Capturing Topology in Graph Pattern Matching

Shuai Ma, Yang Cao, Wenfei Fan, Jinpeng Huai, Tianyu Wo

University of Edinburgh
Graphs are everywhere, and quite a few are huge graphs!

Graph searching is a key to social searching engines!
Graph Pattern Matching

- Given two directed graphs $G_1$ (pattern graph) and $G_2$ (data graph),
  - decide whether $G_1$ “matches” $G_2$ (Boolean queries);
  - identify “subgraphs” of $G_2$ that match $G_1$

- Applications
  - Web mirror detection/Web site classification
  - Complex object identification
  - Software plagiarism detection
  - Social network/biology analyses
  - ...

- Matching Models
  - Traditional: Subgraph Isomorphism
  - Emerging applications: Graph Simulation and its extensions, etc..

A variety of emerging real-life applications!
Subgraph Isomorphism

- Pattern graph $Q$, subgraph $G_s$ of data graph $G$
- $Q$ matches $G_s$ if there exists a bijective function $f: V_Q \rightarrow V_{G_s}$ such that
  - for each node $u$ in $Q$, $u$ and $f(u)$ have the same label
  - An edge $(u, u')$ in $Q$ if and only if $(f(u), f(u'))$ is an edge in $G_s$

- **Goodness:**
  - Keep exact structure topology between $Q$ and $G_s$

- **Badness:**
  - Decision problem is NP-complete
  - May return exponential many matched subgraphs
  - In certain scenarios, too restrictive to find matches

These hinder the usability in emerging applications, e.g., social networks
Graph Simulation

- Given pattern graph \( Q(V_q, E_q) \) and data graph \( G(V, E) \), a binary relation \( R \subseteq V_q \times V \) is said to be a match if
  - (1) for each \((u, v) \in R\), \( u \) and \( v \) have the same label; and
  - (2) for each edge \((u, u') \in E_q\), there exists an edge \((v, v') \in E\) such that \((u', v') \in R\).

- Graph \( G \) matches pattern \( Q \) via graph simulation, if there exists a total match relation \( M \)
  - for each \( u \in V_q \), there exists \( v \in V \) such that \((u, v) \in M\).

- Goodness:
  - Quadratic time solvable

- Badness:
  - Lose structure topology (how much? open question)
  - Return a single unique matched subgraph

Subgraph isomorphism (NP-complete) vs. graph simulation (O\(n^2\))!
Graph Simulation

Set up a team to develop a new software product

Graph simulation returns F3, F4 and F5; Subgraph isomorphism returns empty!

Subgraph Isomorphism is too strict for emerging applications!
Graph Simulation Loses Structures

**Connected pattern graphs match disconnected subgraphs**

- $S(HR) = \{HR\}$
- $S(SE) = \{SE\}$
- $S(Bio) = \{Bio_1, Bio_2\}$

These motivate us to propose a new matching model!
Strong Simulation: A New Model

Strong Simulation = Graph Simulation + Duality + Locality

• **Duality** (dual simulation)
  – Both child and parent relationships
  – Simulation considers only child relationships

• **Locality**
  – Restricting matches within a ball
  – When social distance increases, the closeness of relationships decreases and the relationships may become irrelevant

• The semantics of strong simulation is well defined
  – The matching results are unique

Striking a balance between expressiveness and complexity!
Duality and Locality

- Pattern graph Q matches data graph G via **dual simulation** if there exists a binary match relation $S \subseteq V_Q \times V$ such that:
  - for each $(u, v) \in S$, $u$ and $v$ have **the same label**; and
  - for each $u \in V_q$, there exists $v \in V$ such that $(u, v) \in S$; and
    - for each edge $(u, u_1)$ in $E_q$, there is an edge $(v, v_1)$ in $E$ with $(u_1; v_1) \in S$; **-> Child relationships**
    - for each edge $(u_2, u)$ in $E_q$, there is an edge $(v_2; v)$ in $E$ with $(u_2, v_2) \in S$. **-> Parent relationships**

**Dual simulation**: bring duality intro graph simulation!

- The matched subgraph must be a connected subgraph
  - falling into a **ball** with center $v$ and radius $d_Q$ (diameter of Q)
  - containing the ball center $v$

**Strong simulation**: bring locality intro dual simulation!
Properties of Strong Simulation

Connectivity: Connected pattern graphs match disconnected subgraphs

- $S(HR) = \{HR\}$
- $S(SE) = \{SE\}$
- $S(Bio) = \{Bio_1, Bio_2\}$

Cycles: Cyclic pattern graphs match tree subgraphs

- $S(HR) = \{HR\}$
- $S(SE) = \{SE\}$
- $S(Bio) = \{Bio_1, Bio_2\}$

Strong simulation preserves more topology structures!
Properties of Strong Simulation

**Locality:** The diameter of matched subgraphs is bounded by $2d_Q$.

- **Graph simulation** does NOT have this property.
- **Subgraph isomorphism** does have this property.

**Bounded matches:** The number of matched subgraphs is bounded by $|V|$.

- **Graph simulation** finds at most one matched subgraph.
- **Subgraph isomorphism** may find exponential number of matched subgraphs.

**Bounded cycles:** The length of cycles in matched subgraphs is bounded by the ones in pattern graphs.

- **Graph simulation and strong simulation** does NOT have this property.
- **Subgraph isomorphism** does have this property!

**Strong simulation preserves** more topology structures!
Properties of Strong Simulation
Properties of Strong Simulation

- \(\text{db} \rightarrow \text{SN} \rightarrow \text{graph} \rightarrow Q4\)
- \(\text{db}_1 \rightarrow \text{SN}_1 \rightarrow \text{SN}_2 \rightarrow \text{graph}_1 \rightarrow \text{graph}_2 \rightarrow \text{G}_4\)
- \(\text{SN}_3 \rightarrow \text{SN}_4 \rightarrow \text{G}_4, i, j (i, j \in [1, 2])\)
- \(\text{db}_1 \rightarrow \text{SN}_i \rightarrow \text{graph}_j\)
Properties of Strong Simulation

<table>
<thead>
<tr>
<th>Matching</th>
<th>children</th>
<th>parents</th>
<th>connectivity</th>
<th>cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>≺</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>✓ (directed), × (undirected)</td>
</tr>
<tr>
<td>≺\text{D}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓ (directed &amp; undirected)</td>
</tr>
<tr>
<td>≺^L\text{D}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓ (directed &amp; undirected)</td>
</tr>
<tr>
<td>≪</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓ (directed &amp; undirected)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>locality</th>
<th>matches</th>
<th>Bisimilar&amp;b’ed-cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

A balance between expressiveness and complexity!
Properties of Strong Simulation

- If Q matches G, via subgraph isomorphism, then Q matches G, via strong simulation

- If Q matches G, via strong simulation, then Q matches G, via dual simulation

- If Q matches G, via dual simulation, then Q matches G, via graph simulation
Algorithms for Strong Simulation

- A *cubic* time algorithm
  - Graph simulation: *Quadratic time*
  - Subgraph isomorphism: *NP-Complete*
- A *distributed* algorithm
  - Using the *data locality* property
  - Real life graphs are typically distributed

Connectivity theorem:

- Q matches G, via *dual simulation*
- for any connected component $G_c$ of the match graph w.r.t. the maximum match relation of Q and G,
  - Q matches $G_c$,
  - $G_c$ is exactly the match graph w.r.t. the maximum match relation of Q and $G_c$

*Nontrivial extension* of the algorithm for graph simulation!
Optimization Techniques

Minimizing pattern graphs ($Q \equiv Q_m$): An quadratic time algorithm

Given pattern graph $Q$, we compute a minimized equivalent pattern graph $Q_m$ such that for any data graph $G$, $G$ matches $Q$ iff $G$ matches $Q_m$, via strong simulation.

Dual simulation filtering

- First compute the matched subgraph of dual simulation,
- Then project on each ball of the data graph

Connectivity pruning

- Based on the connectivity theorem
Experimental Study

Real life datasets:
Amazon product co-buy network: 548,552 nodes and 1,788,725 edges
YouTube video network: 155,513 nodes and 3,110,120 edges

Synthetic graph generator: (10^7 nodes and 251,188,643 edges)

Three parameters:
1. The number n of nodes;
2. The number nα of edges; and
3. The number l of node labels

Algorithms:
- Strong simulation algorithm Match and its optimized version Match+.
- Graph simulation algorithm Sim [HHK, FOCS 95]
- Approximate matching algorithm TALE [TP, ICDE 08]
- Maximum common subgraph algorithm VF2 [CN, 2006]

Machines:
- PC machines with Intel Core i7 860 CPUs and 16GB memory
Experiments - Quality

The results of strong simulation are **more realistic!**
The results of strong simulation are more compact!
70%-80% found by subgraph isomorphism are retrieved by strong simulation

Up to 50% found by subgraph isomorphism are retrieved by graph simulation
Experiments - Quality

Strong simulation effectively reduces the number of match results!
Experiments - Quality

<table>
<thead>
<tr>
<th>#nodes</th>
<th>[0, 9]</th>
<th>[10, 19]</th>
<th>[20, 29]</th>
<th>[30, 39]</th>
<th>[40, 49]</th>
<th>≥ 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>0</td>
<td>98</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>YouTube</td>
<td>0</td>
<td>21</td>
<td>18</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Synthetic</td>
<td>0</td>
<td>187</td>
<td>113</td>
<td>65</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Sizes of matched subgraphs

- Pattern graphs have **10** nodes
- Graph simulation (returns a single graph)
  - Amazon: **103**
  - YouTube: **177**
  - Synthetic: **311**

The sizes of matched subgraphs are **small!**
1. Our algorithms scale well;
2. Optimization techniques are effective (reduce about 1/3 time);
3. The time gap between Sim and Match is tolerable, considering the matching quality that Match improves.
Summary

We have proposed and investigated strong simulation to rectify the problems of subgraph isomorphism and graph simulation.
- the duality to preserve the parent relationships
- the locality to eliminate excessive matches.

We identify a set of criteria for topology preservation, and show that strong simulation preserves the topology of pattern graphs and data graphs.
- Children, Parent, Connectivity, Cycles, Bounded matches
- Bounded cycles, Bisimilarity

We show that strong simulation retains the same complexity as earlier extensions of simulation (a cubic-time algorithm)
- Optimizations: minimization, dual simulation filtering, connectivity pruning

We present the locality property of strong simulation, which allows us to effectively conduct pattern matching on distributed graphs

A new matching model with a balance between complexity and expressiveness