Practical Fixed-Parameter Algorithms for Graph-Modeled Data Clustering

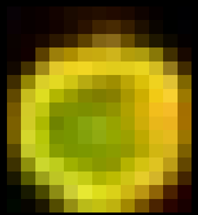
Sebastian Wernicke*

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Fed. Rep. of Germany
wernicke@minet.uni-jena.de

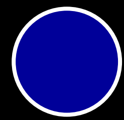
*Research supported by Deutsche Telekom Stiftung

Prologue

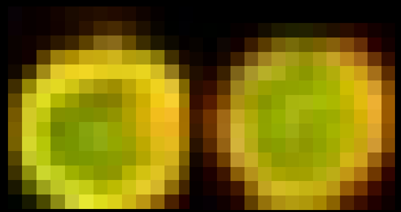
Graph-Modeled Clustering



Data Point

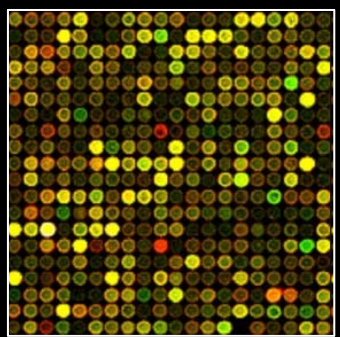


Vertex / Node

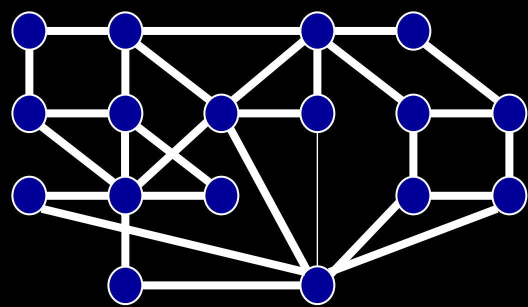


Correlation

Edge / Link

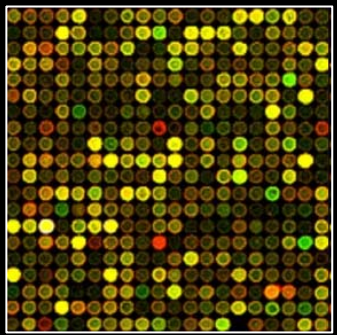


Data

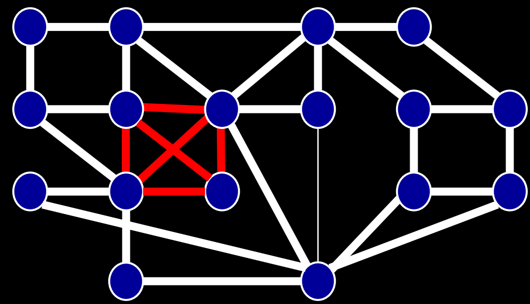


Graph / Network

Graph-Based Clustering



Cluster(s)
in Data



Dense / Complete
Subgraph(s)

Potential Tasks to Face

- Find a large complete subgraph (clique).
- Partition the graph into disjoint cliques.
- Cover the graph with cliques.

Toolbox for Hard Problems

Brute-Force

Sidestepping

Heuristics

Approximation Algorithms

(Integer) Linear Programming

Toolbox for Hard Problems

Brute-Force

Sidestepping

Heuristics

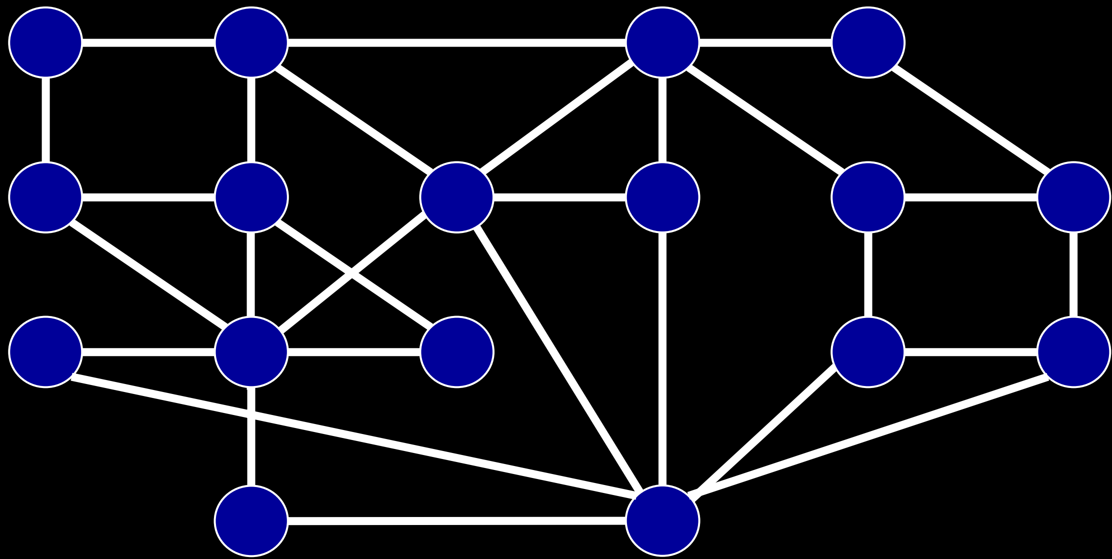
Approximation Algorithms

(Integer) Linear Programming

Fixed-Parameter Algorithms
- data reduction
- search trees
- and more!

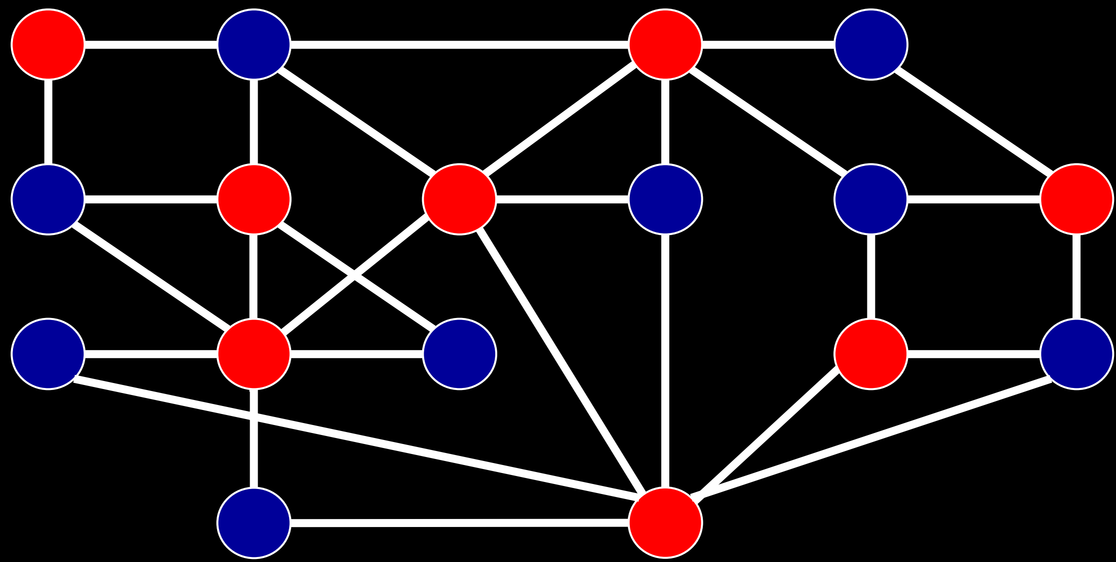
Part I
Fixed-Parameter Tractability Primer

Drosophila of Fixed-Parameter Algorithmics: Vertex Cover



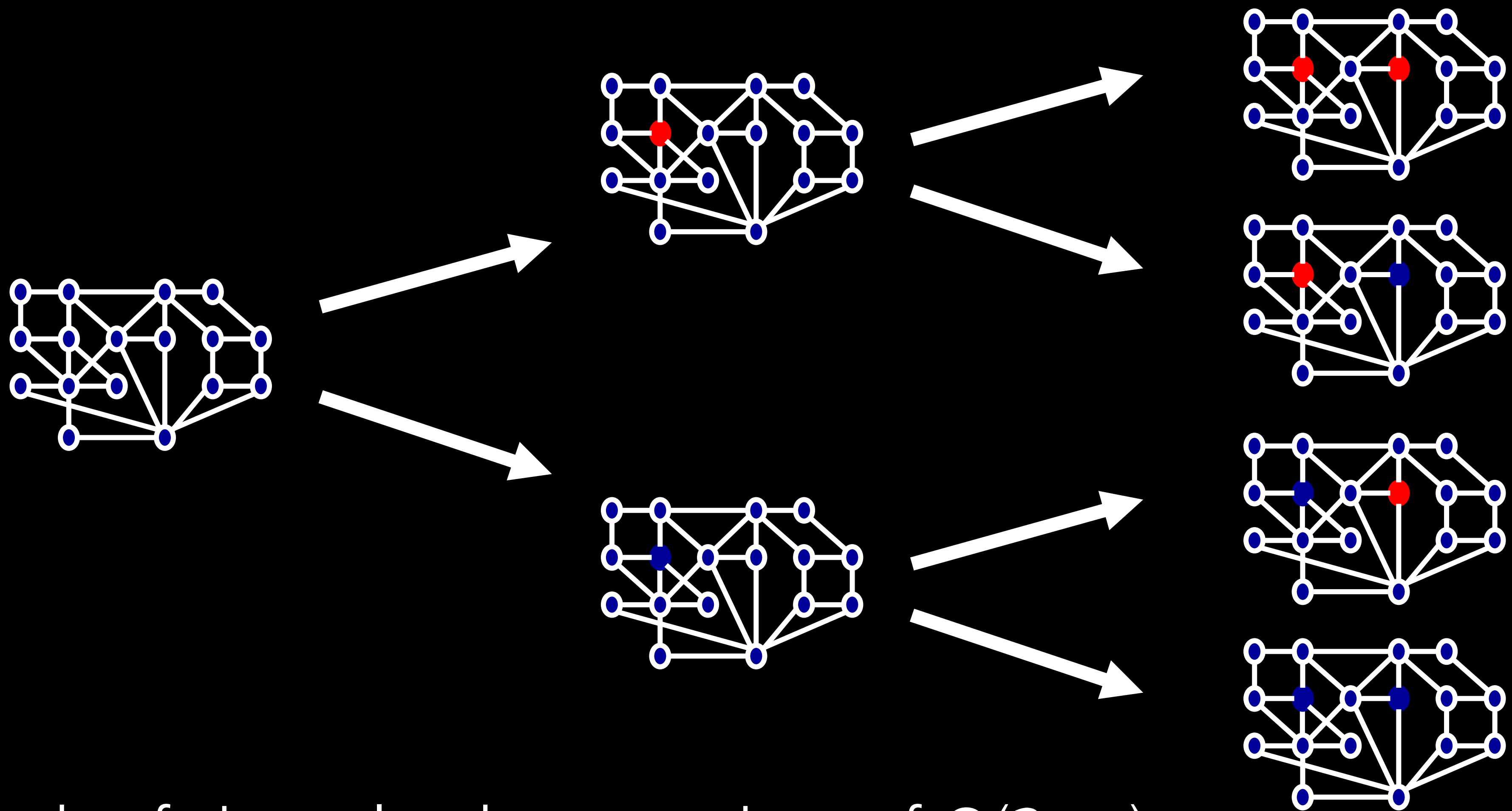
Given a graph *G*, find a set *C* of at most *k* vertices such that each edge has at least one endpoint in *C*.

Drosophila of Fixed-Parameter Algorithmics: Vertex Cover



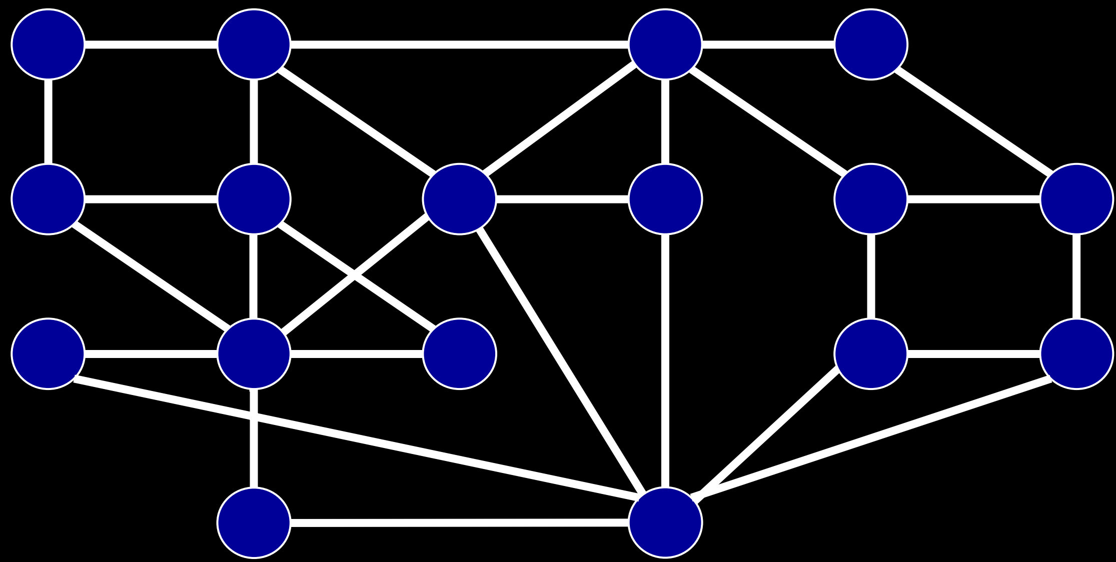
Given a graph *G*, find a set *C* of at most *k* vertices such that each edge has at least one endpoint in *C*.

Vertex Cover – Naïve Brute-Force Approach



Graph of size n leads to runtime of $O(2^n m)$.

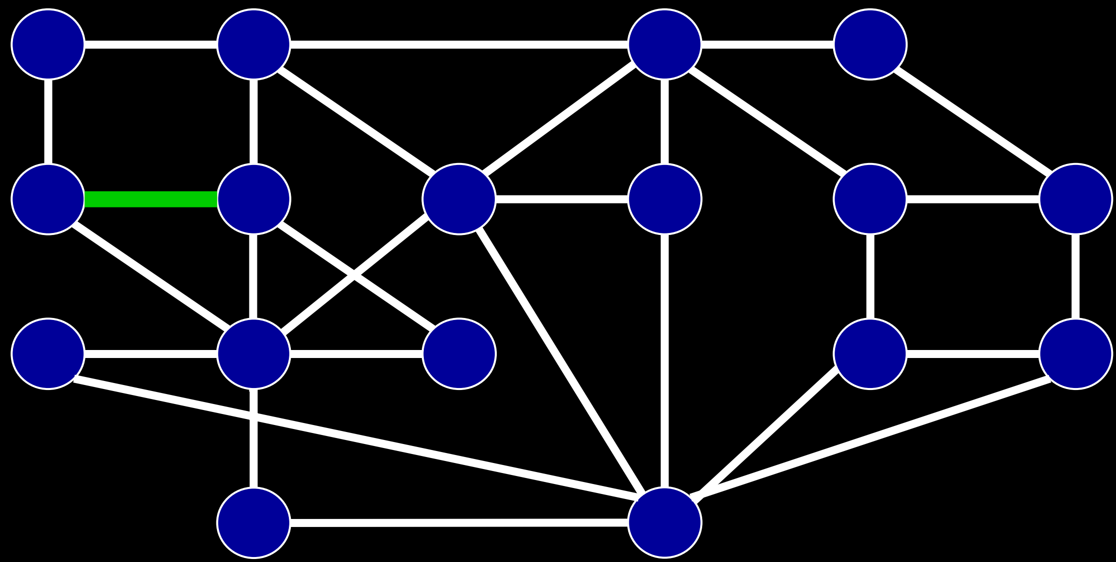
Vertex Cover – Better Approach



Given graph *G*, find a set *C* of at most *k* vertices such that each edge has at least one endpoint in *C*.

Main idea: For each edge, one endpoint *must* be in *C*.

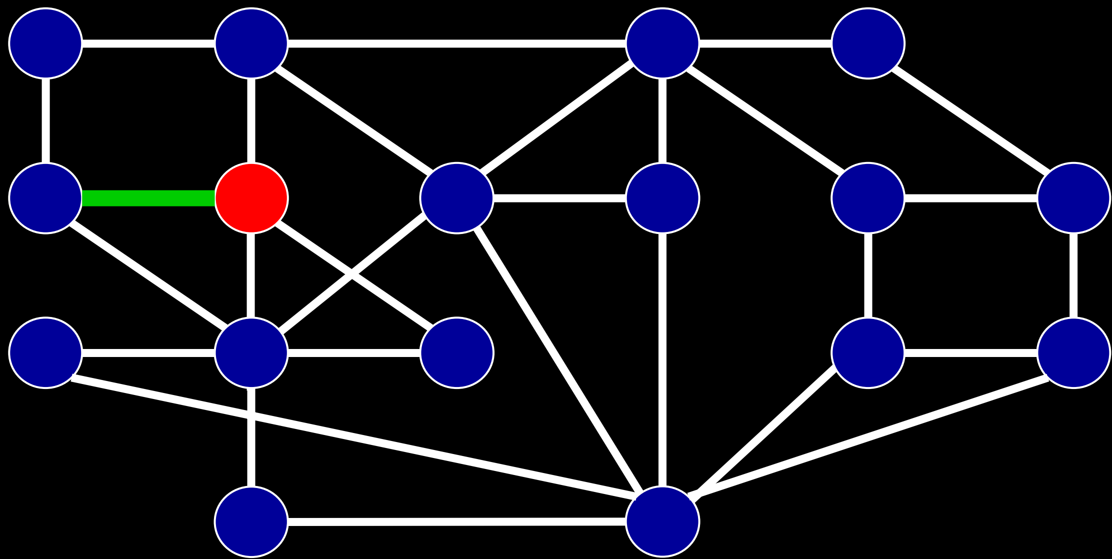
Vertex Cover – Better Approach



Given graph *G*, find a set *C* of at most *k* vertices such that each edge has at least one endpoint in *C*.

Main idea: For each edge, one endpoint *must* be in *C*.

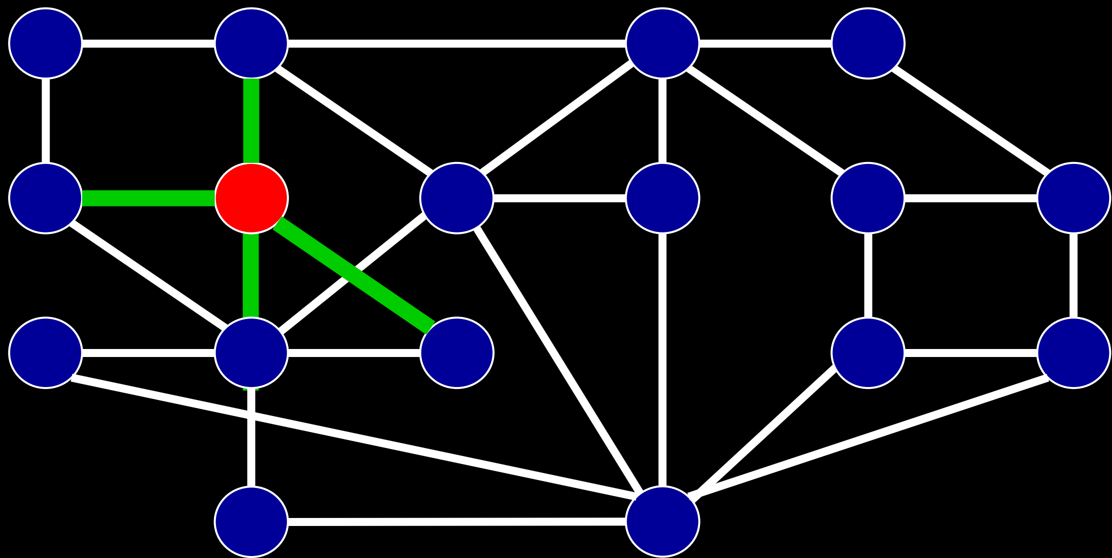
Vertex Cover – Better Approach



Given graph *G*, find a set *C* of at most *k* vertices such that each edge has at least one endpoint in *C*.

Main idea: For each edge, one endpoint *must* be in *C*.

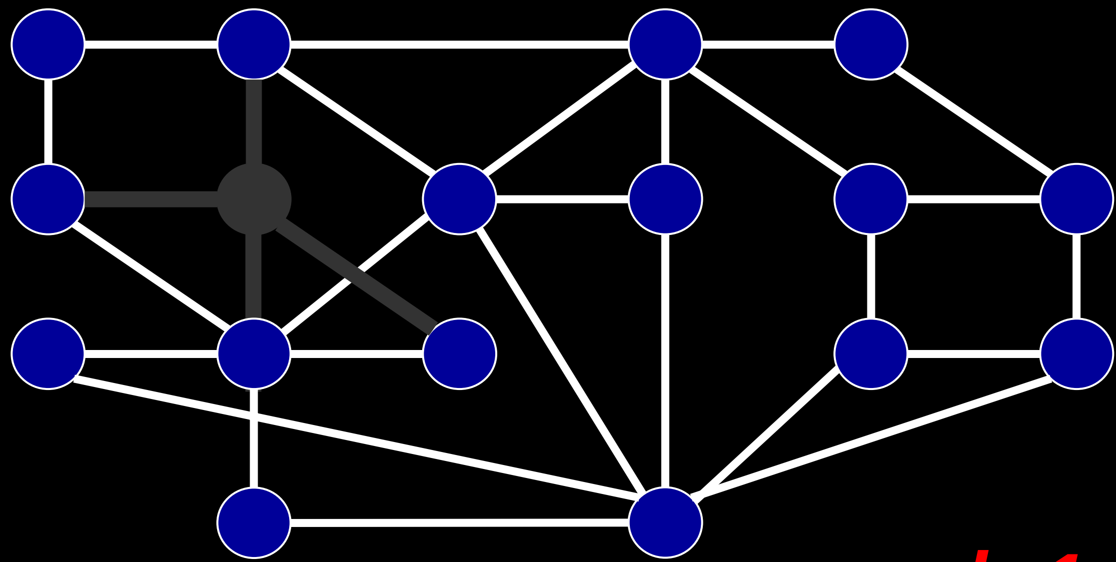
Vertex Cover – Better Approach



Given graph *G*, find a set *C* of at most *k* vertices such that each edge has at least one endpoint in *C*.

Main idea: For each edge, one endpoint *must* be in *C*.

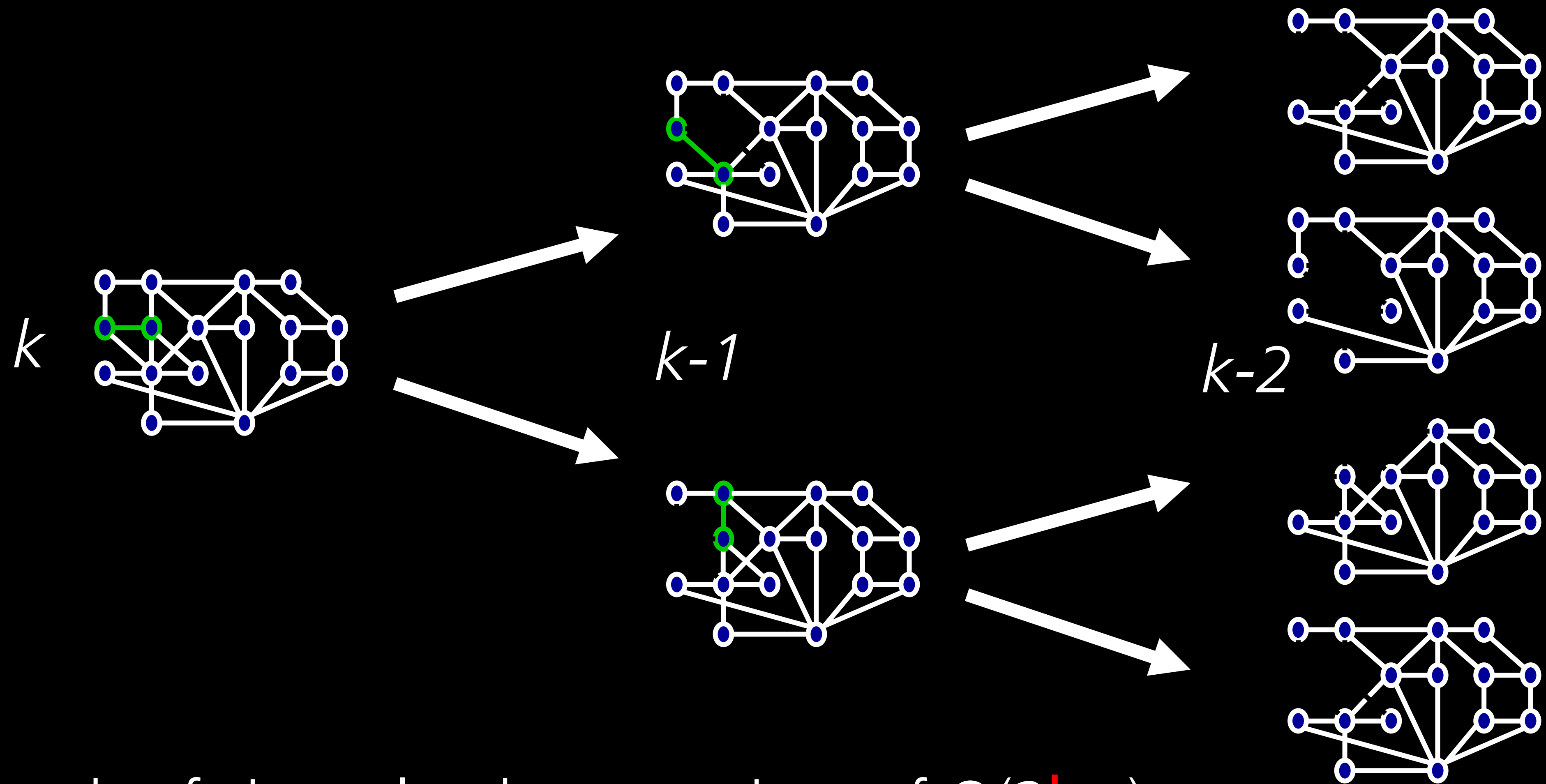
Vertex Cover – Better Approach



Given graph *G*, find a set *C* of at most ~~*k*~~ ***k-1*** vertices such that each edge has at least one endpoint in *C*.

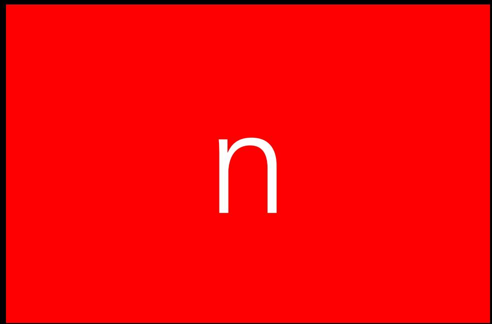
Main idea: For each edge, one endpoint *must* be in *C*.

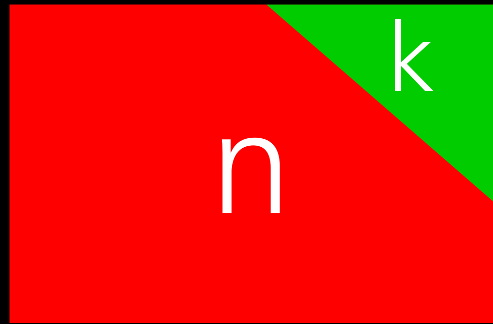
Vertex Cover – Fixed-Parameter Approach



Graph of size n leads to runtime of $O(2^{k}m)$.

Fixed-Parameter Tractability

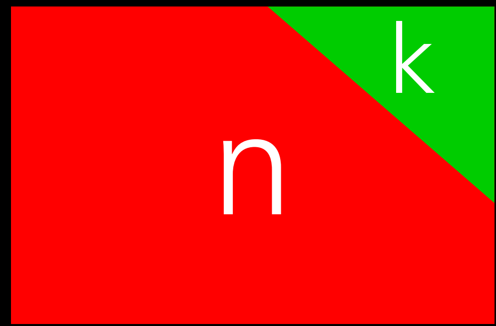




Classical complexity theory: One dimensional. Hard problems take *exponential time in instance size* to solve.

The fixed-parameter approach: Two-dimensional. Hard problem takes *exponential time in solution structure* to solve.

Fixed-Parameter Tractability

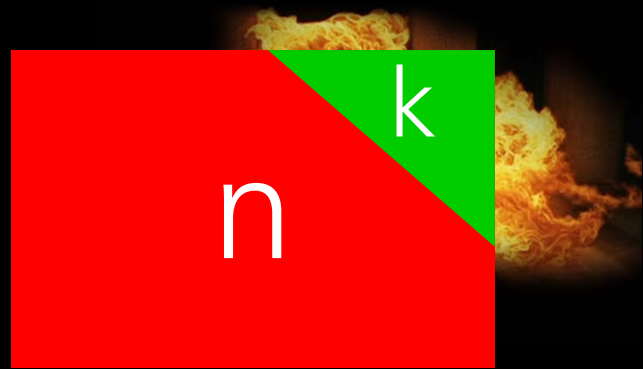


Classical complexity theory: One dimensional. Hard problems take *exponential time in instance size* to solve.

The fixed-parameter approach: Two-dimensional. Hard problem takes *exponential time in solution structure* to solve.

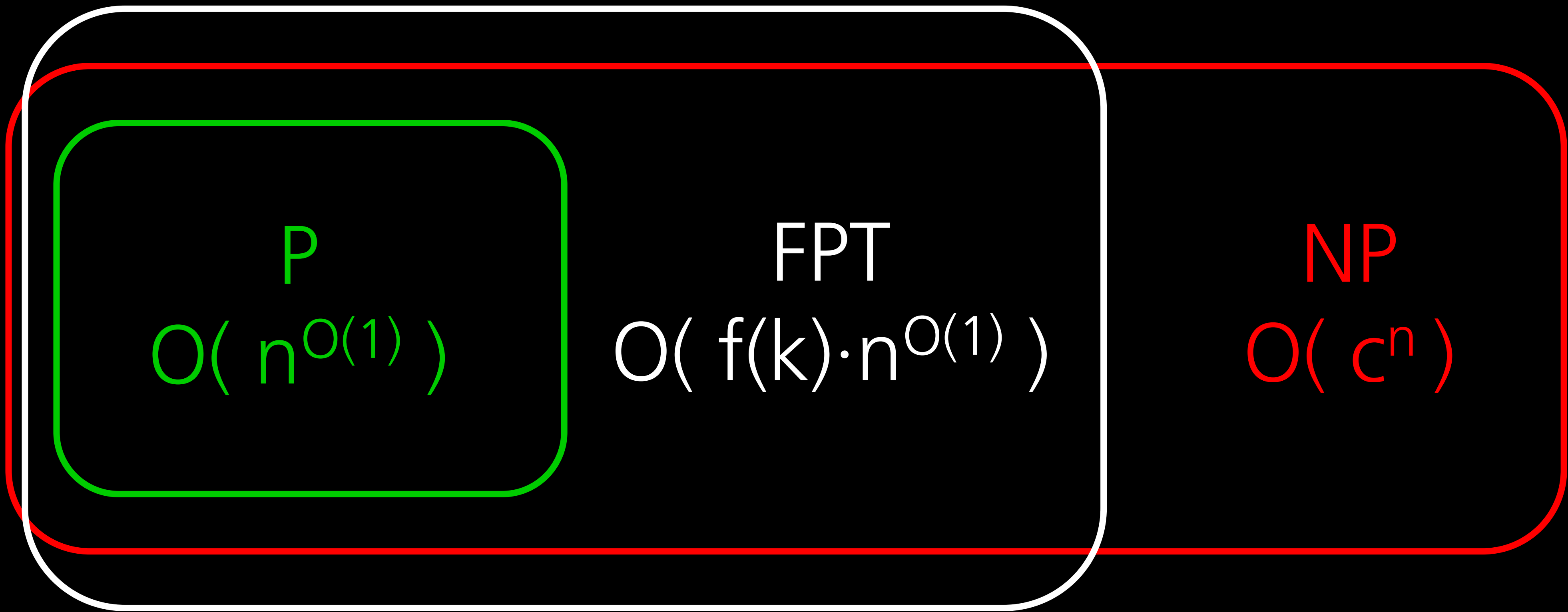
Fixed-Parameter Tractability

Classical complexity theory: One dimensional. Hard problems take *exponential time in instance size* to solve.



The fixed-parameter approach: Two-dimensional. Hard problem takes *exponential time in solution structure* to solve.

Fixed-Parameter Tractability



Key Facts About Fixed-Parameter Tractability

Data Reduction (Kernels)

If a problem is fixed-parameter tractable, we can polynomial-time reduce an instance of size *n* to an instance of size *f(k)* (called *kernel*). Example: A VERTEX COVER instance can be reduced to an instance of size *2k*, where *k* is the size of a minimum cover. [Chen et al., J. Alg., 2001]

Search Tree

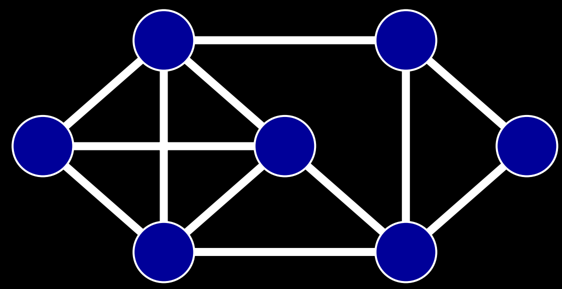
Search trees are depth-bounded by the parameter *k*.

Intractability

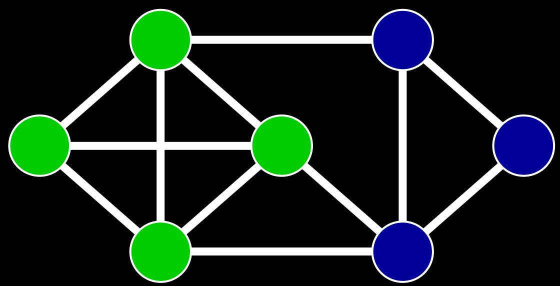
Not all problems are fixed-parameter tractable.

Part II
Case Studies

Case Studies

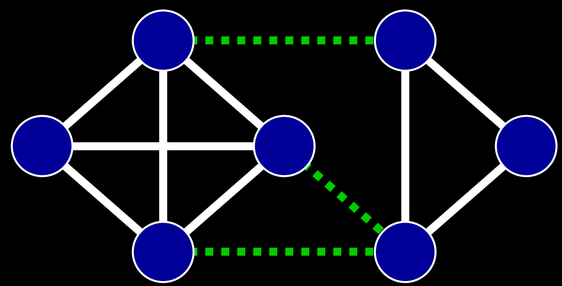


Input



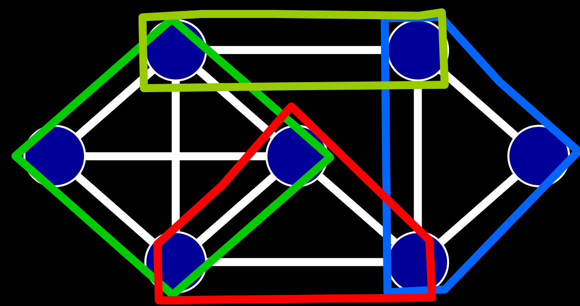
CLIQUE

Task: Find the largest clique in the graph.



CLUSTER EDITING

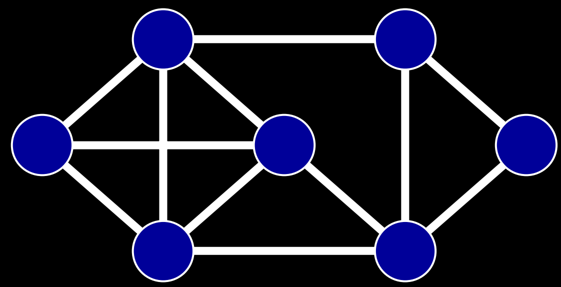
Task: Add / remove few edges so that cliques remain



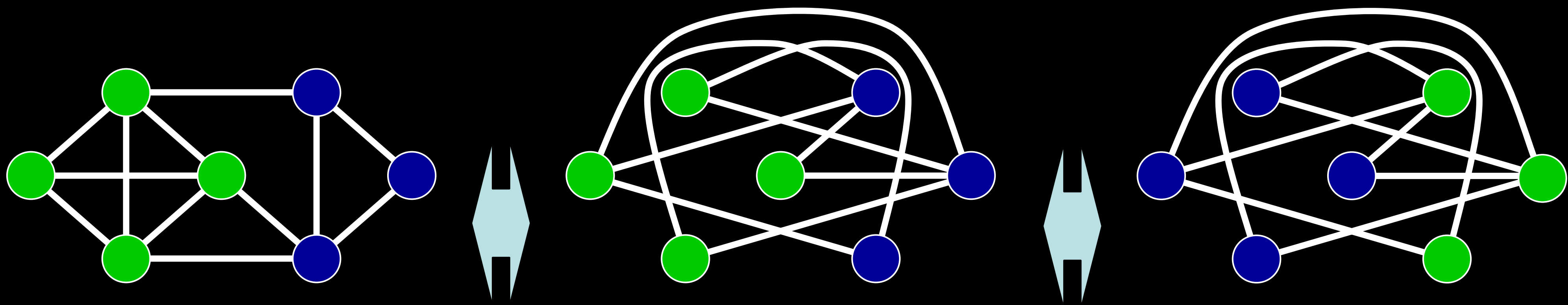
CLIQUE COVER

Task: Find few cliques to cover all edges.

CLIQUE – Main Idea



Input: An *n*-vertex graph



Graph has clique of size *(n-k)*.

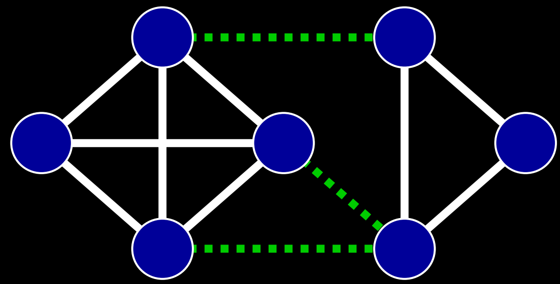
Complement graph has independent set of size *(n-k)*.

Complement graph has vertex cover of size *k*.

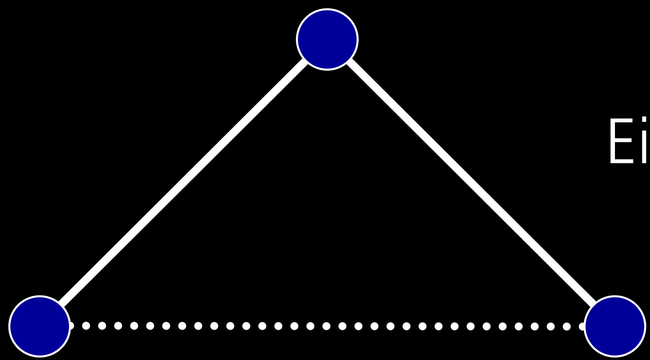
CLIQUE – Results

Vertex Cover	Can be solved optimally with a search-tree of size $O(1.28^k)$.
	Workhorse in practice is data reduction. [Abu-Khzam et al., Proc. ALENEX 2004]
Experimental	Graphs with 10^5 vertices and with $k \leq 300$ can be solved in practice [Cheetham et al., J. Com. Sys. Sci, 2003]
Intractability	Parameterized by the clique size is not fixed-parameter tractable. Graph must contain a large clique.

CLUSTER EDITING – Main Idea



Task: Add / remove few
edges so that cliques remain



Basic search tree strategy:
Either delete one of the edges
or add the missing one

CLUSTER EDITING – Results

Data Reduction

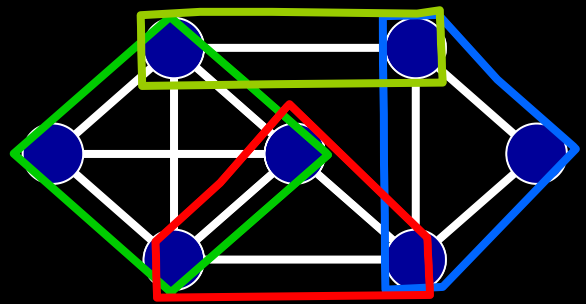
With parameter k, an instance can be reduced to a kernel of size k^3 in polynomial time.

Worst-Case Running Time

For CLUSTER EDITING, $O(1.92^k+|V|^3)$.
[Gramm et al., Algorithmica, 2004]

When only edge deletions are allowed (CLUSTER DELETION), this reduces to $O(1.77^k+|V|^3)$.
[Gramm et al., Theory of Computing Systems, 2005]

CLIQUE COVER – Results



Task: Find few cliques
to cover all edges.

Workhorse | Data reduction rules prove quite effective although they only guarantee a kernel of size $O(2^k)$.

Interleaving | Combining search trees and data reduction can solve instances with clique cover sizes of about 150.

[Gramm et al., Proc. ALENEX 2006]

Part III
Epilogue

Conclusion

Fixed-Parameter Tractability belongs into the toolbox of algorithm designers - also in the area of graph-modeled data clustering.

Reduction rules and data reduction may be very effective despite of seemingly impractical worst-case bounds.

More experimental work and algorithm engineering needs to be done in cooperation with theorists. Be invited!

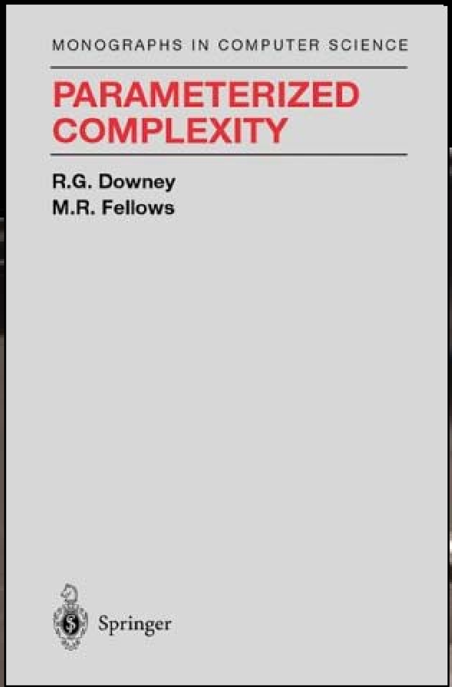
Further Reading

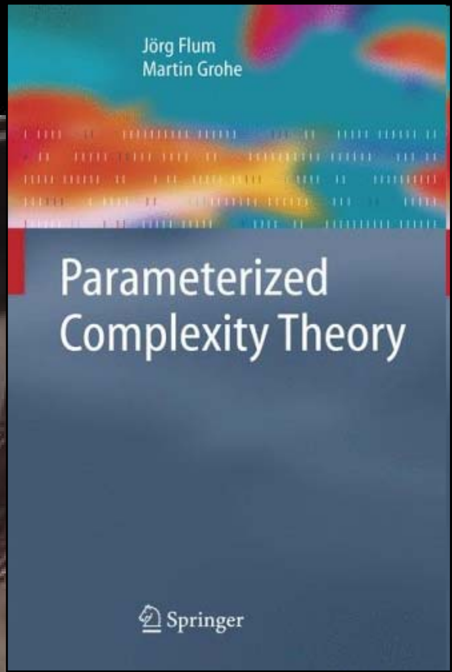
Further Reading

DOWNEY / FELLOWS: PARAMTERIZED COMPLEXITY (1999)

FIRST BOOK ON FPT

Further Reading





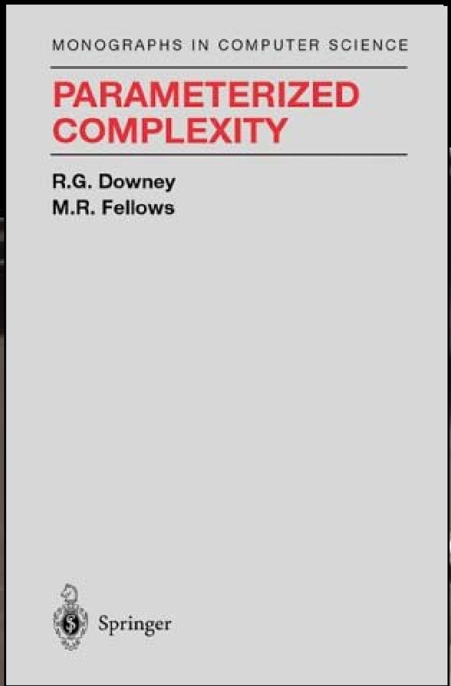
DOWNEY / FELLOWS:
PARAMTERIZED
COMPLEXITY (1999)

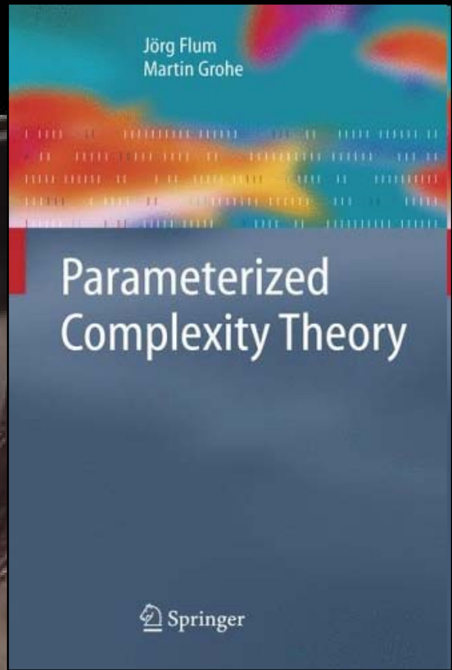
FIRST BOOK ON FPT

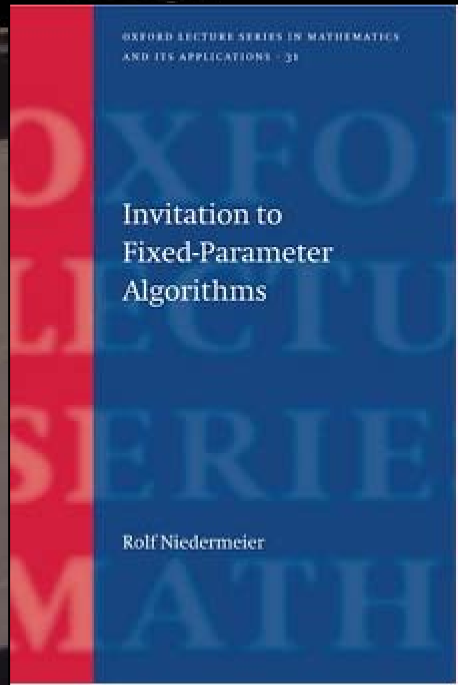
FLUM / GROHE:
PARAM. COMPLEXITY
THEORY (2006)

COVERS THEORY

Further Reading







DOWNEY / FELLOWS: PARAMTEREIZED COMPLEXITY (1999)

FIRST BOOK ON FPT

FLUM / GROHE: PARAM. COMPLEXITY THEORY (2006)

COVERS THEORY

NIEDERMEIER: INVITATION TO FIXED-PARAM. ALGOR. (2006)

COVERS ALGORITHMICS

Group Members Working on Fixed-Parameter Algorithms

Rolf Niedermeier
Chair

Michael Dom

Jiong Guo

Falk Hüffner

Hannes Moser

Sebastian Wernicke

Jochen Alber

Jens Gramm

http://theinf1.informatik.uni-jena.de/

Practical Fixed-Parameter Algorithms for Graph-Modeled Data Clustering

Sebastian Wernicke*

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Fed. Rep. of Germany
wernicke@minet.uni-jena.de

*Research supported by Deutsche Telekom Stiftung