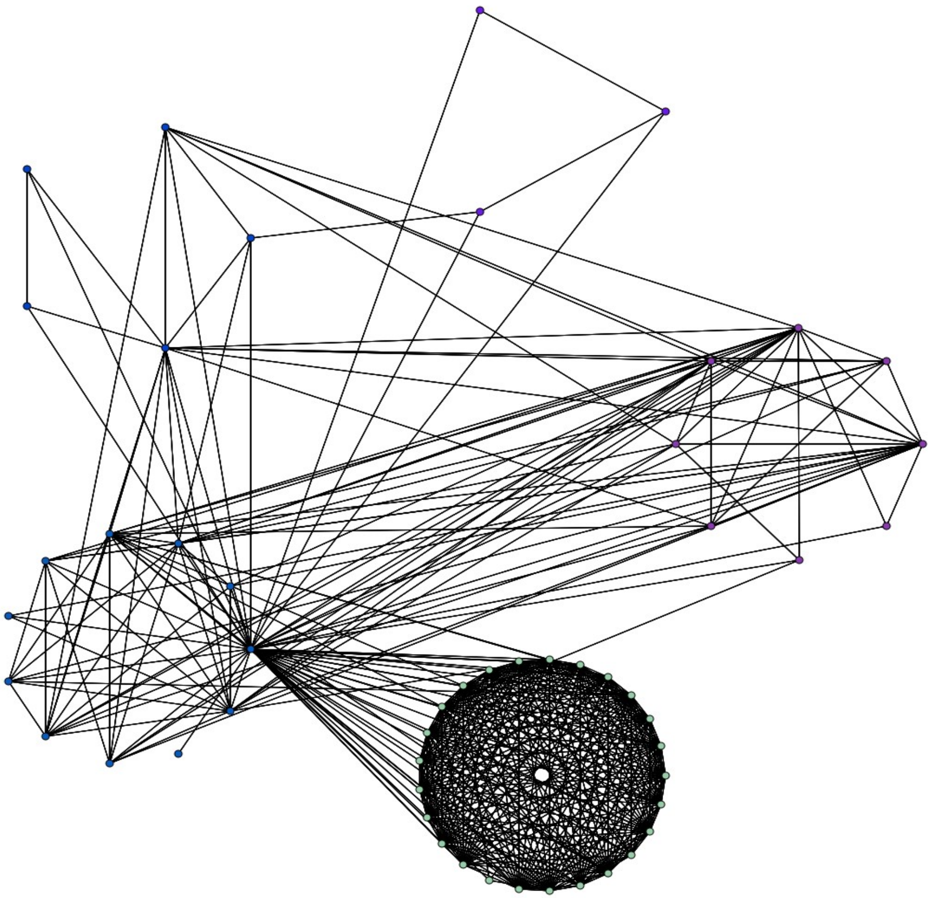


100 iterations

The Emergence of a Good Clustering



1000 iterations



10000 iterations

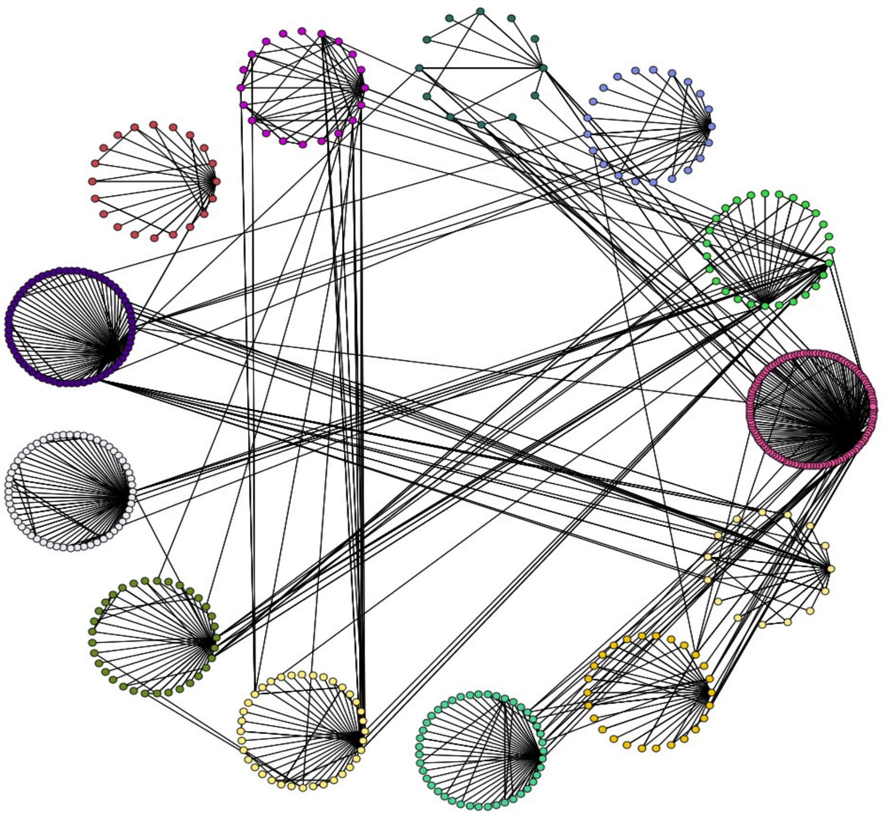
Results on Benchmark Instances

- a comparison of pure MCP-DC and MCP-DC with the metaheuristic on several graphs
  - network clustering benchmarks: *Zachary karate club* [Zachary, 1977] and *American college football network* [Girvan and Newman, 2002]
  - extracts of two different social networks
  - an artificial model from [Chalupa, 2011a]

source	$\|V\|,\|E\|$	s_{max}	τ	MCP-DC	MCP-DC+MH		
				k	k	iter.	time
Zachary karate club	$34,78$	5×10^3	1	7 - 15	2	7035	<1 s
American college football	$115,615$	10^6	0	18 - 23	10 - 12	1237965	252 s
Social network I	$52,830$	5×10^4	0	12 - 16	5 - 6	76194	9 s
Social network II	$500,924$	5×10^4	1	161 - 197	12 - 15	154964	71 s
Artificial model	$500,3536$	5×10^4	0	68 - 79	55 - 60	163449	188 s

Other Results

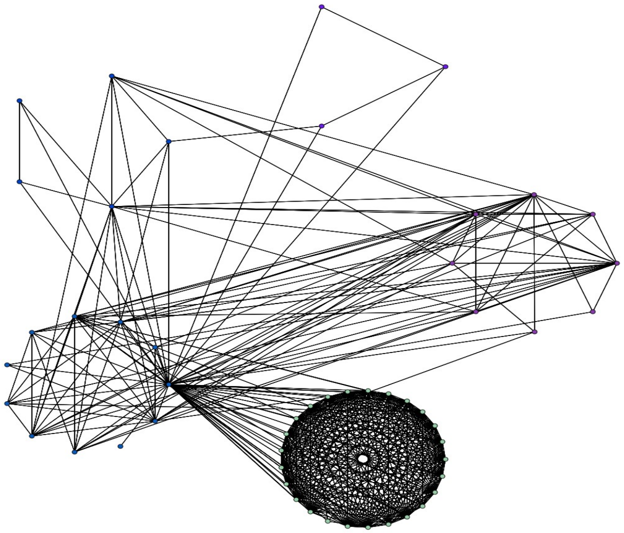
- a clustering of data obtained from a Slovak social network
  - shows a clear presence of hubs – in MCP, we preferred a centrality-based strategy



Conclusions

Conclusions

- introduction to stochastic graph algorithms
  - problems: clique covering, graph clustering
  - strategies, methodologies of evaluation
- an order-based representation for CCP
  - interesting analytical results and promising on real-world networks
- multicriteria construction procedures (MCPs) for graph clustering
  - show promise in both clustering and determining the nature of clustering problem formulation



Thank you for your attention!

chalupa@fiit.stuba.sk

References

[Chalupa, 2011a] D. Chalupa. On the Ability of Graph Coloring Heuristics to Find Substructures in Social Networks. Information Sciences and Technologies, Bulletin of ACM Slovakia, 3(2):51–54, 2011.

[Chalupa, 2011b] D. Chalupa. Population-based and learning-based metaheuristic algorithms for the graph coloring problem. In P. L. Lanzi and N. Krasnogor, editors, Proceedings of the 13th annual conference on Genetic and evolutionary computation, GECCO ’11, pages 465–472, New York, NY, USA, 2011. ACM.

[Chalupa, 2012] D. Chalupa. On the Efficiency of an Order-based Representation in the Clique Covering Problem. In J. Moore and T. Soule, editors, Proceedings of the 14th annual conference on Genetic and evolutionary computation, GECCO ’12, pages 353–360, New York, NY, USA, 2012. ACM.

[Chalupa and Pospíchal, 2012] D. Chalupa and J. Pospíchal: Metaheuristically Optimized Multicriteria Clustering for Medium-scale Networks. In Proceedings of SOCO'12, Advances in Intelligent Systems and Computing 188 (2012). Springer, Berlin/Heidelberg, pages 337-346.

[Girvan and Newman, 1999] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 99(12):7821–7826, 2002.

References

[Sun et al., 2008] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos. Less is more: Sparse graph mining with compact matrix decomposition. Statistical Analysis and Data Mining, 1(1):6–22, 2008.

[Tang et al., 2011] J. Tang, T. Wang, J. Wang, Q. Lu, and W. Li. Using complex network features for fast clustering in the web. In S. Sadagopan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, editors, Proceedings of the 20th international conference companion on World wide web, WWW '11, pages 133–134, New York, NY, USA, 2011. ACM.

[Gao et al., 2009] L. Gao, P. Sun, and J. Song. Clustering algorithms for detecting functional modules in protein interaction networks. Journal of Bioinformatics and Computational Biology, 7(1):217–242, 2009.

[Boyer et al. 2005] F. Boyer, A. Morgat, L. Labarre, J. Pothier, and A. Viari. Syntons, metabolons and interactons: an exact graph-theoretical approach for exploring neighbourhood between genomic and functional data. Bioinformatics, 21(23):4209–4215, 2005.

[Patillo et al., 2012] J. Pattillo, N. Youssef, and S. Butenko. Clique Relaxation Models in Social Network Analysis. In M. T. Thai and P. M. Pardalos, editors, Handbook of Optimization in Complex Networks, pages 143–162. Springer, 2012.

References

[Rothenberg et al., 1996] R. B. Rothenberg, J. J. Potterat, and D. E. Woodhouse. Personal Risk Taking and the Spread of Disease: Beyond Core Groups. The Journal of Infectious Diseases, Supplement 2, 174:S144–S149, 1996.

[Burke et al., 2007] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu. A graph-based hyperheuristic for educational timetabling problems. European Journal of Operational Research, 176(1):177 – 192, 2007.

[Smith et al., 1998] D. H. Smith, S. Hurley, and S. U. Thiel. Improving heuristics for the frequency assignment problem. European Journal of Operational Research, 107(1):76 – 86, 1998.

[Ester et al., 1996] M. Ester, H. P. Kriegel, J. Sander, X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, Usama M. Fayyad. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231. 1996.

[Condon and Karp, 1999] A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. In D. S. Hochbaum, K. Jansen, J. D. P. Rolim, and A. Sinclair, editors, Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization Problems: Randomization, Approximation, and Combinatorial Algorithms and Techniques, RANDOM-APPROX ’99, pages 221–232, London, UK, 1999. Springer.

References

[Watts, 1999] D. J. Watts. Small Worlds. Princeton University Press, 1999.

[Albert and Barabási, 2002] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. Reviews of Modern Physics, 74(1):47–97, 2002.

[Zuckerman, 2007] D. Zuckerman. Linear degree extractors and the inapproximability of Max Clique and Chromatic Number, Theory of Computing 3: 103–128. 2007.

[Hertz and De Werra, 1987] A. Hertz and D. de Werra. Using tabu search techniques for graph coloring. Computing, 39(4):345–351, 1987.

[Morgenstern, 1996] C. Morgenstern. Distributed coloration neighborhood search. In D. S. Johnson and M. Trick, editors, Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, pages 335–358. 1996.

[Titiloye and Crispin, 2011] O. Titiloye, A. Crispin. Quantum annealing of the graph coloring problem. Discrete Optimization, 8(2):376-384. 2011.

[Galinier and Hao, 1999] P. Galinier and J. K. Hao. Hybrid Evolutionary Algorithms for Graph Coloring. Journal of Combinatorial Optimization, 3(4):379–397, 1999.

[Schaeffer, 2007] S. E. Schaeffer. Graph clustering. Computer Science Review, 1(1):27–64, 2007.

References

[Porumbel et al., 2010] D. C. Porumbel, J. K. Hao, and P. Kuntz. An evolutionary approach with diversity guarantee and well-informed grouping recombination for graph coloring. Computers & Operations Research, 37(10):1822–1832, 2010.

[Culberson and Luo, 1996] J. C. Culberson and F. Luo. Exploring the k-colorable landscape with iterated greedy. In D. S. Johnson and M. Trick, editors, Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, pages 245–284. American Mathematical Society, 1996.

[Drechsler et al., 1999] N. Drechsler, W. Günther, and R. Drechsler. Efficient graph coloring by evolutionary algorithms. In B. Reusch, editor, Proceedings of the 6th International Conference on Computational Intelligence, Theory and Applications: Fuzzy Days, pages 30–39, London, UK, 1999. Springer.

[Neumann and Witt, 2010] F. Neumann, C. Witt. Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity. Springer Berlin/Heidelberg, 2010.

[Karp, 1972] R. M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, Complexity of Computer Computations, pages 85–103. Plenum Press, New York, NY, USA, 1972.

[Kaufman and Rouseeuw, 1990] L. Kaufman and P. J. Rousseeuw. Finding groups in data: an introduction to cluster analysis. Wiley. 1990.