

Fixed-Parameter Tractability: Foundations, Implementations and Applications

Mike Langston

Faisal Abu-Khzam, Nicole Baldwin, Rebecca Collins,
Mahesh Dorai, John Eblen, Lan Lin, Daniel Lucio,
Jon Scharff, Xinxia Peng, Pushkar Shanbhag,
Yongling Song, Henry Suters, Chris Symons,
Ian Watkins

Salient Features

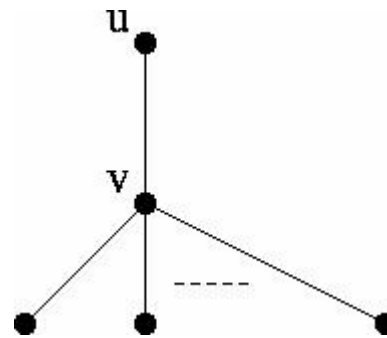
- Kernelization and branching methodology
- Parallelism based on search tree decomposition
- Fixed-parameter paradigms in optimization

The Vertex Cover Project

- Preprocessing plus kernelization
- Sequential branching
- Parallel branching
 - Search tree decomposition
 - Dynamic load balancing

Preprocessing

- Low degree rules (e.g., degree one)



- High degree rule
- Resultant graph has size at most $k(1+k/3)$

Kernelization

- Based on linear programming:

- minimize: $\sum_{u \in V(G)} X_u$

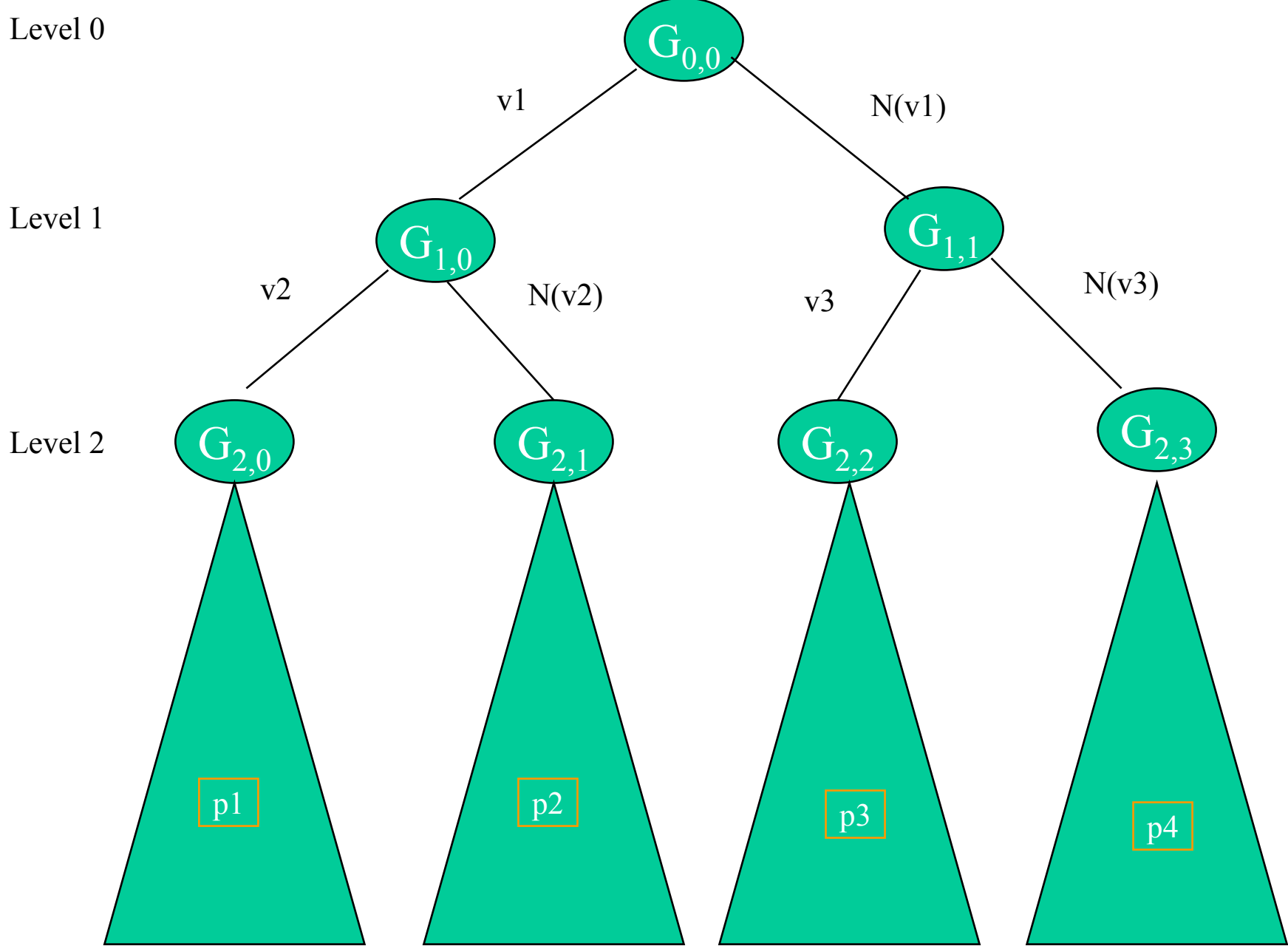
- subject to: $X_u + X_v \geq 1 \forall uv \in E(G)$

- where: $X_u \geq 0 \forall u \in V(G)$

- Resultant graph has size at most $2k$

Parallel Branching

- Focus on vertices of high degree
- Interleave preprocessing rules
- Based on a decomposition of the search tree
- Exploits the number of processors available
- Implemented using SSH or grid middleware
- Tasks run in a virtually unstructured manner



Intriguing Results on Synthetic Graphs of Size 600

Graph name	Sequential Reduction	Sequential branching	Parallel branching
RG30	1 second	Halted after two days	5 sec
RG31	1 second	Halted after two days	4 sec
RG32	1 second	Halted after two days	4 sec

Problems With the Parallel Method

- Extremely data dependent
- Unusually sensitive to rule order
- Super-super linear speedup on “yes” instances can equate to super-super linear slowdown on “no” instances
- Load balancing needed to achieve scalability

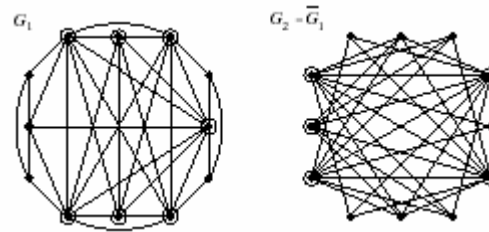
Solution: Dynamic Decomposition

Representative Results on Large Non-Synthetic Graphs

Graph Name	Graph Size	Cover Size	Instance Type	Sequential Reduction	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

Sample Applications

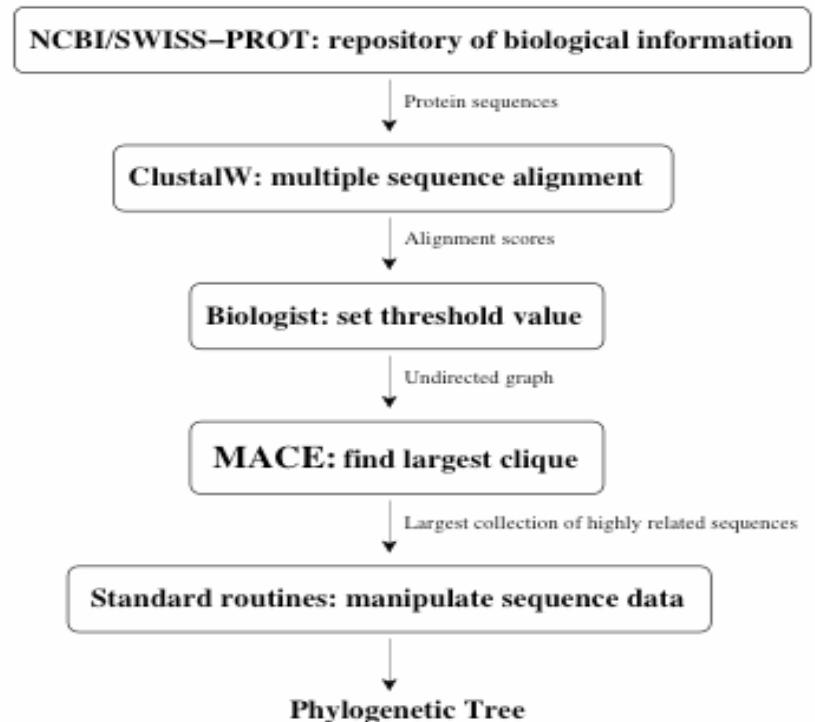
- Dual to Clique



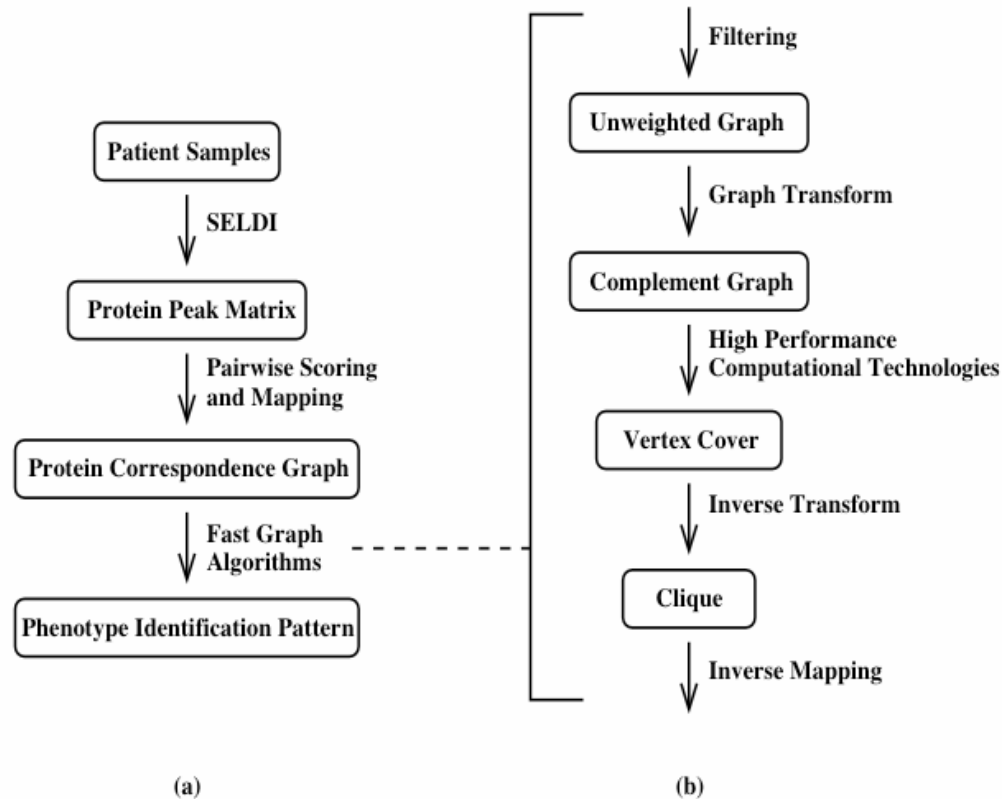
- Clique Useful in
 - - Genomics (e.g., Phylogeny, Microarray Analysis)
 - Proteomics (e.g., SELDI)
- Codes Released in Clustal-XP

Applications to Phylogeny

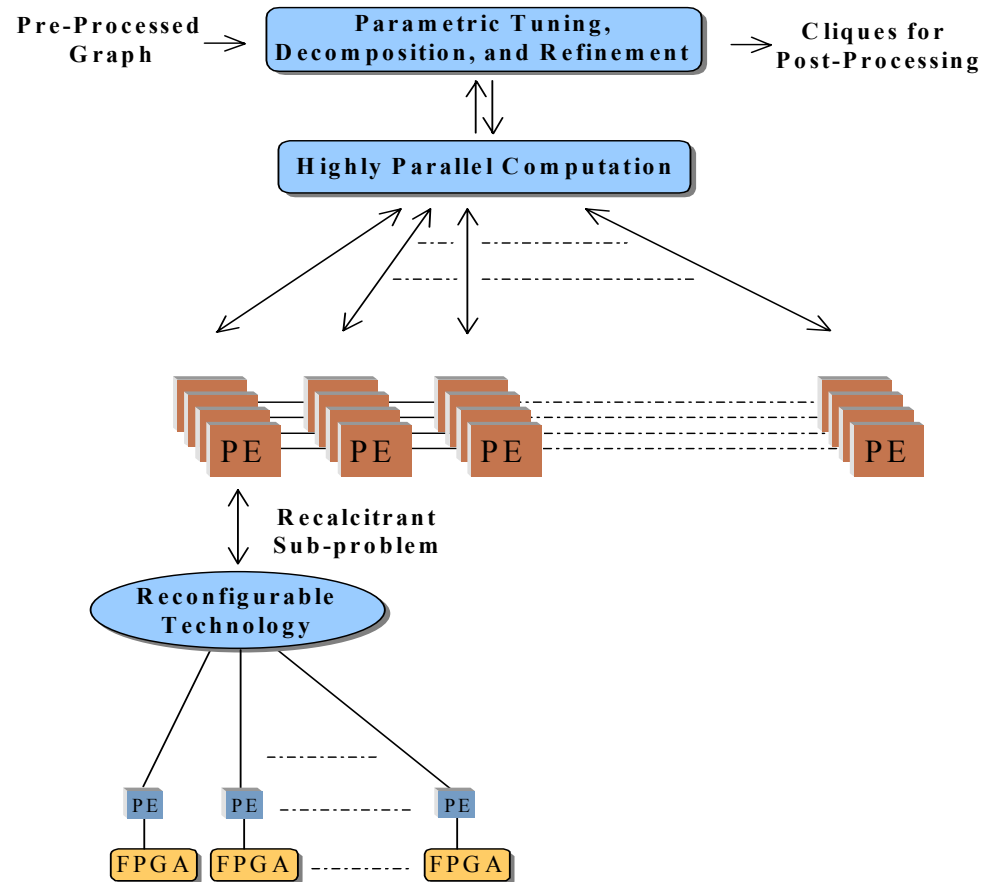
A phylogenetic tree organizes life forms according to their putative evolutionary relationships.



Applications to SELDI



The Clique Compute Engine



Applications to Microarray Data Analysis

