NZIMA
Napier
2008

New Zealand Institute of Mathematics & its Applications
programme in algorithmics

Analysis of High-Throughput Biological Data Part I: Scalable High Performance Algorithms and Implementations

Mike Langston

Professor
Department of Electrical Engineering and Computer Science
University of Tennessee
and
Collaborating Scientist
Biological Sciences Division
Oak Ridge National Laboratory
USA

21 February 2008

NZIMA
Napier
2008

Outline of Talk

- Sample Application
- Tools and Technologies
  - Complexity Theory
  - Graph Algorithms
  - High Performance Computation
  - Reconfigurable Computation
- Compute Engine
- Problem Variants

NZIMA
Napier
2008

Outline of Talk

Sample Application

Tools and Technologies

- **Complexity Theory**
- **Graph Algorithms**
- **High Performance Computation**
- **Reconfigurable Computation**

Compute Engine

Problem Variants

NZIMA
Napier
2008

Technology Mapping

Analysis Tools

.
.

Ontology

.
.

***Cis*-Regulatory Elements**

.
.

Quantitative Trait Loci

.
.

Combinatorial Algorithms

.
.

Bayesian Networks

.
.

Biological Knowledge

.
.

Protein Structure

.
.

Gene Regulatory Networks

.
.

Sequence Homology

.
.

Protein function

.
.

Cell Physiology

.
.

NZIMA
Napier
2008

Technology Mapping

Analysis Tools

.
.

Ontology

.
.

***Cis*-Regulatory Elements**

.
.

Quantitative Trait Loci

.
.

Combinatorial Algorithms

.
.

Bayesian Networks

.
.

Biological Knowledge

.
.

Protein Structure

.
.

Gene Regulatory Networks

.
.

Sequence Homology

.
.

Protein function

.
.

Cell Physiology

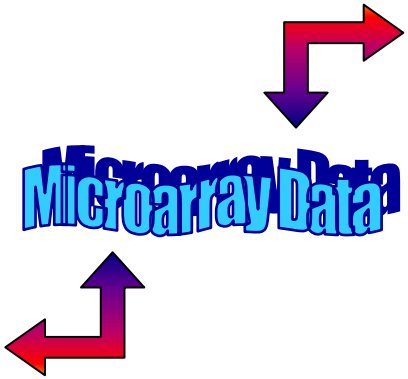
.
.

NZIMA
Napier
2008

Technology Mapping

Analysis Tools

- Ontology
- *Cis*-Regulatory Elements
- Quantitative Trait Loci
- Combinatorial Algorithms
- Bayesian Networks



Biological Knowledge

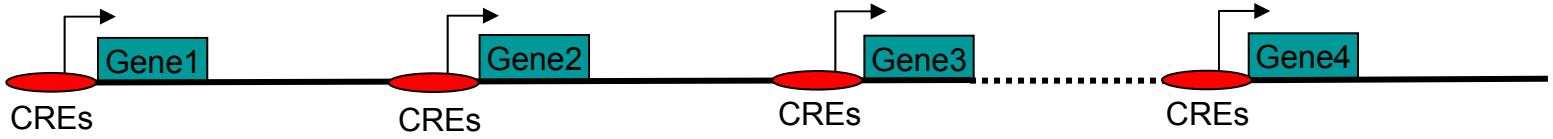
- Protein Structure
- Gene Regulatory Networks
- Sequence Homology
- Protein function
- Cell Physiology

NZIMA
Napier
2008

Gene Regulatory Networks

central dogma: one gene one protein

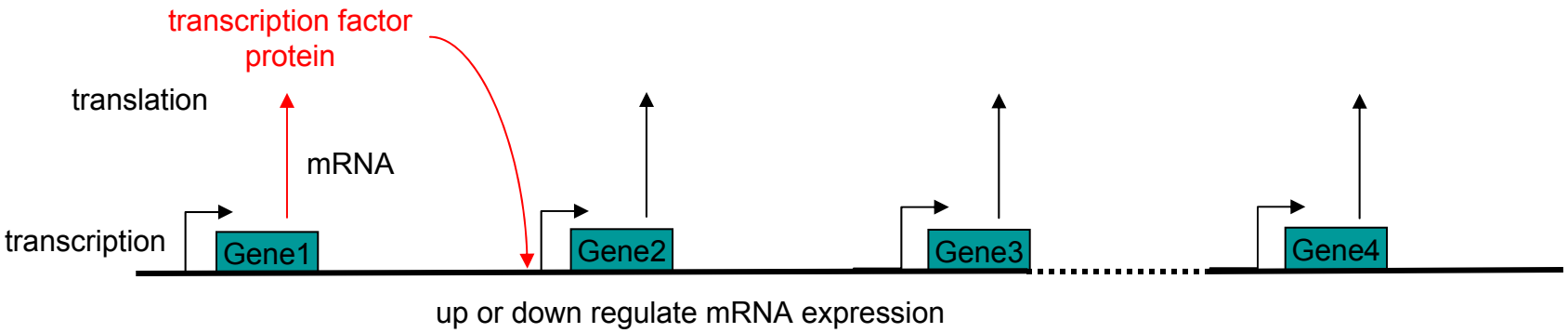
cis regulation



regulation via *cis* regulatory elements (CREs)
promoter, TATA box, motifs, modules
8-15 bp in length, action often at the ends

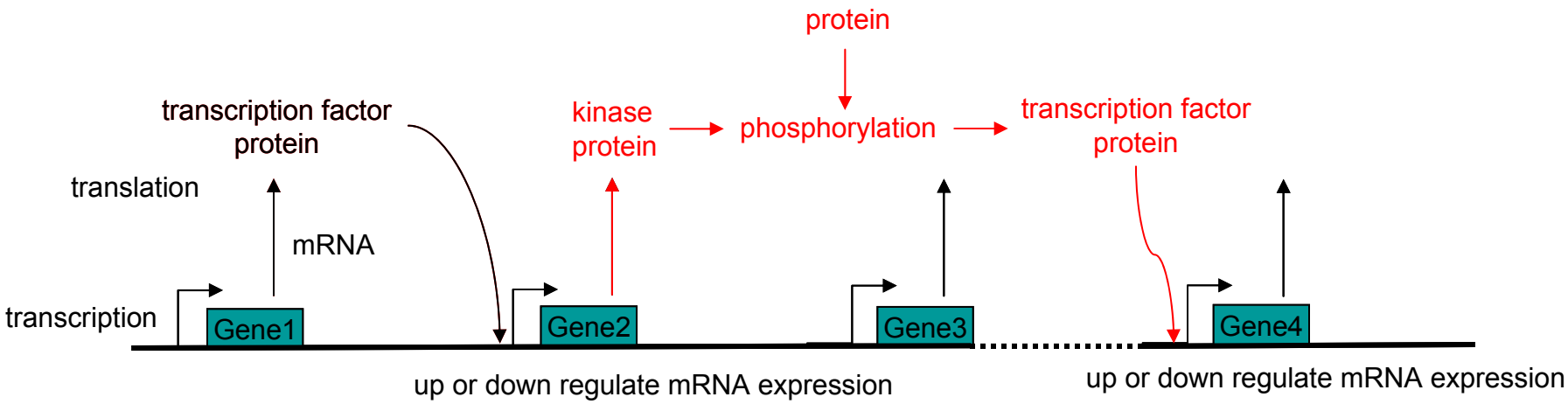
Gene Regulatory Networks

trans regulation (direct) via gene products

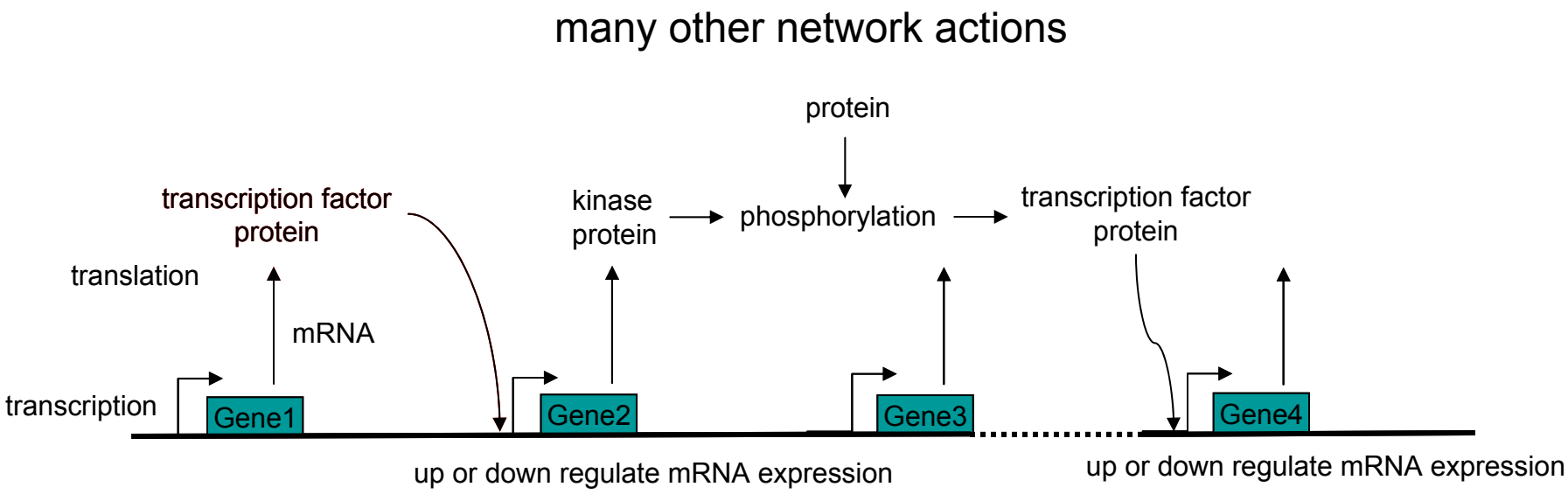


Gene Regulatory Networks

trans regulation (indirect) via post-translational modification



Gene Regulatory Networks



post-transcriptional regulation (e.g., alternate splicing)

μRNA (e.g., functional RNA, RNAi and gene silencing)

but all are forms of *co-regulation*

NZIMA
Napier
2008

Currently Awash in a Sea of Transcriptomic Data

An organism's mRNA transcripts:

- link between the genome, the proteome and the cellular phenotype
- data quality and richness increasing
  - noise reduction
  - more conditions
  - correlation, putative coregulation, regulatory networks
- cannot see post-translational modifications (e.g., phosphorylation)
- huge range of prokaryotic and eukaryotic data coming on line
- timely confluence of technologies
- proteomics, metabolomics data not far behind

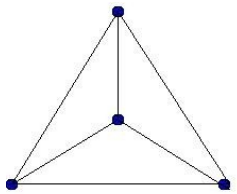
NZIMA
Napier
2008

A Major Computational Bottleneck: Clique

Data transformation:

- representing biological networks with graphs is well understood
- genes (via transcripts, probesets) are denoted by vertices
- edges denote significant gene-gene correlations
- we seek genesets with common regulatory mechanisms
- thus we want to identify dense subgraphs, in particular *cliques*
  - complete subgraphs
  - special case of subgraph isomorphism
  - *NP*-complete to decide
  - *NP*-complete even to approximate

K_4



NZIMA
Napier
2008

Outline of Talk

Sample Application

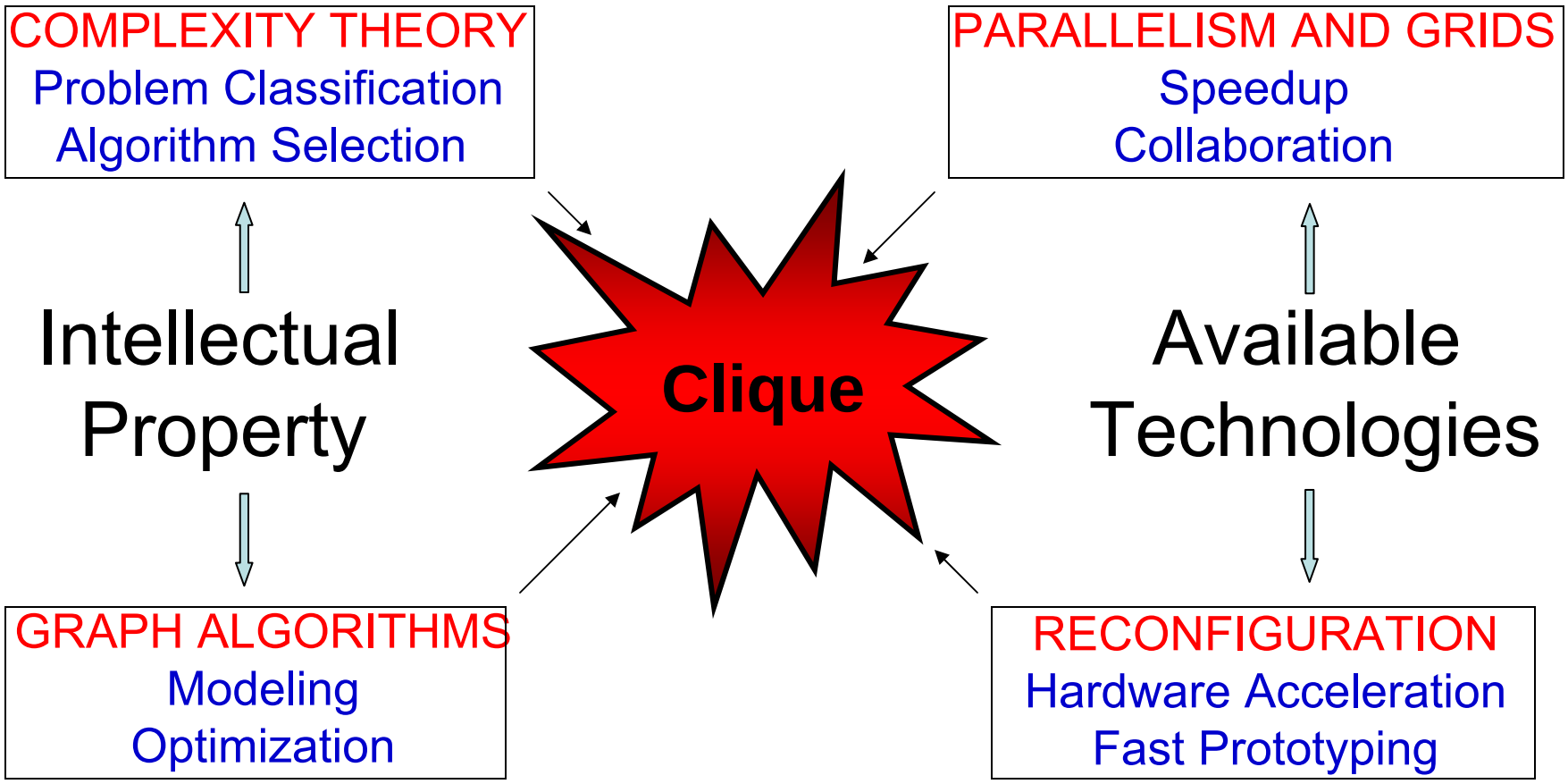
Tools and Technologies

- **Complexity Theory**
- **Graph Algorithms**
- **High Performance Computation**
- **Reconfigurable Computation**

Compute Engine

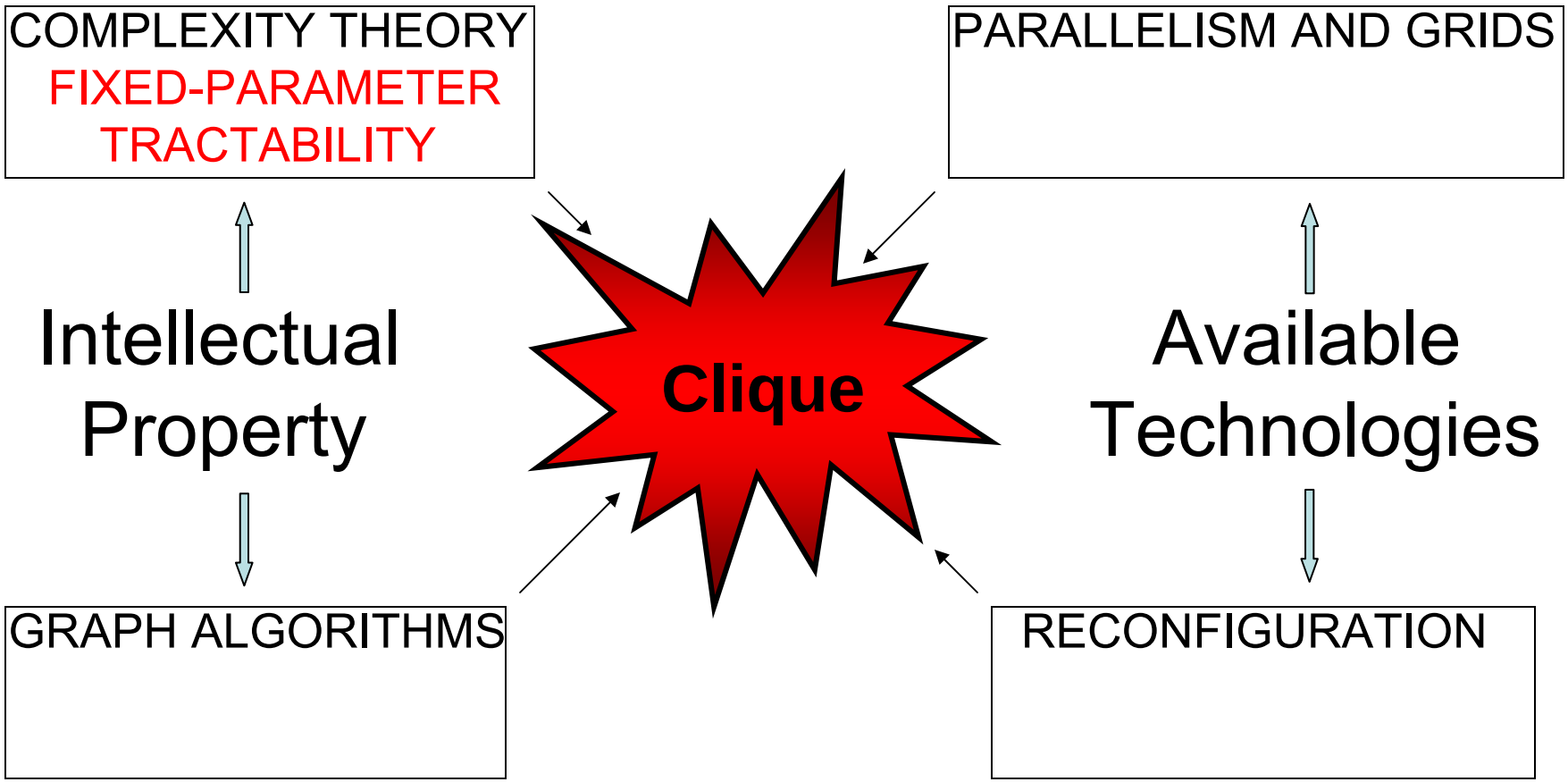
Problem Variants

Tools and Technologies



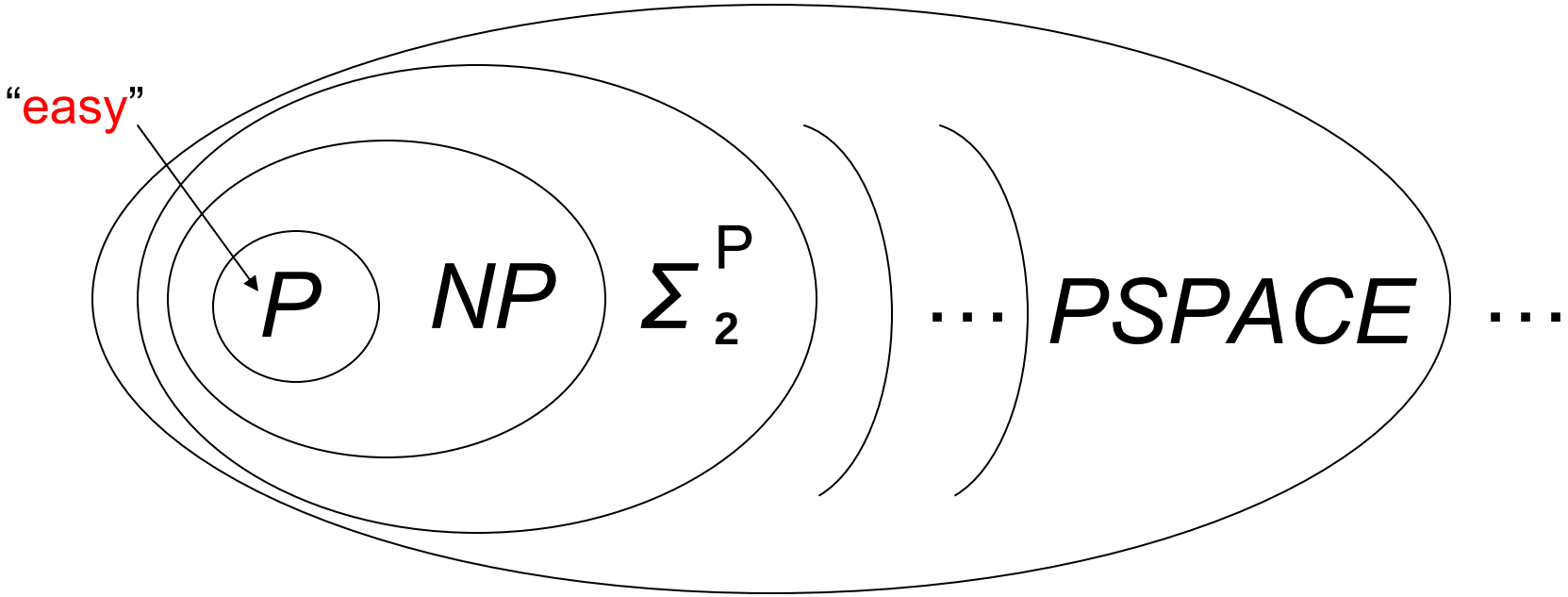
NZIMA
Napier
2008

Tools and Technologies



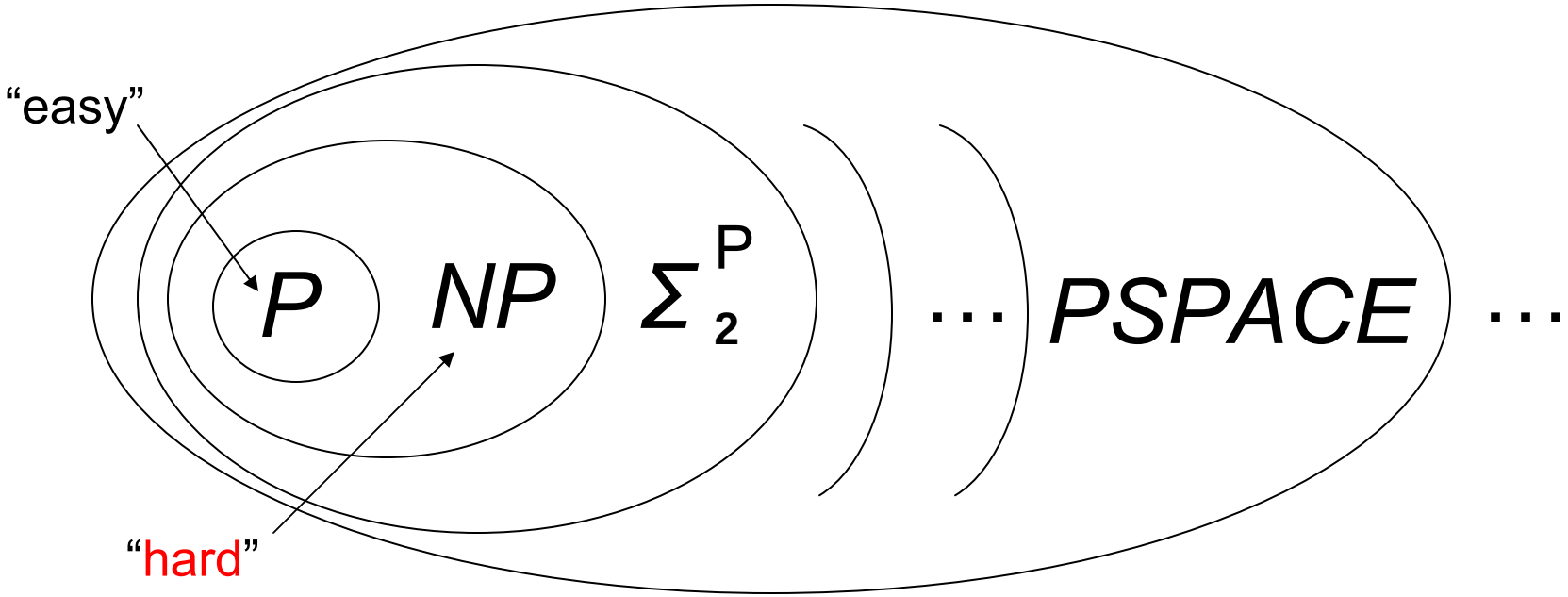
A Little Complexity Theory

The Classic View:



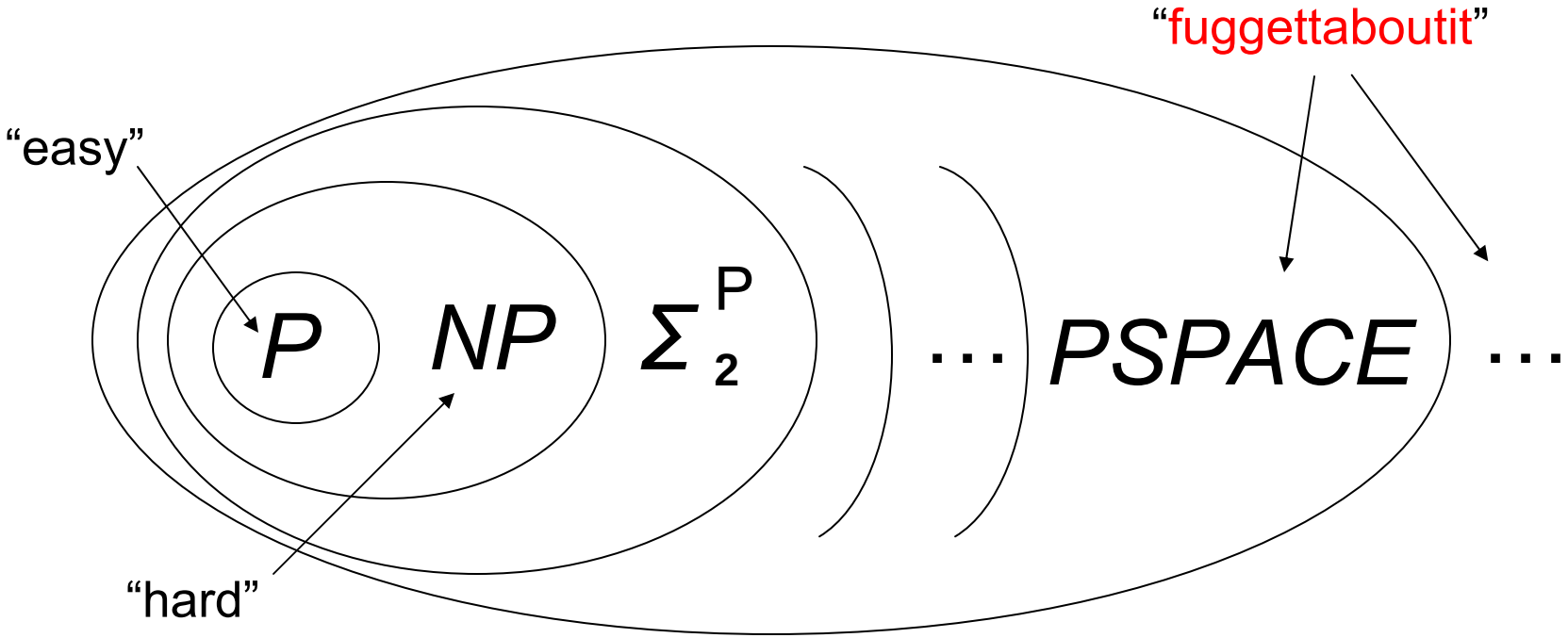
A Little Complexity Theory

The Classic View:



A Little Complexity Theory

The Classic View:



NZIMA
Napier
2008

Fixed-Parameter Tractability

Pioneering approach going back twenty years

- Well-Quasi-Order theory
- nonuniform measure of complexity

NZIMA
Napier
2008

Fixed-Parameter Tractability

Pioneering approach going back twenty years

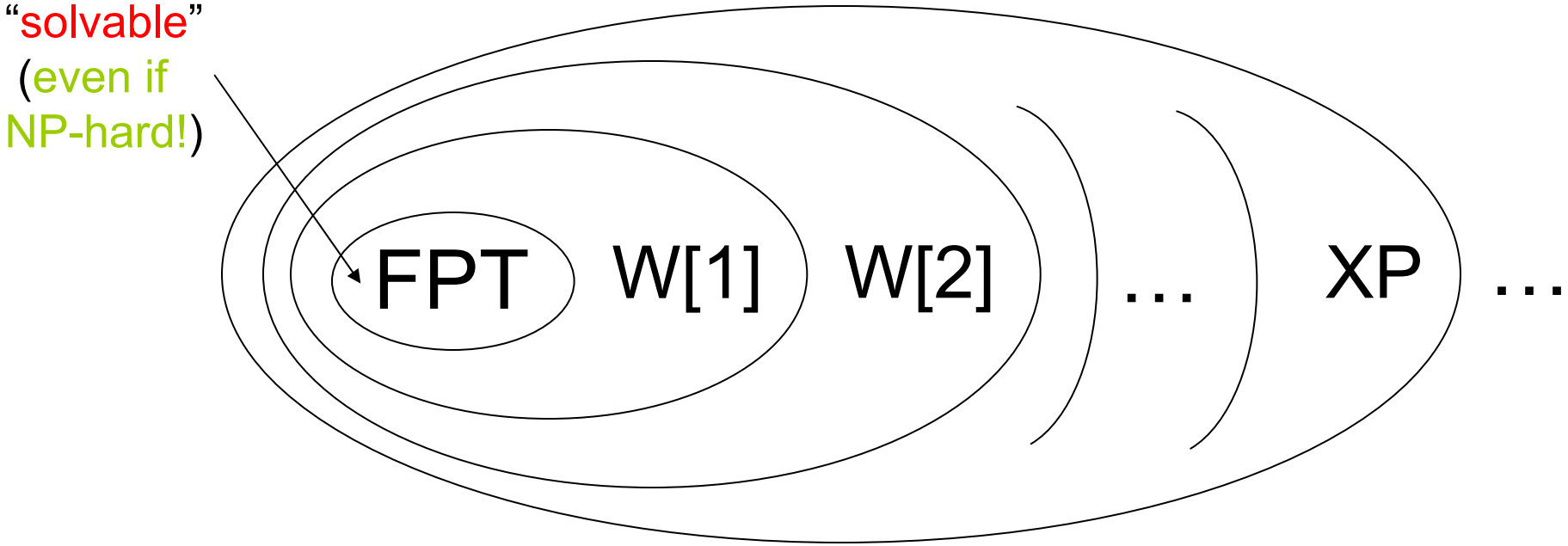
- Well-Quasi-Order theory
- nonuniform measure of complexity

Exploit knowledge of the solution space

- Consider an algorithm with a time bound such as $O(2^{kn})$.
- And now one with a time bound more like $O(2^k n)$.
- Both are exponential in parameter value(s).
- But what happens when k is fixed?
- Fixed-Parameter Tractable (FPT) iff $O(f(k)n^c)$
- Confines superpolynomial behavior to the parameter

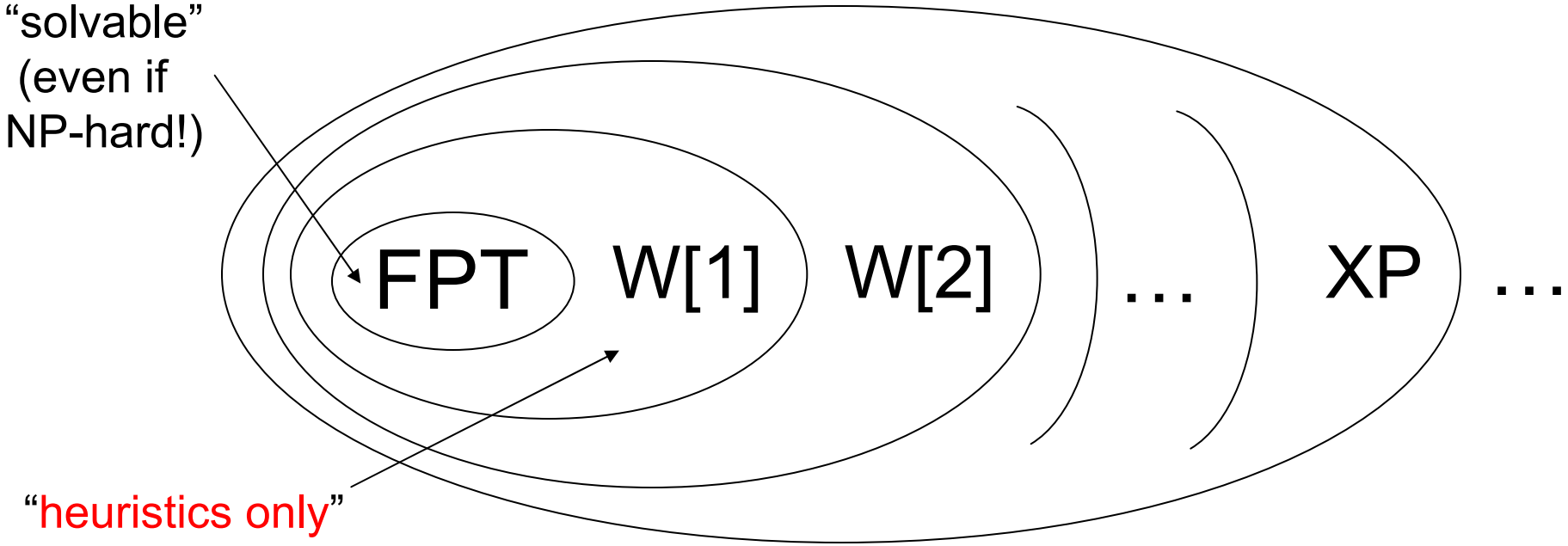
Complexity Theory, Refined

Hence, the Parameterized View:



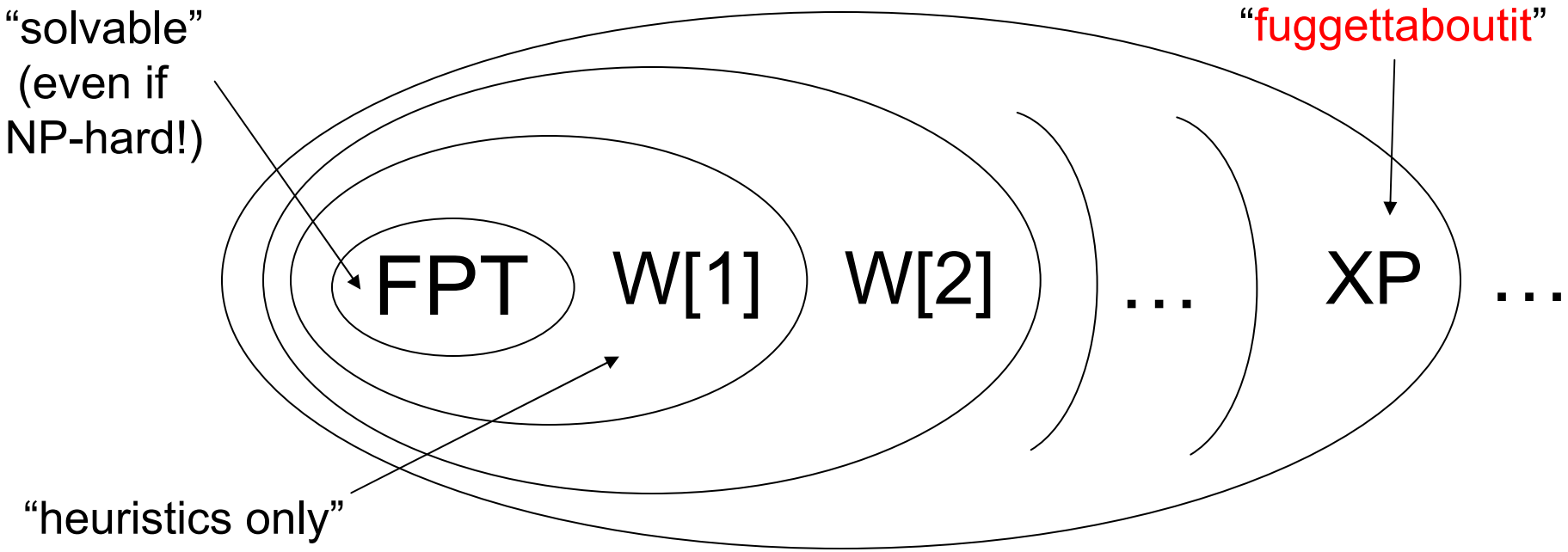
Complexity Theory, Refined

The Parameterized View:

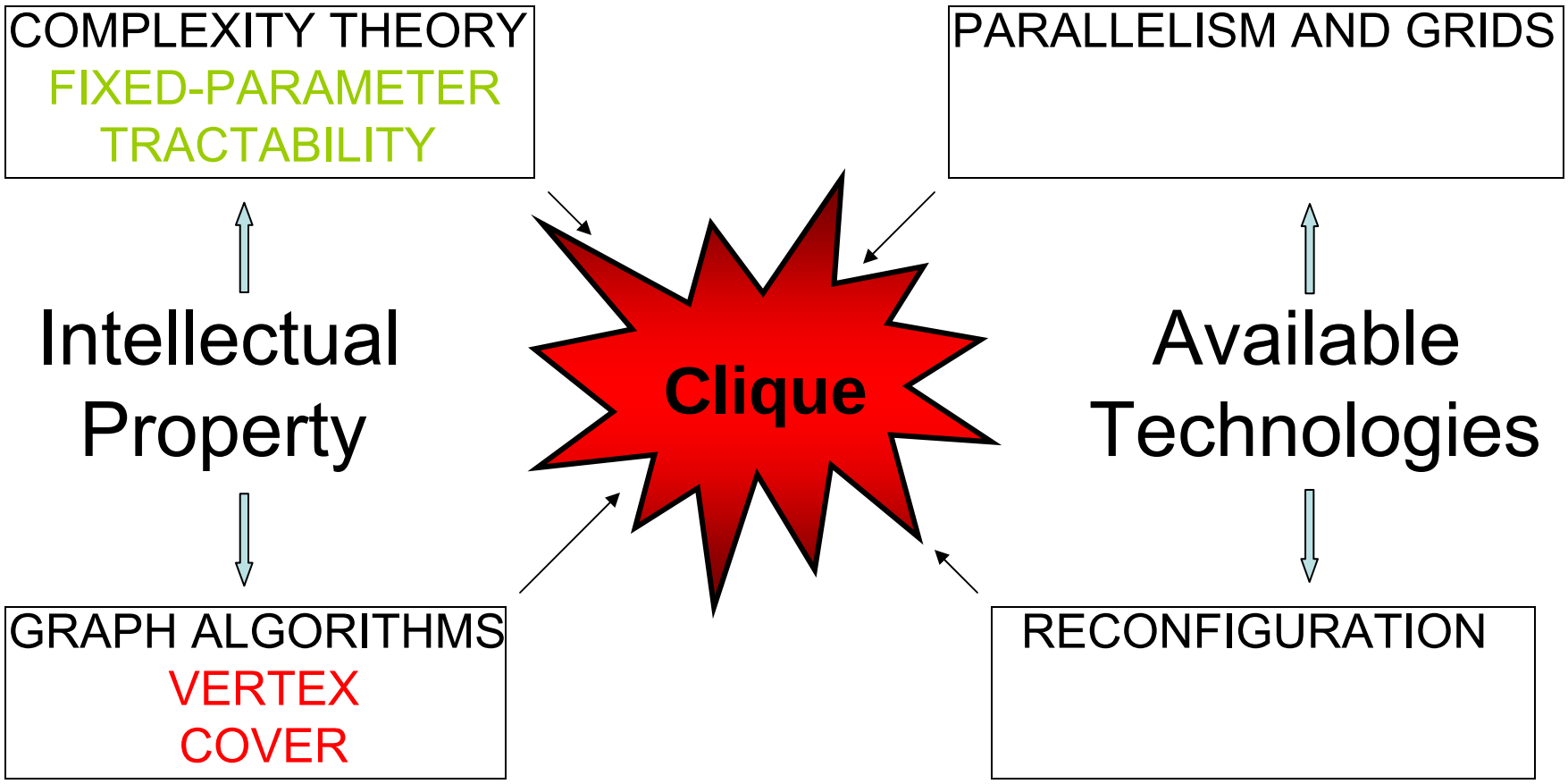


Complexity Theory, Refined

The Parameterized View:



Tools and Technologies



NZIMA
Napier
2008

The Vertex Cover Project

Pioneering approach going back twenty years

- Well-Quasi-Order theory
- nonuniform measure of complexity

Exploit knowledge of the solution space

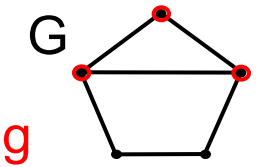
- Consider an algorithm with a time bound such as $O(2^{kn})$.
- And now one with a time bound more like $O(2^k n)$.
- Both are exponential in parameter value(s).
- But what happens when k is fixed?
- Fixed-Parameter Tractable (FPT) iff $O(f(k)n^c)$
- Confines superpolynomial behavior to the parameter

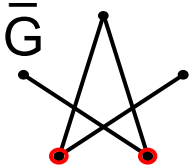
Duality

- We solve **vertex cover**, clique's complementary dual
- $O(1.2759^k k^{1.5}+kn)$ time

Key features

- Kernelization, branching and interleaving





The Vertex Cover Project

- use preprocessing via degree structures
  - Low degree rules
  - High degree rule
  - Resultant graph has size $O(k^2)$
    [at most k(1+k/3) vertices]

NZIMA
Napier
2008

The Vertex Cover Project

- use preprocessing via degree structures
- then kernelize to reduce to a computational core
  - suite of codes

NZIMA
Napier
2008

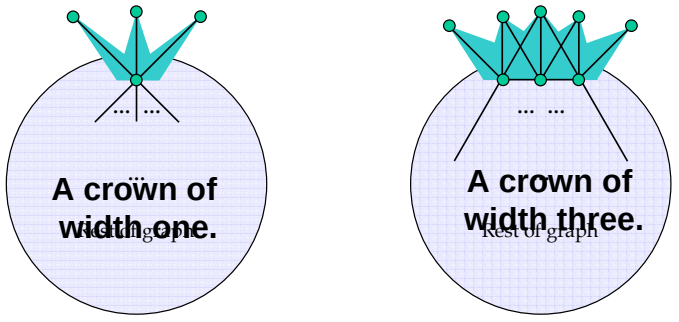
The Vertex Cover Project

- use preprocessing via degree structures
- then kernelize to reduce to a computational core
  - suite of codes
  - LP variants

minimize: ΣX_i, i in $V(G)$
subject to: $X_u + X_v >= 1$ for all uv in $E(G)$
where: $X_i >= 0$ for all i in $V(G)$

The Vertex Cover Project

- use preprocessing via degree structures
- then kernelize to reduce to a computational core
  - suite of codes
  - LP variants
  - crown rule



A crown of width 1 A crown of width 3

NZIMA
Napier
2008

Representative Kernelization Results

Algorithm	Run Time	Kernel (n')	Parameter (k')
LP	69.49	616	389
Network Flow	40.53	622	392
Crown Rule	0.07	630	392

Preprocessing completed first. All times in seconds.

NZIMA
Napier
2008

Representative Kernelization Results

Algorithm	Run Time	Kernel (n')	Parameter (k')
LP	69.49	616	389
Network Flow	40.53	622	392
Crown Rule	0.07	630	392

Preprocessing completed first. All times in seconds.

Some conclusions:

- Perform preprocessing, then the crown rule.
- If dense, stop trying to kernelize.
- If sparse, try LP or network flow before stopping.

NZIMA
Napier
2008

The Vertex Cover Project

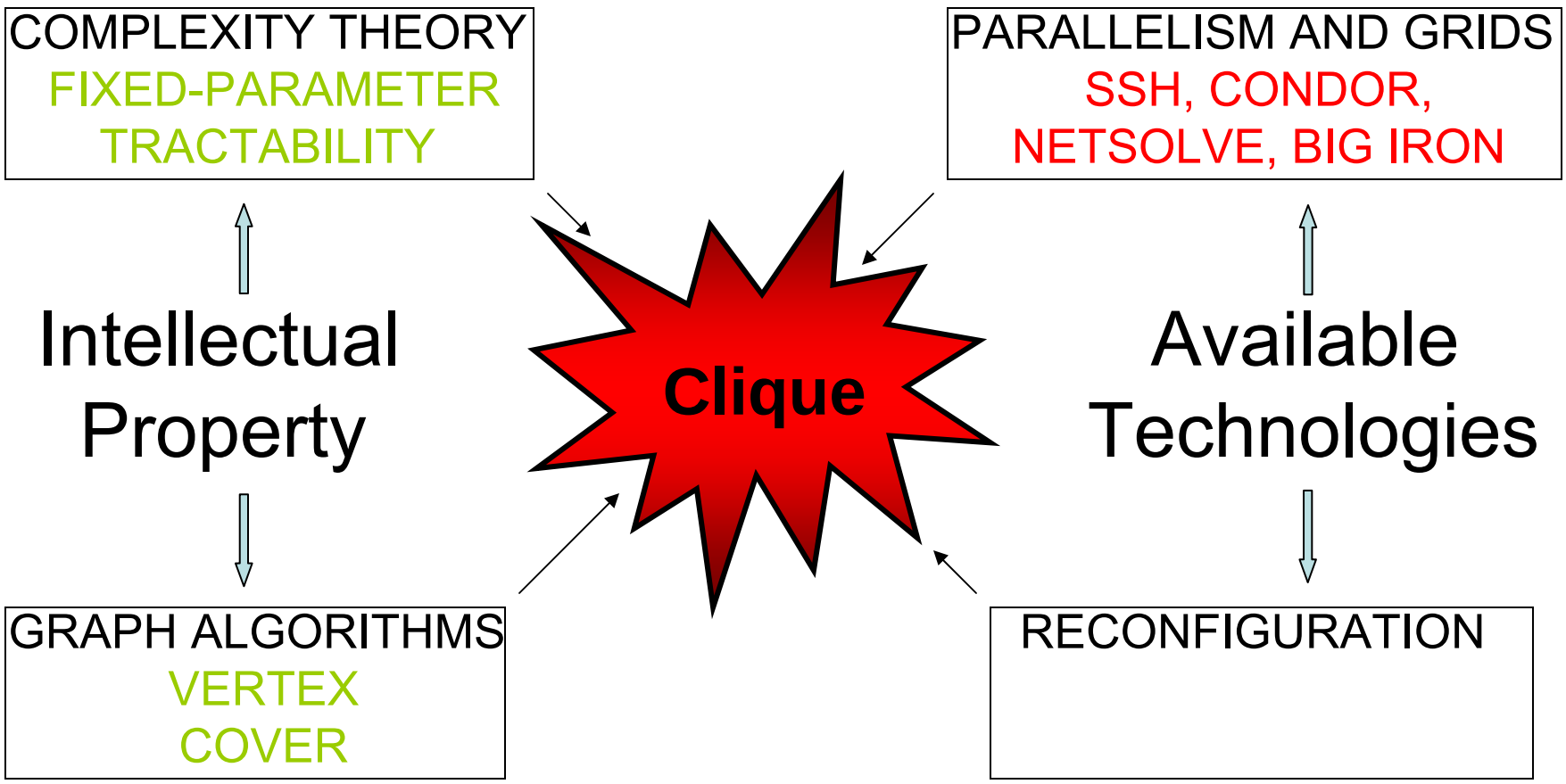
- use preprocessing via degree structures
- then kernelize to reduce to a computational core
- employ branching to explore the core
  - exhaustive search
  - highly parallel

NZIMA
Napier
2008

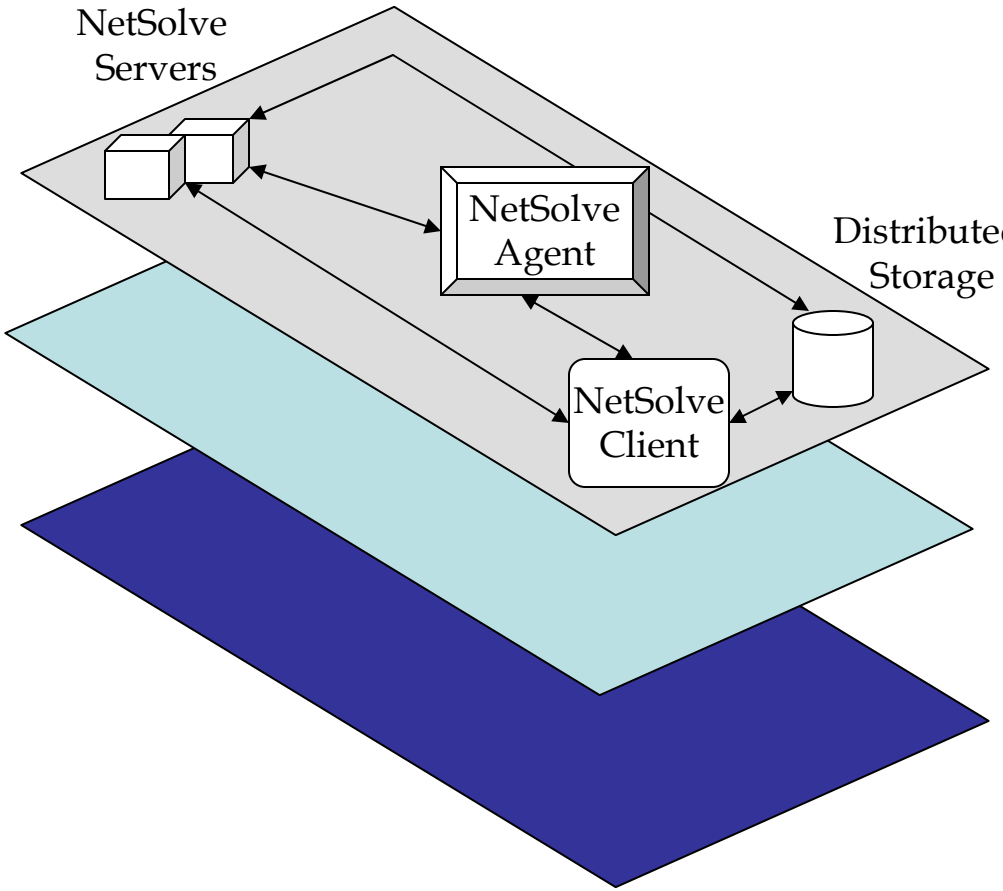
The Vertex Cover Project

- use preprocessing via degree structures
- then kernelize to reduce to a computational core
- employ branching to explore the core
- finally, interleave all three

Tools and Technologies



Sample Grid Architecture



Key: NetSolve's program description file facility

Middleware
(NetSolve)

Compute Resources
(Grid Service Clusters)

Foundational Fabric
(Switches and Depots)

NZIMA
Napier
2008

Supercomputer Platforms

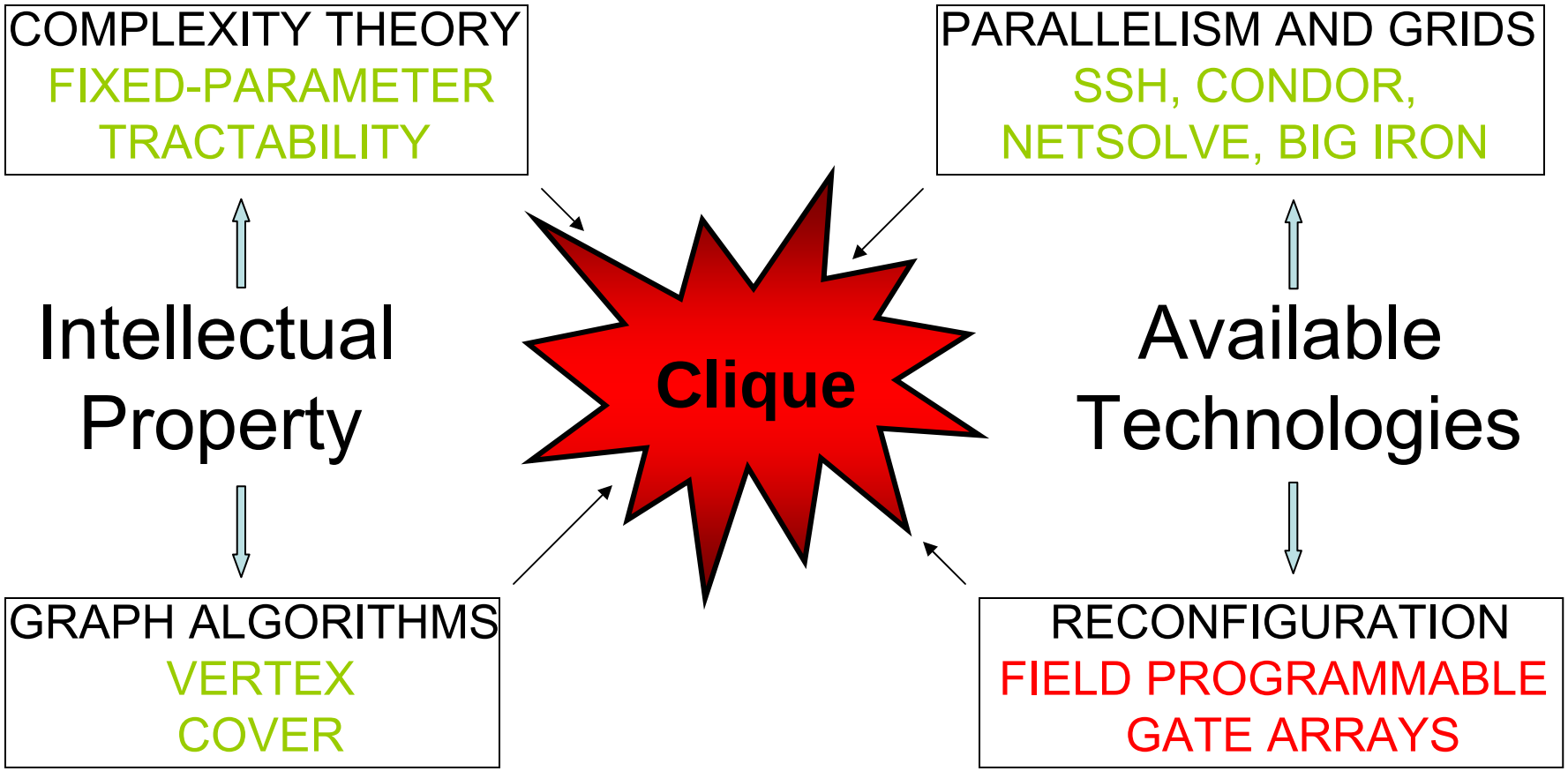
High Performance Implementations

- **suites of maximum/maximal/bi/para clique methods**
- **have processed graphs with over 3M vertices**
- **memory often a limiting factor**
- **currently working on out-of-core methods**

SGI Altix supercomputer at ORNL
256 dual-CPU processors, two terabytes of shared memory

NZIMA
Napier
2008

Tools and Technologies



NZIMA
Napier
2008

Hardware Acceleration

Algorithms are very different.

VHDL versus C.

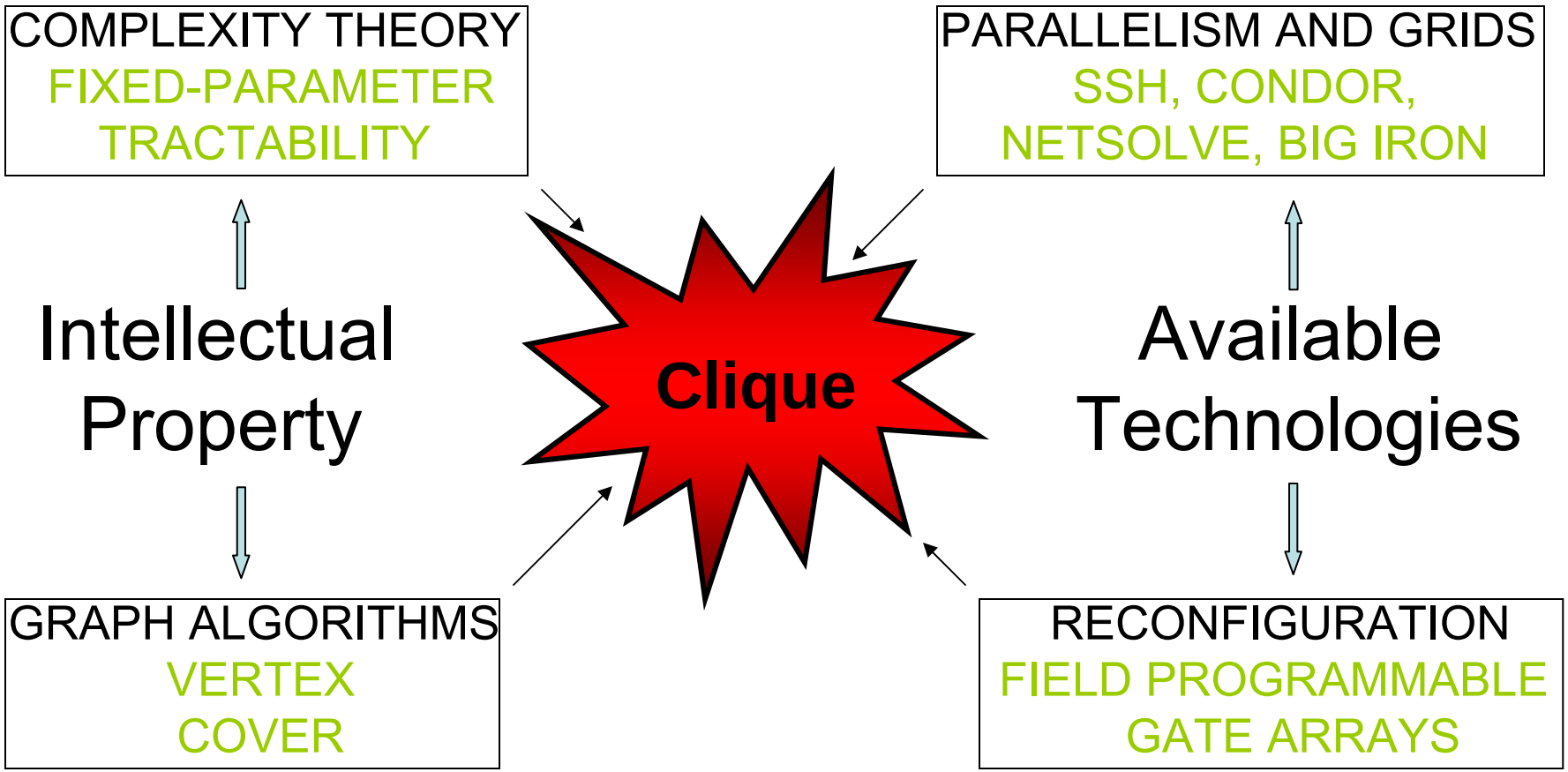
I/O is often the most critical resource.

Sample FPGA

With current implementations, we are able to solve sub-instances:

- of size 512 or less,
- with speedups north of about 125.

Put the Pieces Together



NZIMA
Napier
2008

Outline of Talk

Sample Application

Tools and Technologies

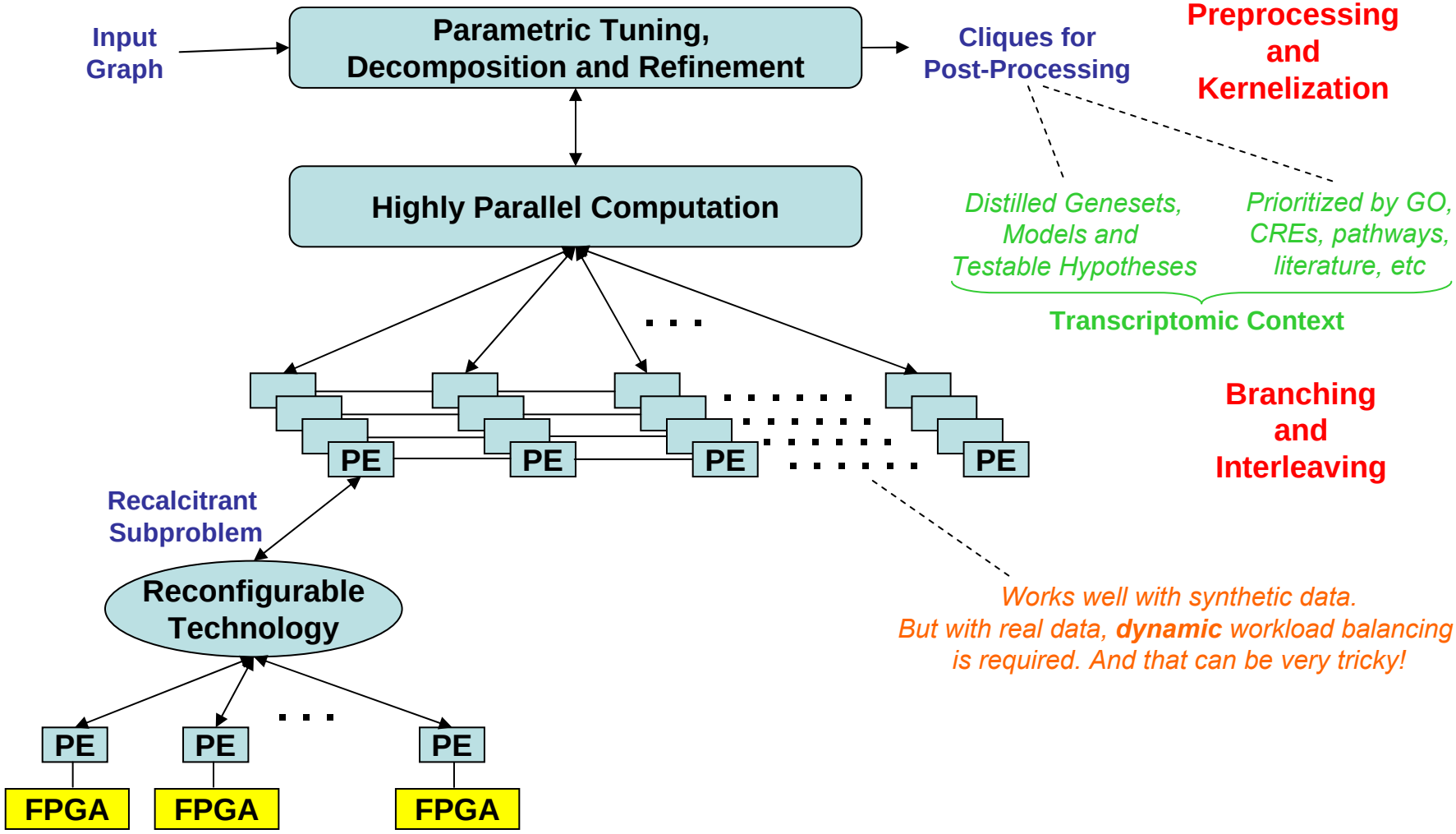
- **Complexity Theory**
- **Graph Algorithms**
- **High Performance Computation**
- **Reconfigurable Computation**

Compute Engine

Problem Variants

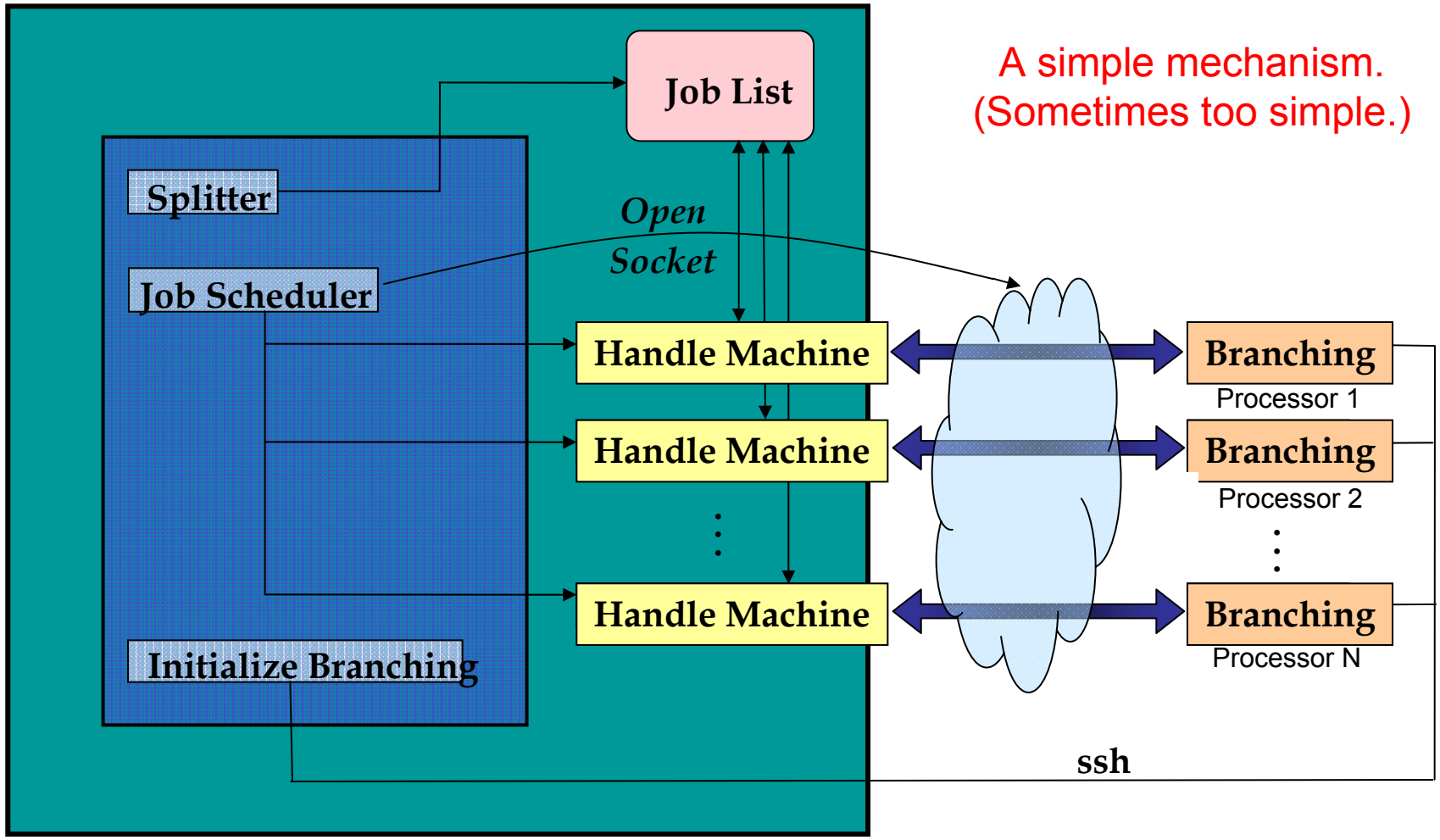
NZIMA
Napier
2008

A Clique Compute Engine



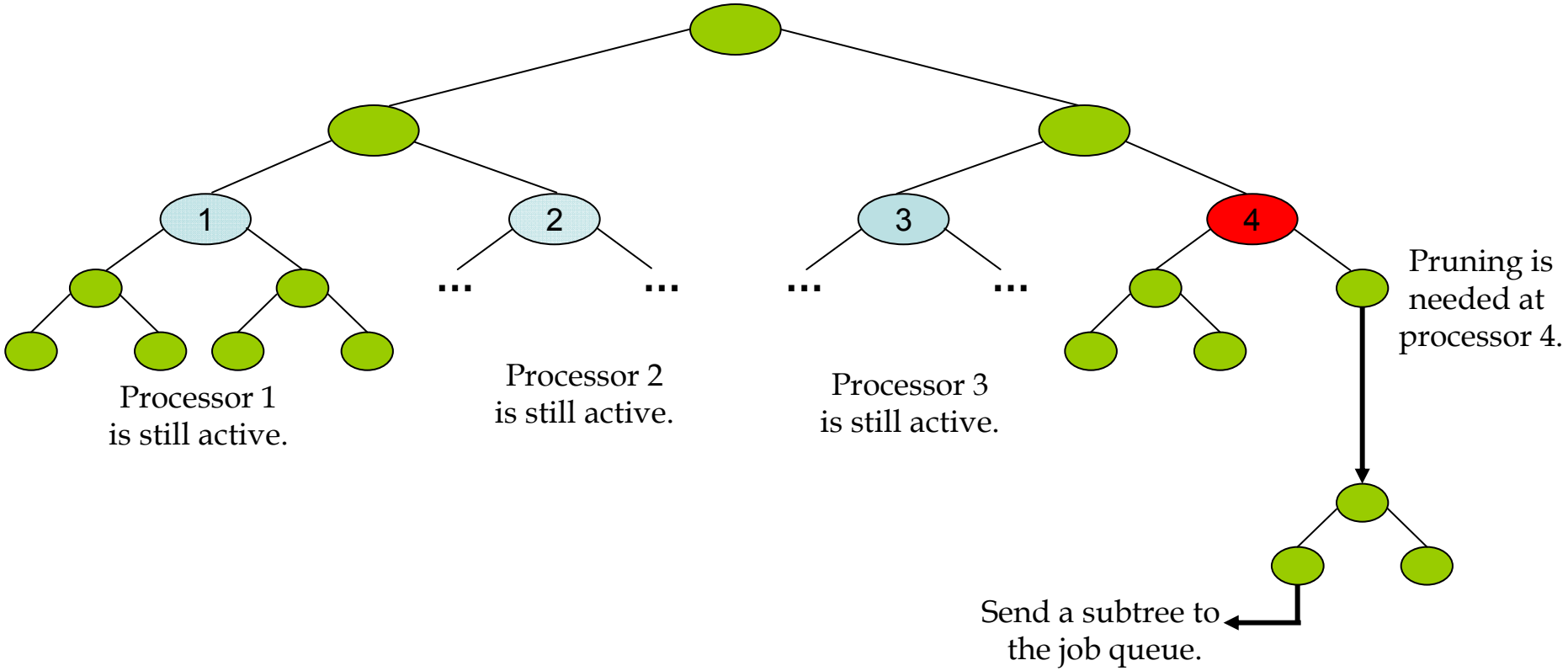
NZIMA
Napier
2008

Workload Balancing: A Vertex Cover Driver



A simple mechanism.
(Sometimes too simple.)

Workload Balancing: Distributed Subtree Splitting



NZIMA
Napier
2008

Sample Results on Protein Sequence Data

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Kernelization	Sequential Branching	Parallel Branching	Dynamic Decomposition
SH2-5	839	399	Yes	34 seconds	7 seconds	Not needed	Not needed
SH2-5	839	398	No	34 seconds	141 minutes	82 minutes	20 minutes
SH3-10	2466	2044					
SH3-10	2466	2043					

So clique size is 422.

A direct assault ~ 2466^{422}.

NZIMA
Napier
2008

Sample Results on Protein Sequence Data

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Kernelization	Sequential Branching	Parallel Branching	Dynamic Decomposition
SH2-5	839	399	Yes	34 seconds	7 seconds	Not needed	Not needed
SH2-5	839	398	No	34 seconds	141 minutes	82 minutes	20 minutes
SH3-10	2466	2044	Yes				
SH3-10	2466	2043	No				

So clique size is 422.

The hardest computations.

NZIMA
Napier
2008

Sample Results on Protein Sequence Data

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Kernelization	Sequential Branching	Parallel Branching	Dynamic Decomposition
SH2-5	839	399	Yes	34 seconds	7 seconds	Not needed	Not needed
SH2-5	839	398	No	34 seconds	141 minutes	82 minutes	20 minutes
SH3-10	2466	2044	Yes	203 minutes			
SH3-10	2466	2043	No	203 minutes			

So clique size is 422.

The hardest computations.

NZIMA
Napier
2008

Sample Results on Protein Sequence Data

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Kernelization	Sequential Branching	Parallel Branching	Dynamic Decomposition
SH2-5	839	399	Yes	34 seconds	7 seconds	Not needed	Not needed
SH2-5	839	398	No	34 seconds	141 minutes	82 minutes	20 minutes
SH3-10	2466	2044	Yes	203 minutes	~ 5 days		
SH3-10	2466	2043	No	203 minutes	6+ days		

So clique size is 422.

The hardest computations.

NZIMA
Napier
2008

Sample Results on Protein Sequence Data

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Kernelization	Sequential Branching	Parallel Branching	Dynamic Decomposition
SH2-5	839	399	Yes	34 seconds	7 seconds	Not needed	Not needed
SH2-5	839	398	No	34 seconds	141 minutes	82 minutes	20 minutes
SH3-10	2466	2044	Yes	203 minutes	~ 5 days	~ 5 days	
SH3-10	2466	2043	No	203 minutes	6+ days	6+ days	

So clique size is 422.

The hardest computations.

32 PEs @ 500MHz.

NZIMA
Napier
2008

Sample Results on Protein Sequence Data

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Kernelization	Sequential Branching	Parallel Branching	Dynamic Decomposition
SH2-5	839	399	Yes	34 seconds	7 seconds	Not needed	Not needed
SH2-5	839	398	No	34 seconds	141 minutes	82 minutes	20 minutes
SH3-10	2466	2044	Yes	203 minutes	~ 5 days	~ 5 days	140 minutes
SH3-10	2466	2043	No	203 minutes	6+ days	6+ days	620 minutes

So clique size is 422.

The hardest computations.

32 PEs @ 500MHz.

Load balancing is critical.
"No" is harder than "yes."

NZIMA
Napier
2008

Sample Results on Protein Sequence Data

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Kernelization	Sequential Branching	Parallel Branching	Dynamic Decomposition
SH2-5	839	399	Yes	34 seconds	7 seconds	Not needed	Not needed
SH2-5	839	398	No	34 seconds	141 minutes	82 minutes	20 minutes
SH3-10	2466	2044	Yes	203 minutes	~ 5 days	~ 5 days	140 minutes
SH3-10	2466	2043	No	203 minutes	6+ days	6+ days	620 minutes

We now routinely solve these sorts of instances in seconds. But these are not genome scale problems!

NZIMA
Napier
2008

A Bit of Blasphemy: The Real Power of FPT

Guides our thinking, steering us to exploit parameters

NZIMA
Napier
2008

A Bit of Blasphemy: The Real Power of FPT

Guides our thinking, steering us to exploit parameters

Kernelization sets the stage for efficiency

NZIMA
Napier
2008

A Bit of Blasphemy: The Real Power of FPT

Guides our thinking, steering us to exploit parameters

Kernelization sets the stage for efficiency

Branching still requires serious computation

NZIMA
Napier
2008

A Bit of Blasphemy: The Real Power of FPT

Guides our thinking, steering us to exploit parameters

Kernelization sets the stage for efficiency

Branching still requires serious computation

Interleaving is indispensible in practice

NZIMA
Napier
2008

A Bit of Blasphemy: The Real Power of FPT

Guides our thinking, steering us to exploit parameters

Kernelization sets the stage for efficiency

Branching still requires serious computation

Interleaving is indispensible in practice

Solve problems directly (clique not vertex cover)

NZIMA
Napier
2008

A Bit of Blasphemy: The Real Power of FPT

Guides our thinking, steering us to exploit parameters

Kernelization sets the stage for efficiency

Branching still requires serious computation

Interleaving is indispensible in practice

Solve problems directly (clique not vertex cover)

Better subtree pruning via iterated preprocessing

NZIMA
Napier
2008

A Bit of Blasphemy: The Real Power of FPT

Guides our thinking, steering us to exploit parameters

Kernelization sets the stage for efficiency

Branching still requires serious computation

Interleaving is indispensible in practice

Solve problems directly (clique not vertex cover)

Better subtree pruning via iterated preprocessing

Examples: Common Neighbor Preprocessing (CNP)

Color Preprocessing (CP)

A Bit of Blasphemy: The Real Power of FPT

Representative Computational Results

Common Neighbor Preprocessing versus Color Preprocessing

Method	Kernel Size		Time		
CNP					
CP					
CNP+CP					

Data Source: Gerling Affymetrix 430A
read time 0:40, probe sets 22690, threshold edges 7,534,598, maximum clique size 248

A Bit of Blasphemy: The Real Power of FPT

Representative Computational Results

Common Neighbor Preprocessing versus Color Preprocessing

Method	Kernel Size		Time		
	Vertices	Edges			
CNP	5896	2785k			
CP	1700	585k			
CNP+CP	1692	576k			

Data Source: Gerling Affymetrix 430A
read time 0:40, probe sets 22690, threshold edges 7,534,598, maximum clique size 248

A Bit of Blasphemy: The Real Power of FPT

Representative Computational Results

Common Neighbor Preprocessing versus Color Preprocessing

Method	Kernel Size		Time		
	Vertices	Edges	Preprocess		
CNP	5896	2785k	2:24		
CP	1700	585k	1:22		
CNP+CP	1692	576k	3:46		

Data Source: Gerling Affymetrix 430A
read time 0:40, probe sets 22690, threshold edges 7,534,598, maximum clique size 248

A Bit of Blasphemy: The Real Power of FPT

Representative Computational Results

Common Neighbor Preprocessing versus Color Preprocessing

Method	Kernel Size		Time		
	Vertices	Edges	Preprocess	Branch	
CNP	5896	2785k	2:24	51:54	
CP	1700	585k	1:22	4:04	
CNP+CP	1692	576k	3:46	3:46	

Data Source: Gerling Affymetrix 430A
read time 0:40, probe sets 22690, threshold edges 7,534,598, maximum clique size 248

A Bit of Blasphemy: The Real Power of FPT

Representative Computational Results

Common Neighbor Preprocessing versus Color Preprocessing

Method	Kernel Size		Time		
	Vertices	Edges	Preprocess	Branch	Total
CNP	5896	2785k	2:24	51:54	54:18
CP	1700	585k	1:22	4:04	5:26
CNP+CP	1692	576k	3:46	3:46	6:58

Data Source: Gerling Affymetrix 430A
read time 0:40, probe sets 22690, threshold edges 7,534,598, maximum clique size 248

NZIMA
Napier
2008

Outline of Talk

Sample Application

Tools and Technologies

- **Complexity Theory**
- **Graph Algorithms**
- **High Performance Computation**
- **Reconfigurable Computation**

Compute Engine

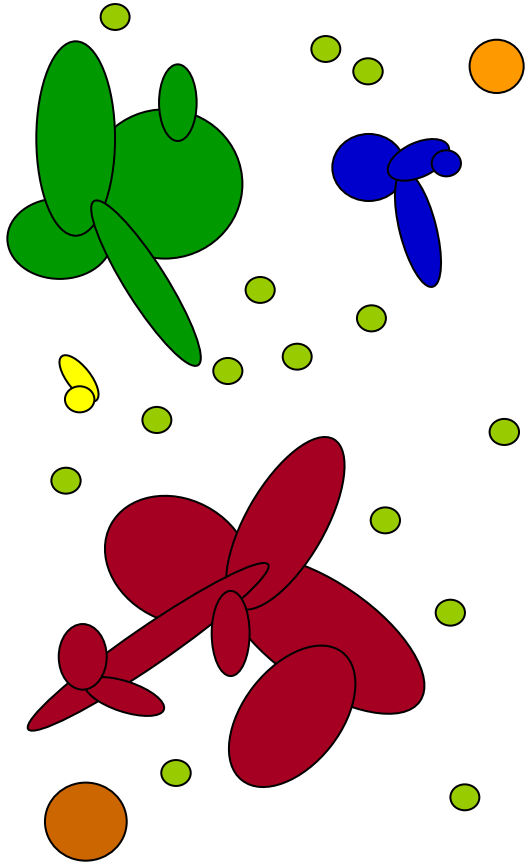
Problem Variants

NZIMA
Napier
2008

Maximal Clique

Biological Fidelity

- Genes are Pleiotropic
- Maximal Cliques May Overlap



NZIMA
Napier
2008

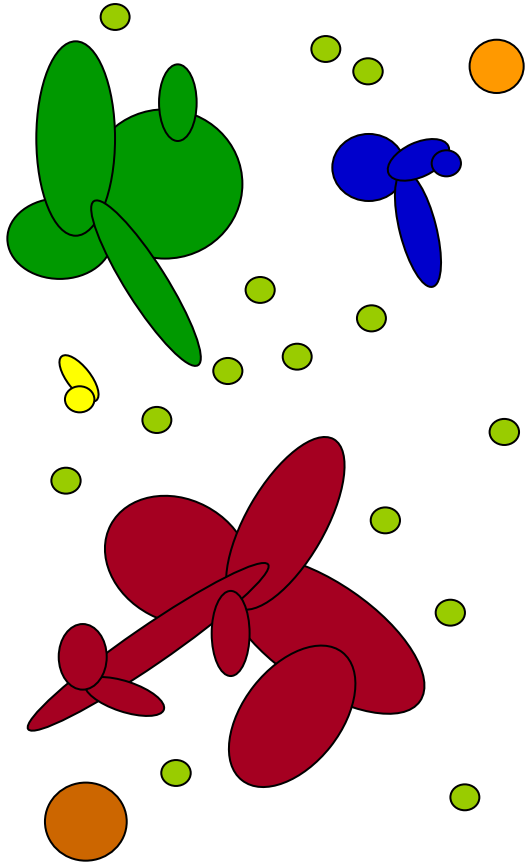
Maximal Clique

Biological Fidelity

- Genes are Pleiotropic
- Maximal Cliques May Overlap

Results

- Efficiency
- Predictable Range of Outputs



NZIMA
Napier
2008

Maximal Clique

Biological Fidelity

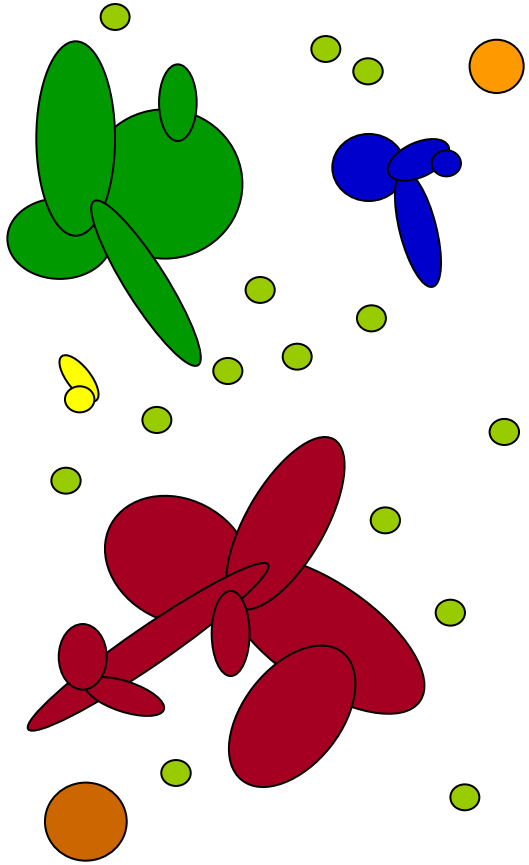
- Genes are Pleiotropic
- Maximal Cliques May Overlap

Results

- Efficiency
- Predictable Range of Outputs

Keys

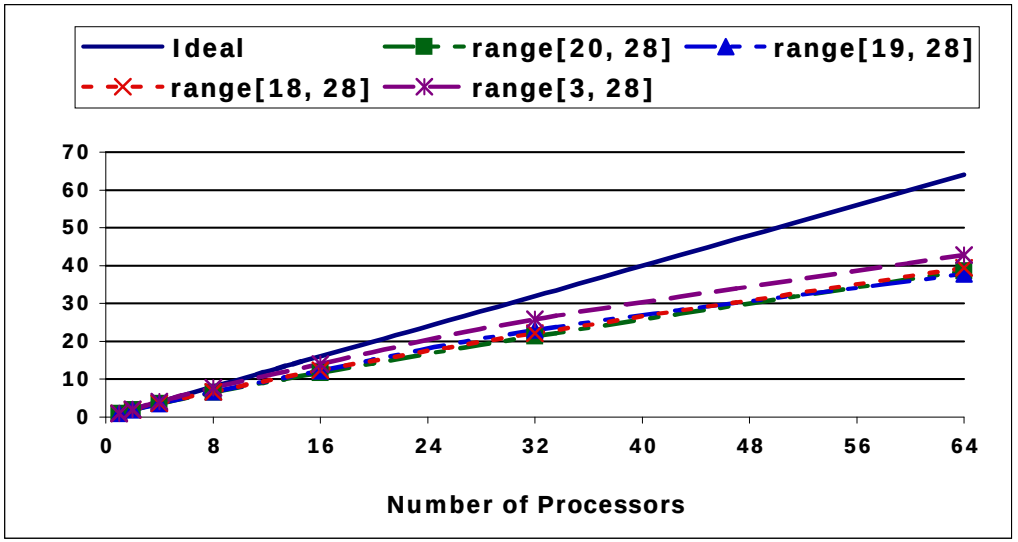
- Global Shared Memory Map
- Bitmapped Implementations
- Synchronization and Load Balancing



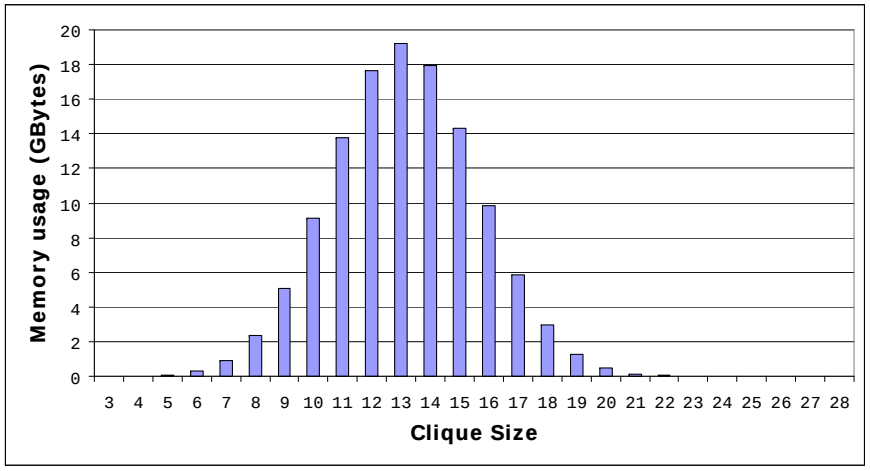
NZIMA
Napier
2008

Maximal Clique

Near Linear Speedup



Significant Memory Requirements



Biclique

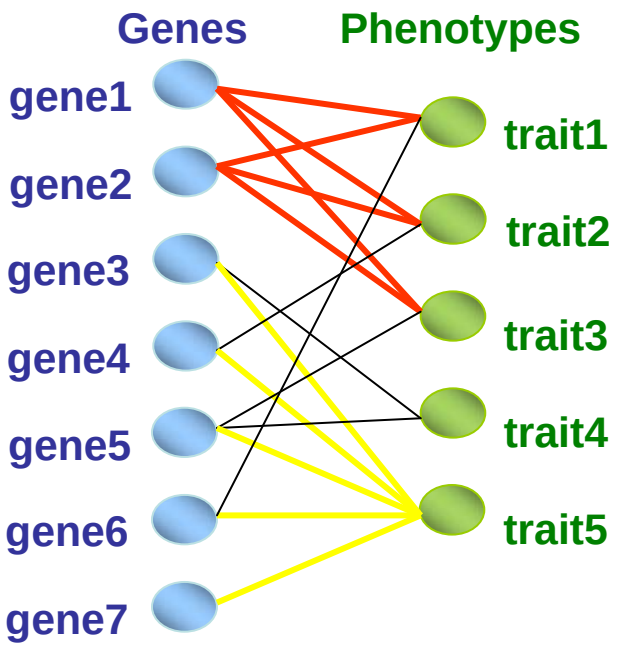
Concentrate on Bipartite Graphs

Previous Algorithms Make Unwarranted Assumptions

Bookkeeping

Branch & Bound

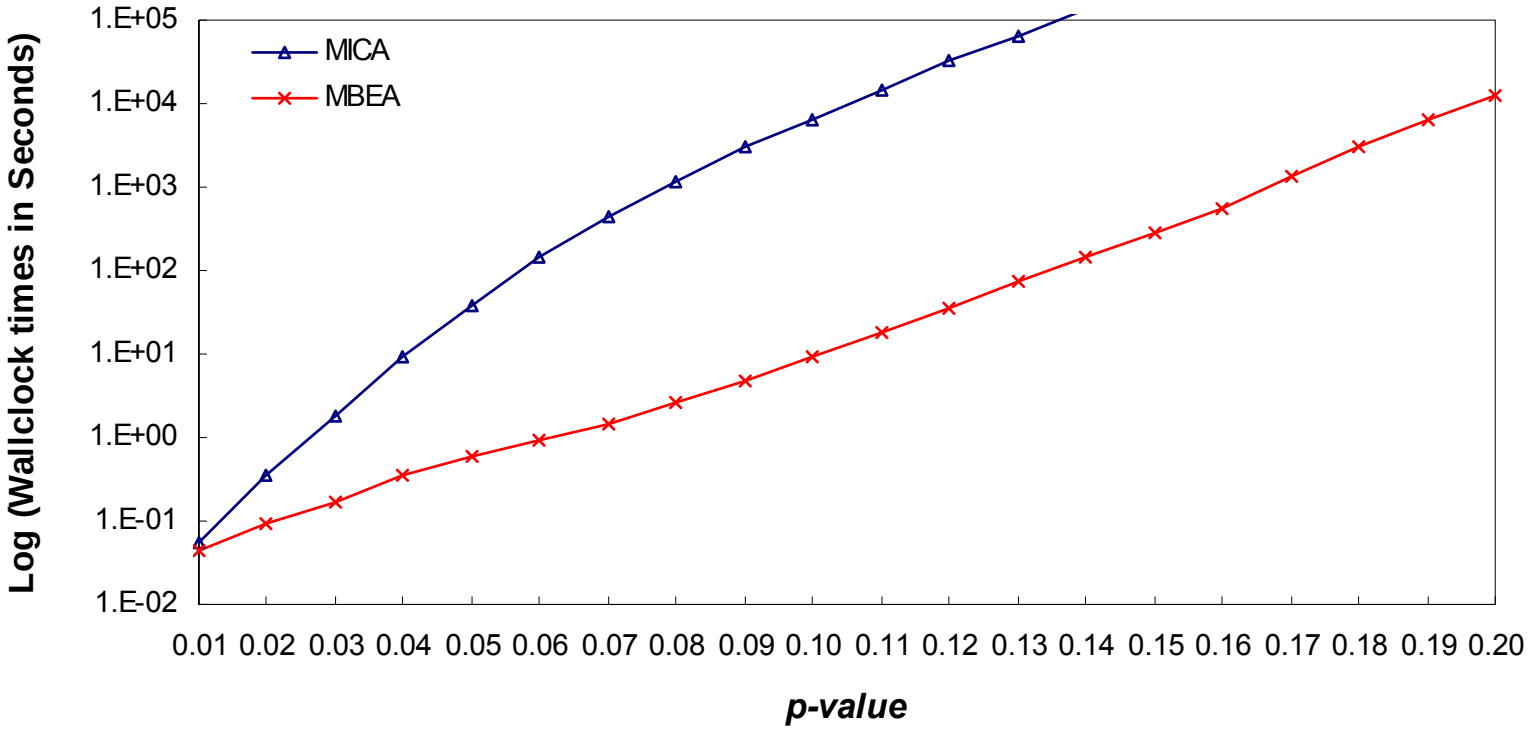
Ontological Discovery



Biclique

Observed Biclique Runtimes

2 ~ 3 orders of magnitude faster than the best previous alternative



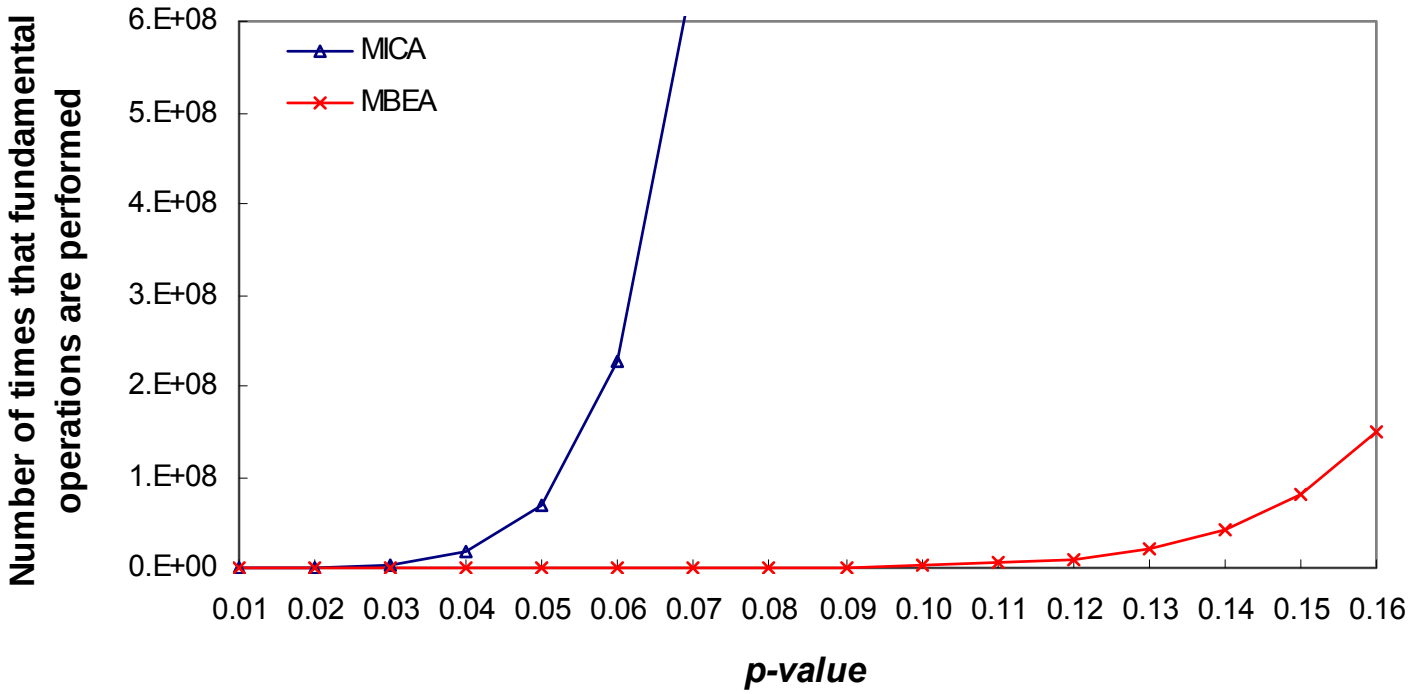
Time Complexity: $O(dn^2B)$, where d is maximum degree and B is the number of maximal bicliques.

Keys: preprocess and exploit structure. Sound familiar?

Biclique

Discretionary Power

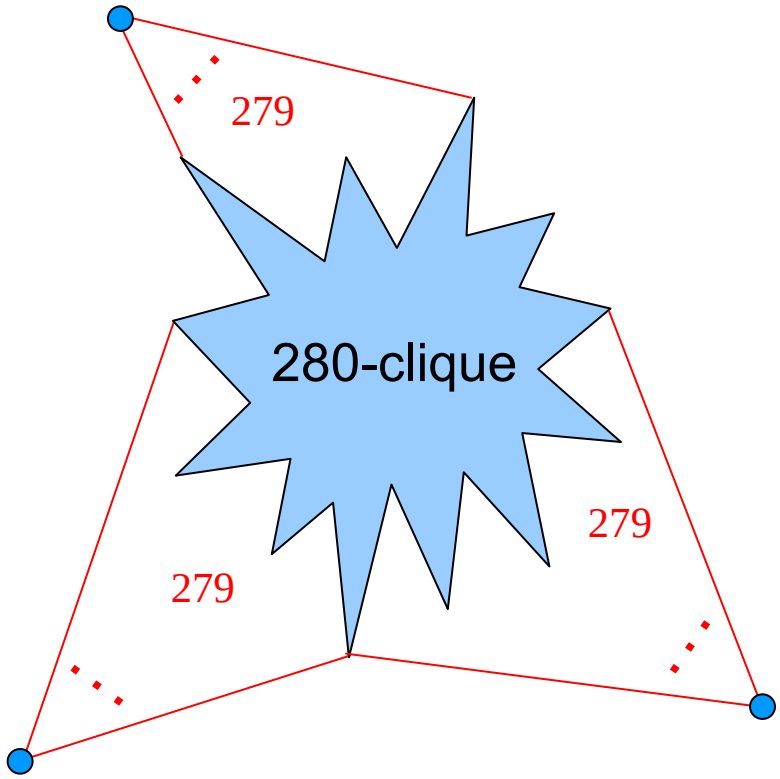
We can now explore much denser graphs, as shown by edge weights.



NZIMA
Napier
2008

Paraclique

- Clique gloms onto highly connected vertices.
- Here a 280-clique is transformed into a 466-paraclique.
- Edge density remains north of about 95%.
- Lift and separate.



466-paraclique

NZIMA
Napier
2008

Collaborators

Research Scientists (Incomplete!):

Mikael Benson
Elissa Chesler
Frank Dehne
Mike Fellows
Ivan Gerling
Dan Goldowitz
Malak Kotb
Mark Ragan
Arnold Saxton
Brynn Voy
Rob Williams
Bing Zhang

Current Students:

Bhavesh Borate
Patricia Carey
John Eblen
Jeremy Jay
Zuopan Li
Sudhir Naswa
Andy Perkins
Vivek Philip
Charles Phillips
Gary Rogers
Jon Scharff
Yun Zhang

NZIMA
Napier
2008

Geeks Я Us