Hierarchical High Level Information Fusion
Using Graph Structures, Subgraph Matching and State Space Search

Moises Sudit
Rakesh Nagi
Kedar Sambhoos

February 7, 2007

In Nearly Every Application:

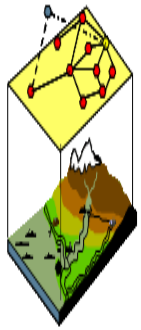
We Are Drowning in Data

- More Sources
- More Source Types: NTM, Tactical, Commercial
- Wider Bandwidths
- More Connectivity: GCCS, DII/COE

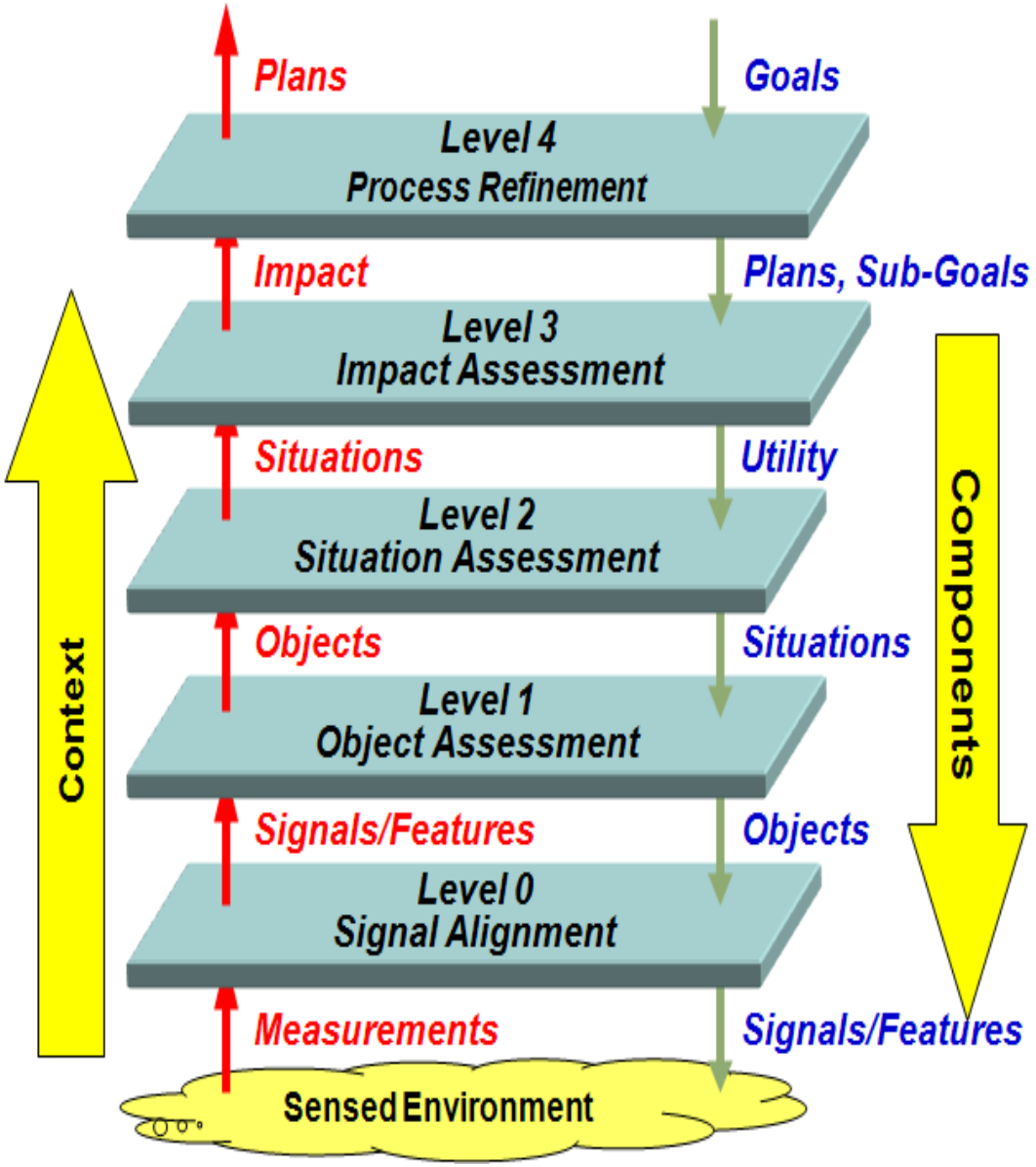
DATA:
Observations & Measurements

But Starving for Information

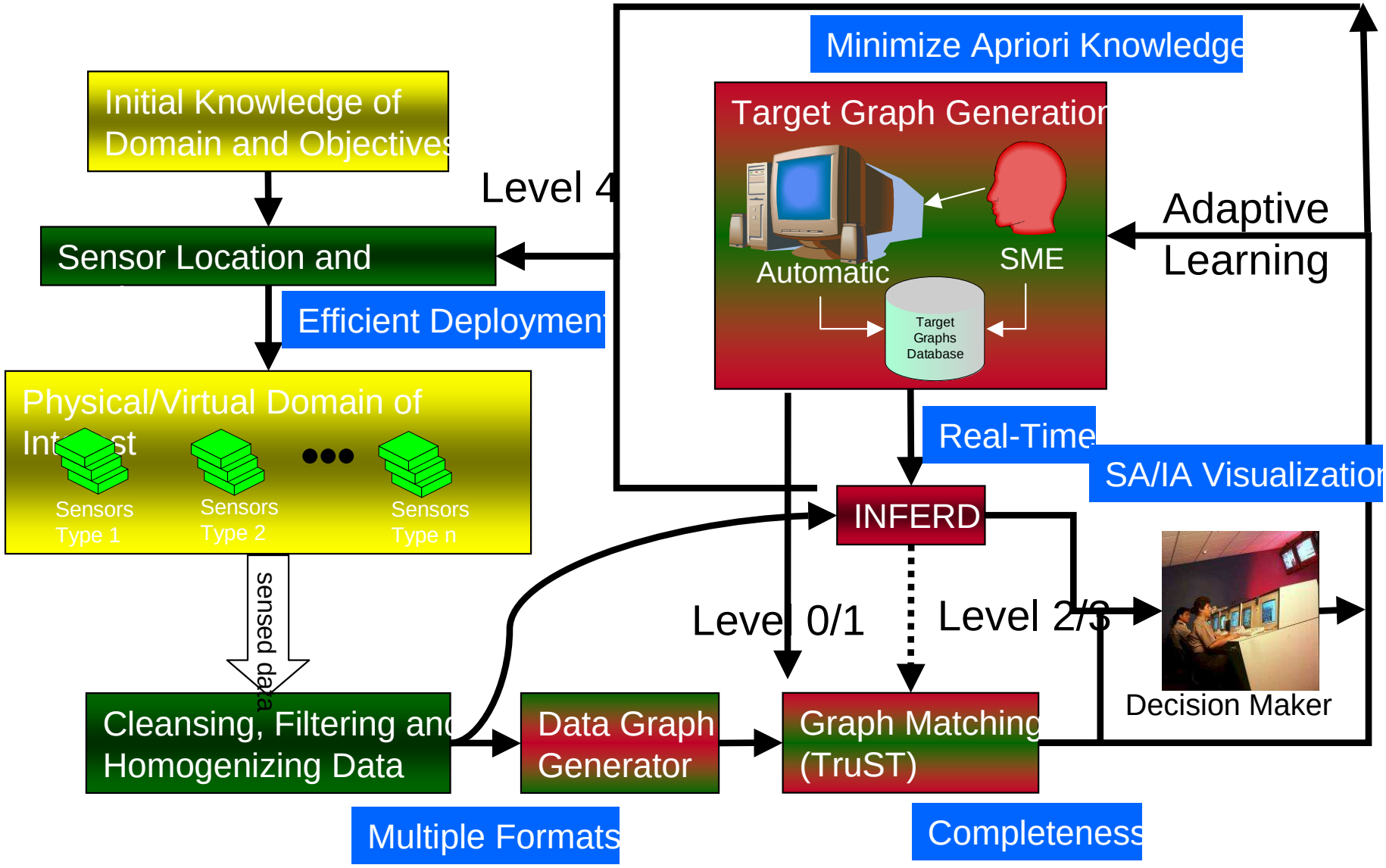
- How does data relate to the situation (context)?
- What is its significance (impact on my plans)?
- What is its quality? Can I trust it?
- How can it be confirmed, refuted or refined?



INFORMATION:
Organized Data

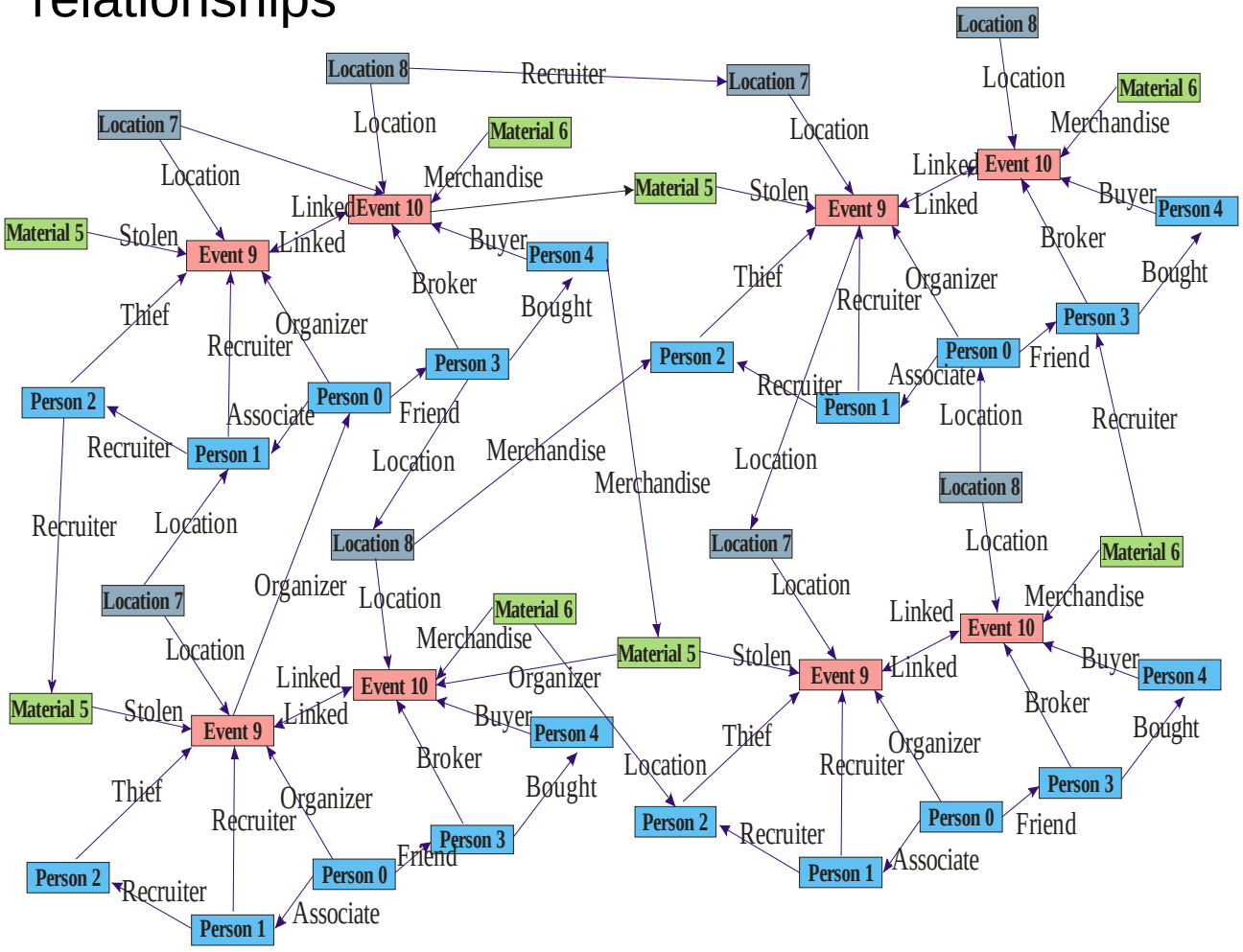


Technology Approach and Justification

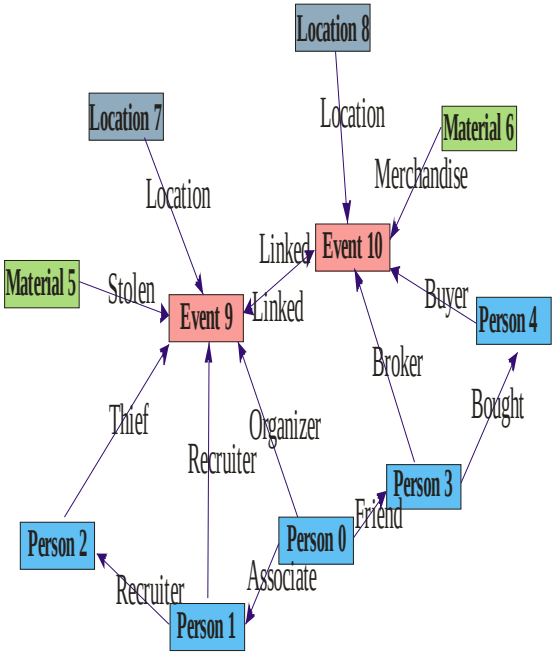


Data Graph and Template Graph

- Decision-maker often encounter complex uncertain situations and they need to develop *relationship* between situational elements.
- *Attributed Relational Graphs* (ARGs) represent situational elements and relationships



Data Graph



Template Graph

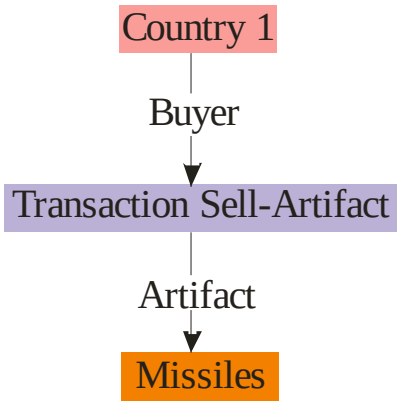
Graph Matching Structures

		Exact	Inexact
Syntactic (labelled-graphs)		Graph Isomorphism	Graph Homeomorphism
Semantic (attributed-graphs)	Structured	- attributed-vertex graph - Greedy Search	- attributed-vertex graph - Greedy Search
	Semistructured	- attributed-vertex graph - Graph Isomorphism - Graph Bundling	- attributed-vertex graph - Graph Homeomorphism - Graph Bundling
	Unstructured	Heuristics	Heuristics

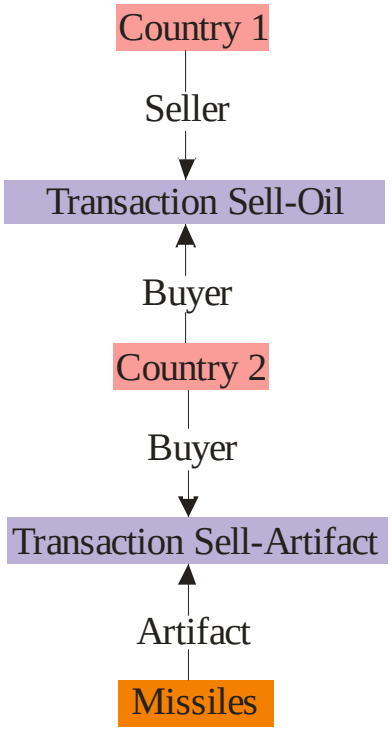
- Structured data is rigidly organized & well defined: Predictable
- Semistructured data is organized enough to be predictable
  - Data is organized in semantic entities
  - Similar entities are grouped together but,
    - Entities in the same group may not have the same attributes
    - The order of the attributes is not necessarily important
    - The presence of some attributes may not always be required
    - The size of same attributes of entities in a same group may not be the same
    - The type of the same attributes of entities in a same group may not be of the same type
- Unstructured data is disordered and unruly: Unpredictable

- Difference between Graph Matching and Subgraph Matching

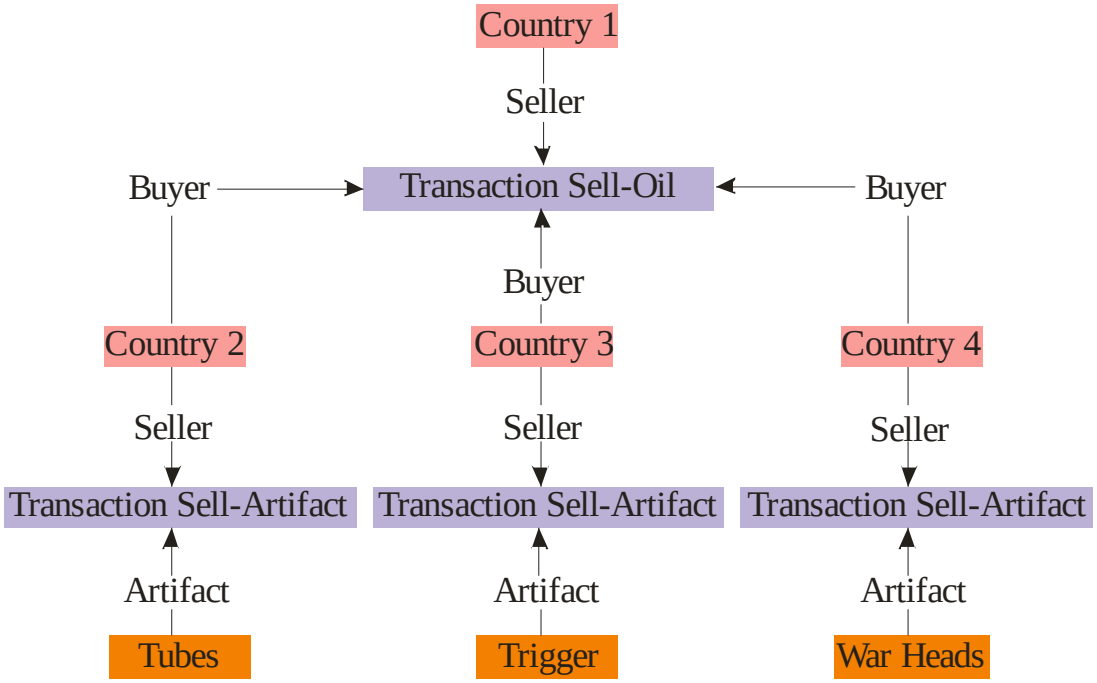
Isomorphism, Homeomorphism and Bundle Matching



Isomorphism



Homeomorphism



Bundle matching

Motivation for our Approach

- Enhancements in Level 2 and 3 fusion capabilities through a new class of models and algorithms in graph matching.
- Create a Framework to Span Temporal Decision-Making (from “real-time” to forensics”)
- Most of literature is in subgraph isomorphism (Cordella *et al.*, 2004)
- Most of the matching techniques are restricted to Simple Graphs
- *Truncated branch and bound* performs several times better than the local search (Zhang, 2000) motivates for Unstructured Graphs
- Attributed Relational Graphs are Visually Appealing, but don’t give us much value-added compared with Classical Graph Structures in Solving Graph Matching Problems.
  Theorem: Any graph matching problem over an attributed-graph (attributed-vertices and attributed-edges) can be polynomially transformed in to an equivalent attributed-vertex graph (no attributes on the edges)

1. L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(10):1367, 2004.
2. W. Zhang. Depth-first branch-and-bound versus local search: A case study. In Proc. 17th National Conf. on Artificial Intelligence, pages 930–935, 2000.

Definitions and Notation

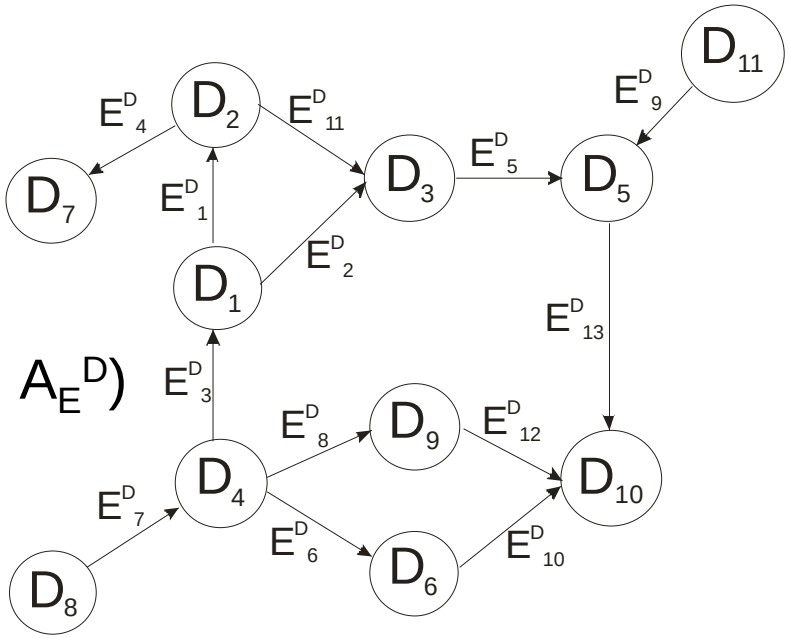
- Attributed Graph Structure

$$G = (V, E, A_v, A_E)$$

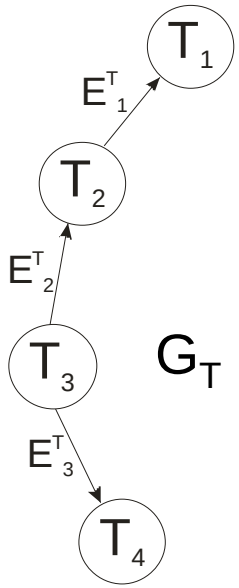
where V - the set of nodes; E - the set of arcs;

A_v - the set of node attributes; A_v - the set of arc attributes.

$G_D = (V_D, E_D, A_V^D, A_E^D)$



Data Graph



Template

$G_T = (V_T, E_T, A_V^T, A_E^T)$

Most Graph Matching Techniques based on Triplets

- Triplets:

Template Data Graph

- One-to-one Scores of Nodes and Arcs:

$S(v_i, v_j) = 0.9 \quad S(v_h, v_k) = 0.7 \quad S(e_{ih}, e_{jk}) = 0.8$

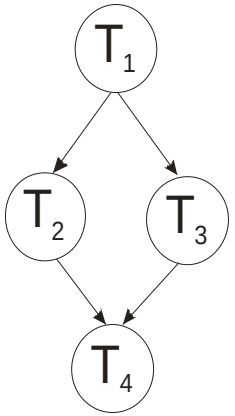
- Similarity Functions for Triplets:

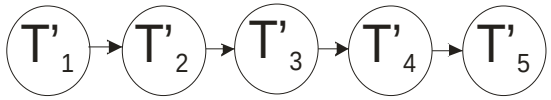
$$f_{ihjk} = f(S(v_i, v_j), S(v_h, v_k), S(e_{ih}, e_{jk}))$$
$$= 0.8$$

Triplet Deficiency

Problem:

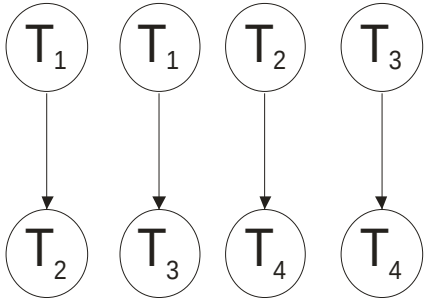
- Two totally different topological templates could receive exactly the *same* similarity values.

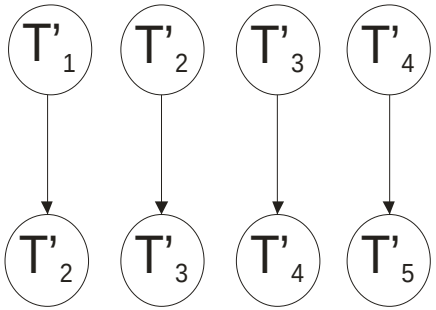




Goal:

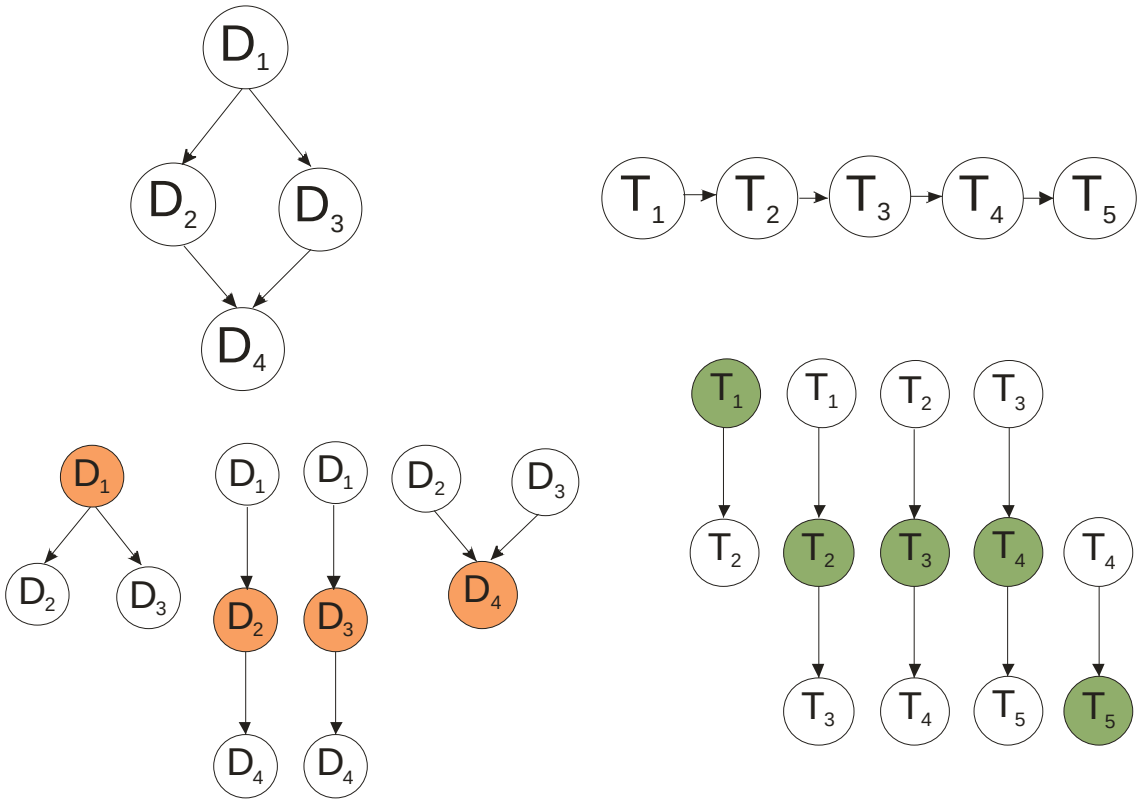
- Develop an algorithm that uses the current system's rule based scoring engine while increasing its *topological accuracy* during matching.



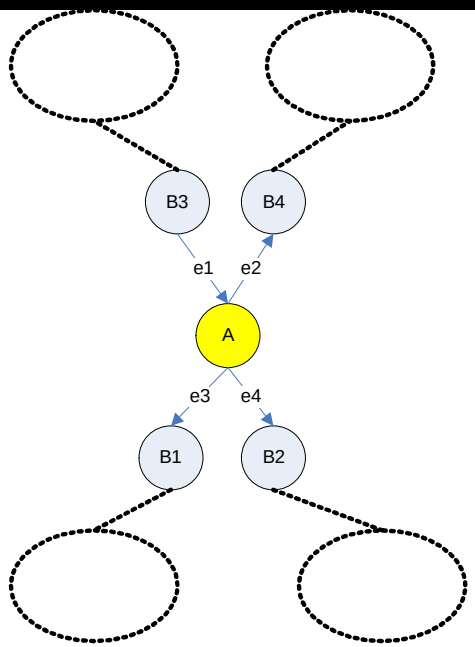


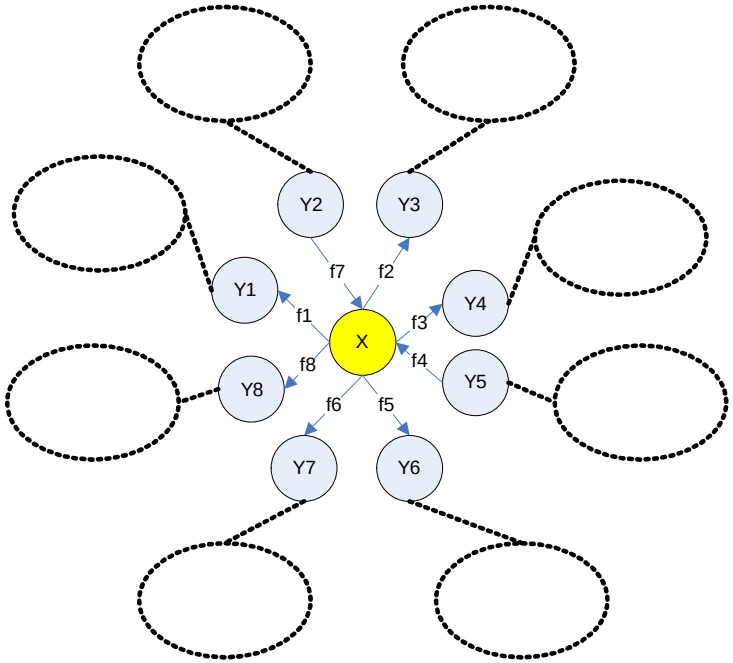
1-Hop Neighborhood Breakdown

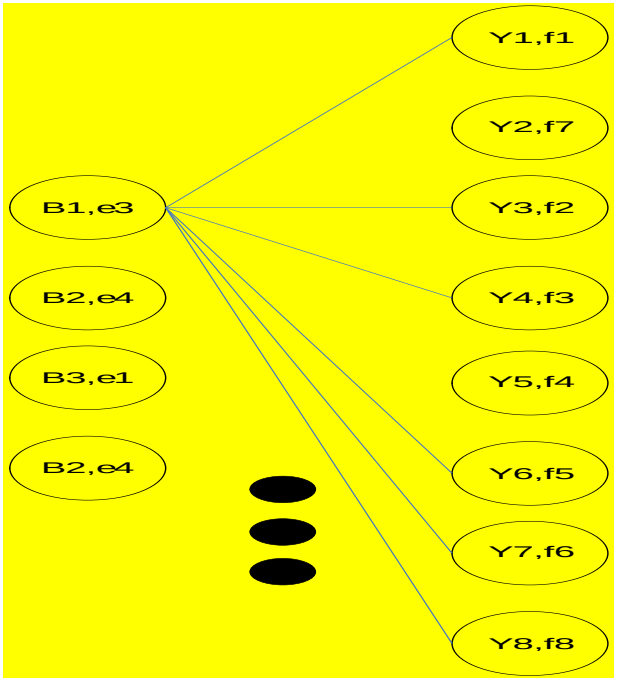
- 1-Hop Neighborhood
  - A root node and all other nodes of edge distance 1 away.
  - Benefit:



1-Hop Neighbor Matching







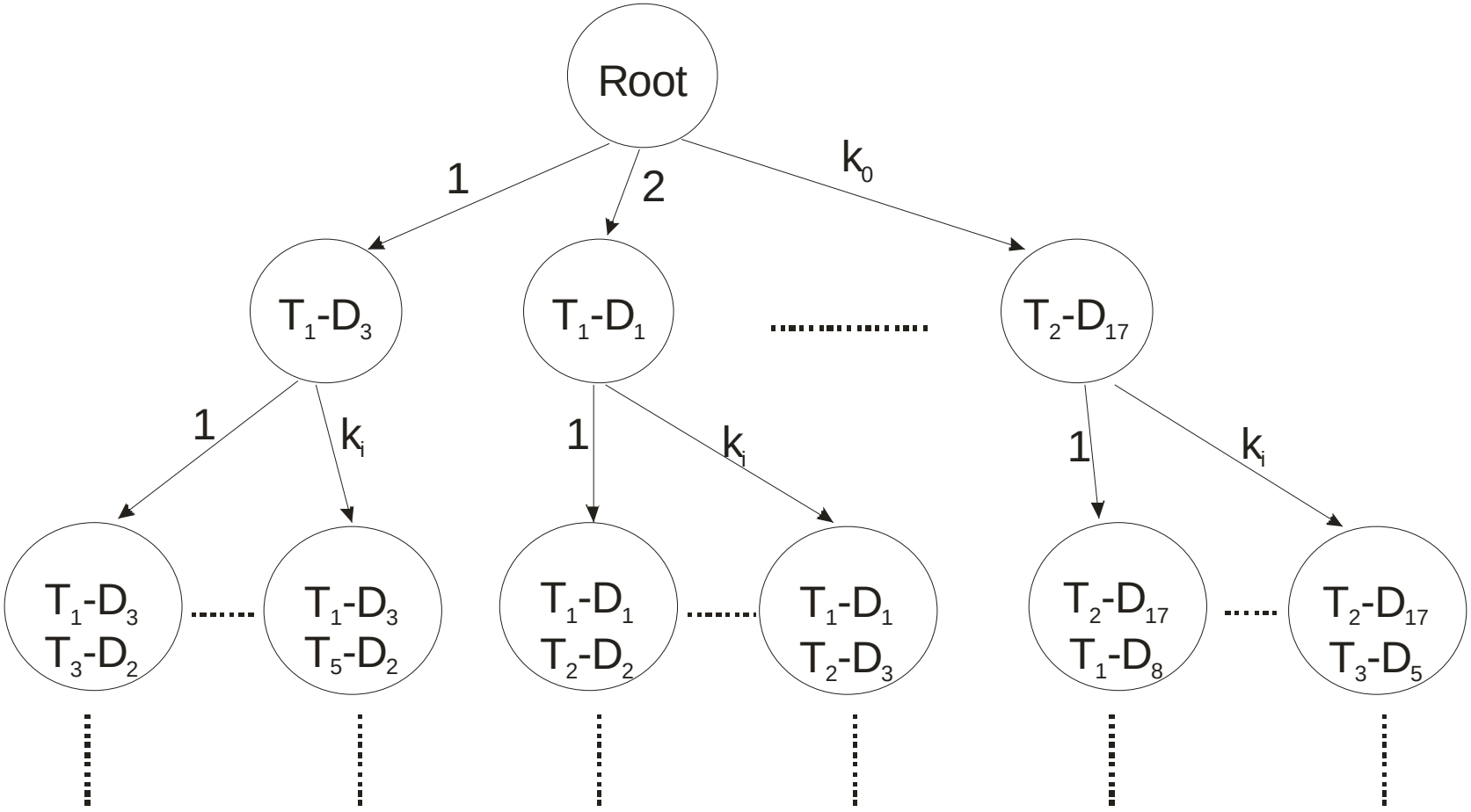
- **Algorithm**
  - **Step 1:** Compute a node score, denoted as C_{ij}, for each node in the template graph to each node in the data graph.
  - **Step 2:** Compute the scores, denoted as W_{ij}, for the 1-Hop neighbors of each root node pair.
  - The score is given by $\alpha\, C_{ij} + (1\text{-}\alpha)\, W_{ij}$

"α" is the Score vs. Topology Parameter

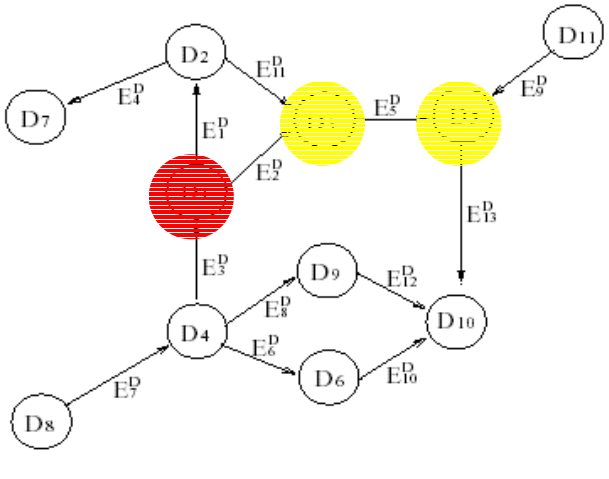
TruST (Truncated Search Tree) Algorithm

- Advantages:
  - Easier to implement, faster to execute and requires less computing resources.
  - User can decide tradeoff between time and quality.
  - Converges to optimality.
- Disadvantages:
  - Not guaranteed to yield an optimal solution.
  - Parametric approach
  - Greedy in nature.
- Parameters of controlling state space
  - k_0 : The number of child problems of the root.
  - k_i : The number of child problems of the problem at level i-1.
  - β : The total number of problems.
  - δ : The total number of levels.

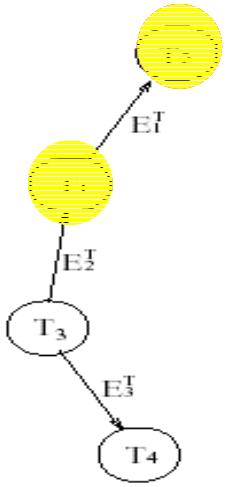
TruST Algorithm (cont'd)



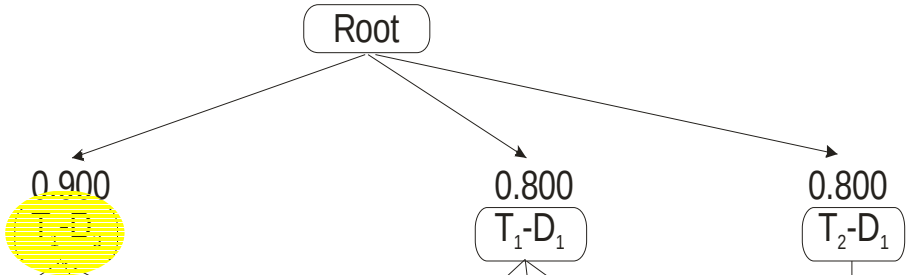
Example of Best-Bound with Topology



Data Graph



Template



T_1		T_2		T_3		T_4	
D_3	0.9	D_1	0.8	D_6	0.77	D_{10}	0.6
D_1	0.8	D_5	0.75	D_1	0.75	D_7	0.45
D_9	0.75	D_3	0.71	D_2	0.63	D_2	0.4
D_6	0.73	D_7	0.68	D_4	0.52	D_6	0.35
D_5	0.7	D_9	0.55	D_8	0.44	D_4	0.28
D_4	0.65	D_6	0.53	D_9	0.27	D_8	0.18
D_2	0.4	D_4	0.23	D_3	0.23	D_1	0.15
D_7	0.17	D_2	0.21	D_7	0.17	D_3	0.14
D_{10}	0.15	D_{10}	0.13	D_{11}	0.11	D_5	0.12
D_{11}	0.11	D_8	0.09	D_5	0.09	D_9	0.1
D_8	0.09	D_{11}	0.05	D_{10}	0.05	D_{11}	0.07

1-Hop Neighbor Optimal Assignments

Design of experiment

- But we still don't know what are the *optimal parameters* to be set.
  - Responses: *Runtime* and *Maximum Heuristic Score*.
  - 2-level factorial design: 9 {512 (2^9) runs}
  - Screening experiment: 1/4th factorial design with 128 runs.
  - depth $\delta = 6$

Factors	Levels		Range	
Number of nodes in data graph	25	50	-	-
Number of nodes in template graph	6	10	-	-
Data graph density	10%	15%	0%	100%
Template graph density	24%	50%	0%	100%
t	0.1	1	0	1
α	0.1	1	0	1
k_0	50	100	0	mn
k_i	50	100	0	mn
β	50	300	0	$\prod_i k_i$

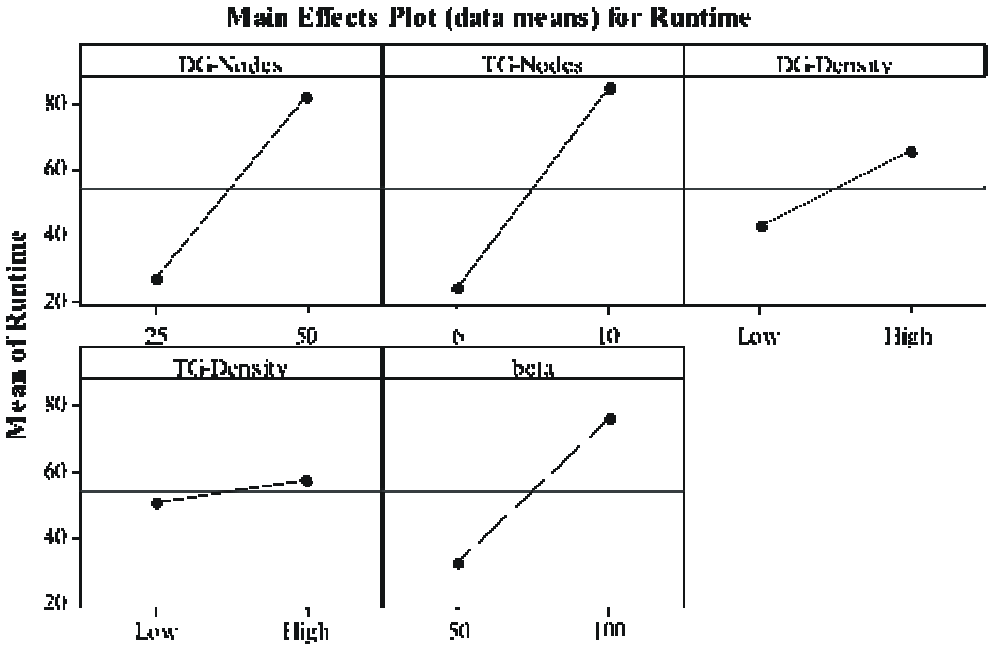
m =The number of data graph nodes.

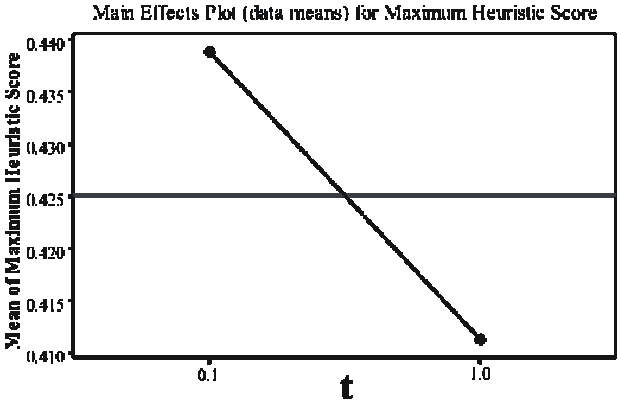
n =The number of data graph

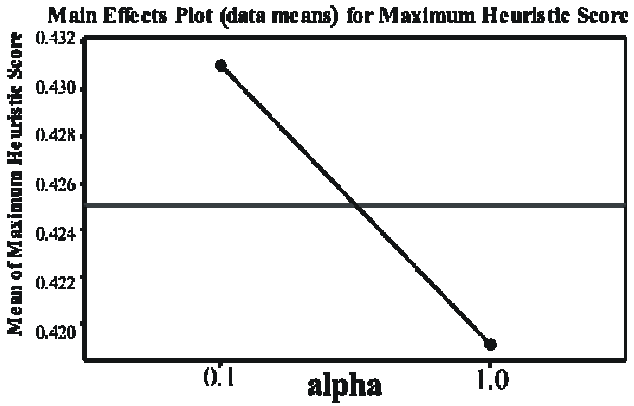
Design of experiment

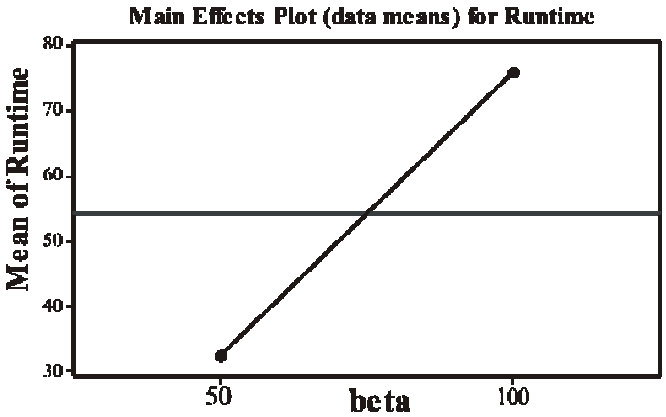
INTUITIVE CONCLUSIONS

- Template graph density has *less effect* on runtime as compared to data graph density.
- β dominates all other parameters for runtime.
- The constant α are much more significant in obtaining quality of the heuristic.



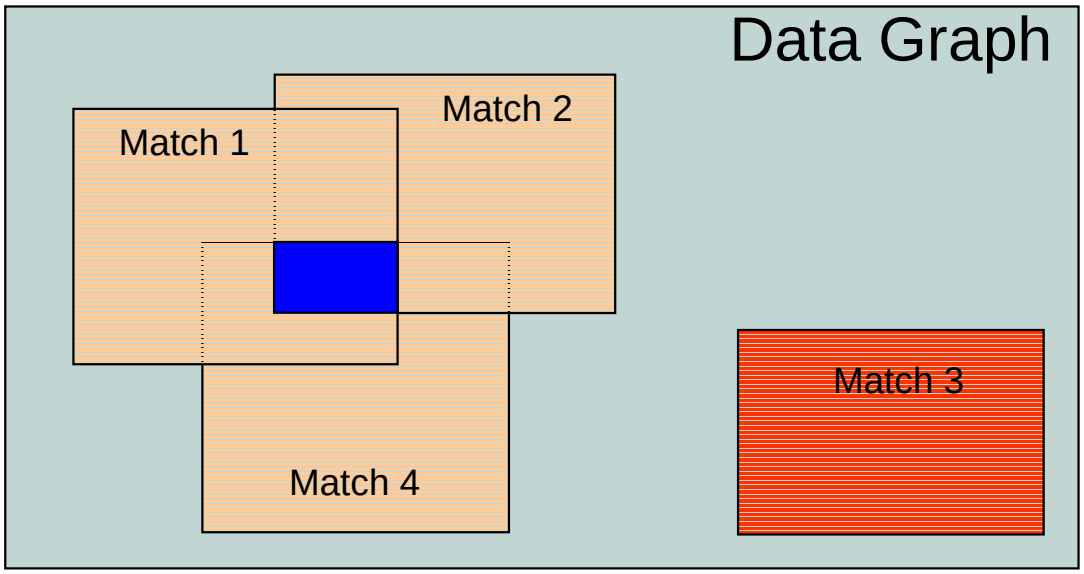






Clustering

- TruST results in *large number* of matches.
- What do the results tell us?



- To analyze the results we group the results using K means clustering

K-Means Clustering: Hypercube vs. Fuzzy Hamming Distance

- Hypercube Distance
  - Result becomes a point M in the N dimensional hypercube
  - The metric to minimize distance will determine the shape of the optimum clusters.
  - *Hypercube distance measure* $(\Delta(A,B))$ is used

$$\Delta(A,B) = 1 - \frac{\mu(A \cap B)}{\mu(A \cup B)}$$

where,

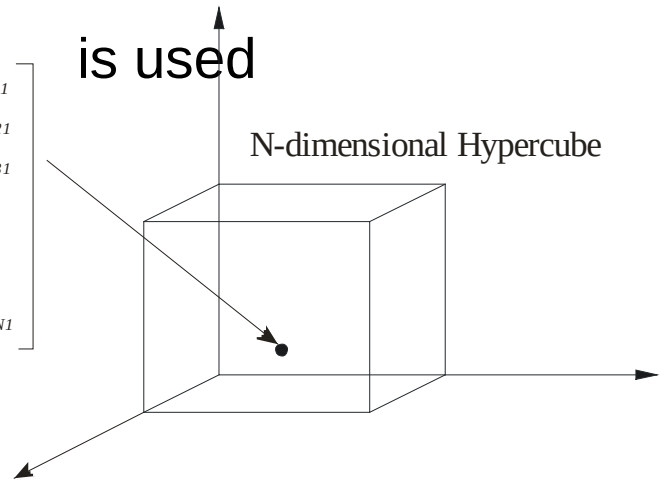
$$0 \le \Delta(A,B) \le 1;$$

$$\mu(A) = \sum_{i=1}^{n} m_A(y_i);$$

$A \cap B$ = Element - wise minima of A and B

$A \cup B$ = Element - wise maxima of A and B

$$M_{(N \times N)} = \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \\ \cdot \\ \cdot \\ \cdot \\ m_{N1} \end{bmatrix}$$



- Fuzzy Hamming Distance
  - *Fuzzy Cardinality CardA* of a fuzzy set A $= \sum_{i=1}^{n} \frac{x_i}{\mu_i}$

$$CardA = \sum_{i=0}^{n} \frac{i}{\mu_{CardA}(i)}$$

where $\mu_{CardA}(i) = min(\mu_i, (1 - \mu_{i+1}))$. μ_i denotes the i^{th} largest value of μ.

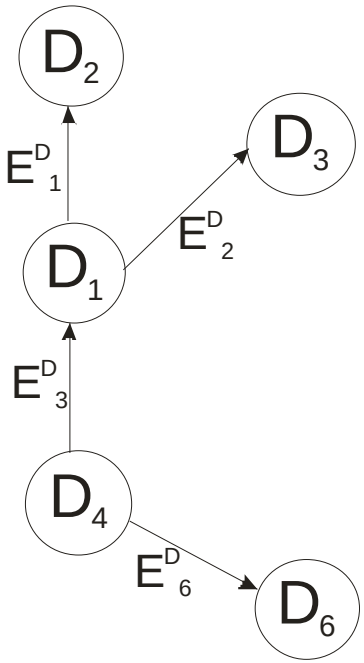
Compared using Silhouette validation technique and Mann-Whitney test and Hypercube Distance performed better.

Aggregating cluster elements

- Aggregate the matches to find the correlated nodes & edges in the formed clusters.
- The matches are combined using union and intersection operation

D_1	D_1
E^{D}_{2}	E^{D}_{1}
D_3	D_2
E^{D}_{3}	E^{D}_{3}
D_4	D_4
E^{D}_{6}	E^{D}_{6}
D_6	D_6

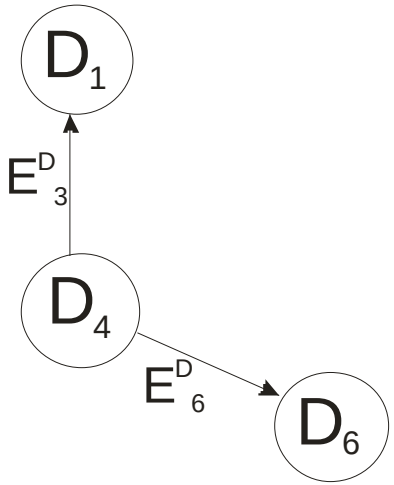
Diversification (Union)



Nodes	Edges
D_1	E^{D}_{1}
D_2	E^{D}_{2}
D_3	E^{D}_{3}
D_4	E^{D}_{6}
D_5	
D_6	

Intensification (Intersection)

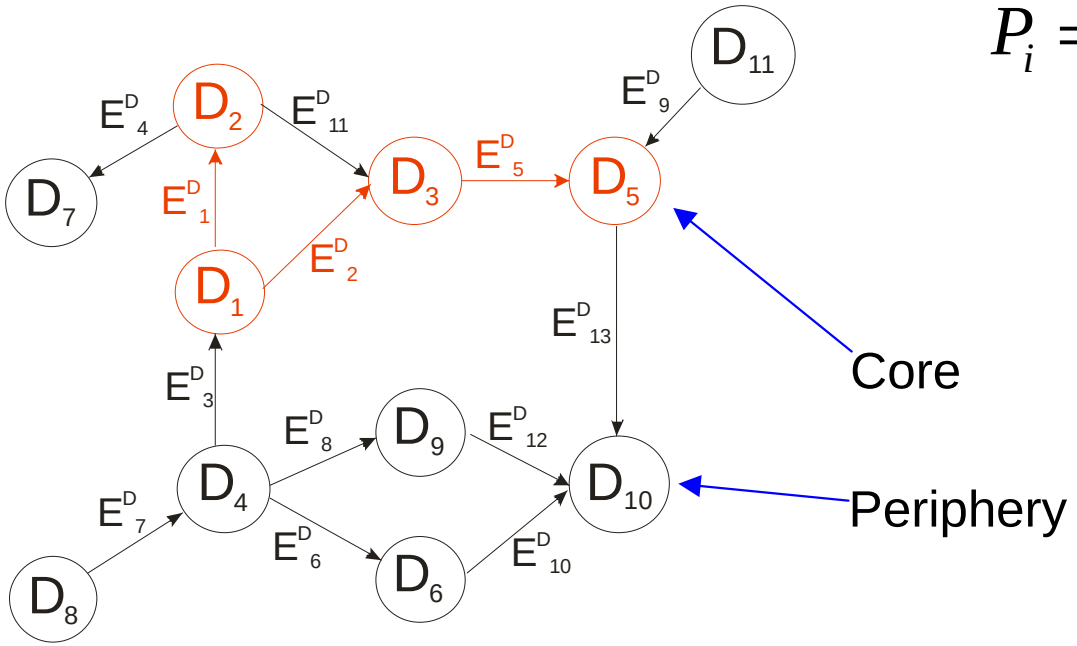
Nodes	Edges
D_1	E^{D}_{3}
D_4	E^{D}_{6}
D_6	

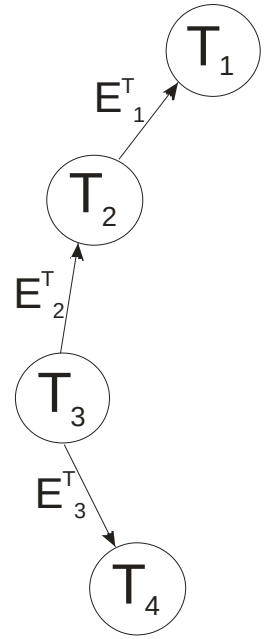


Neighborhood Information

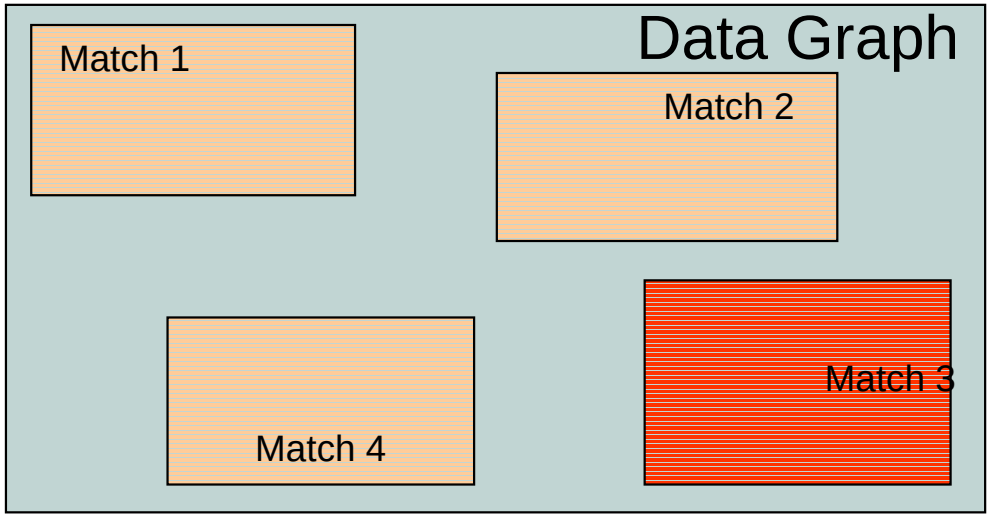
- P_i is the periphery node score
- N_T is the number of nodes in the match
- d_{ij} is the shortest path distance (*Floyd-Marshall*) between periphery node i and core node j

$$P_i = \frac{\sum_j d_{ij}}{N_T}$$

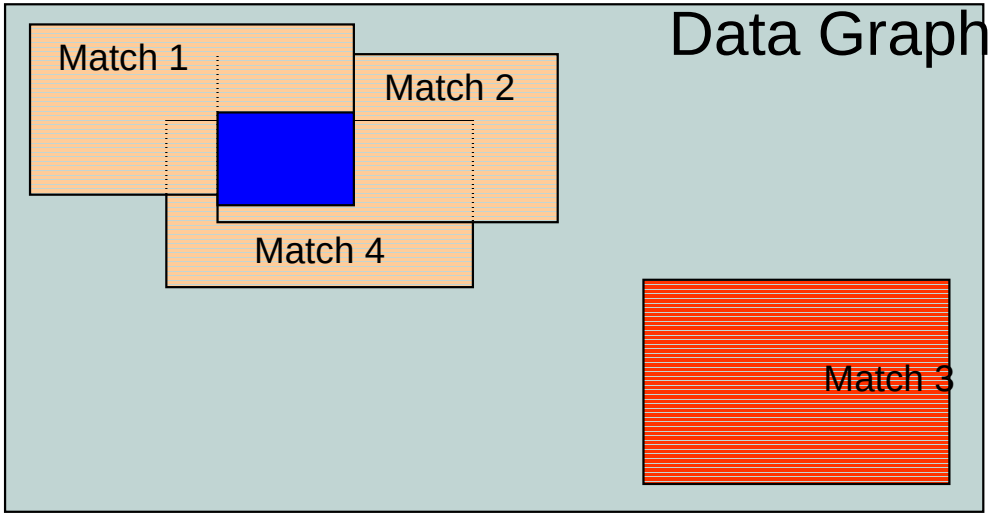




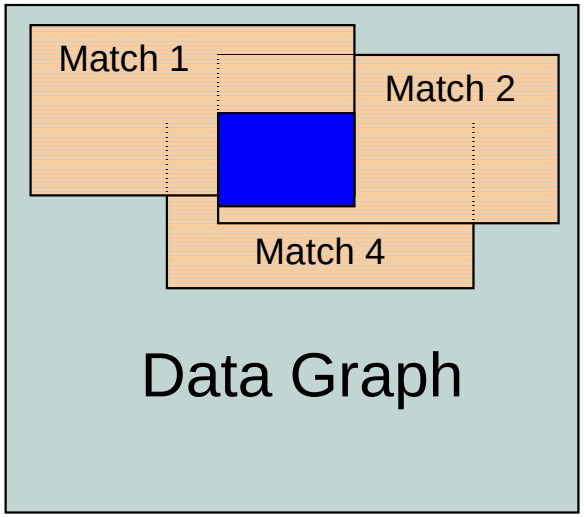
TruST Capabilities and Functionality



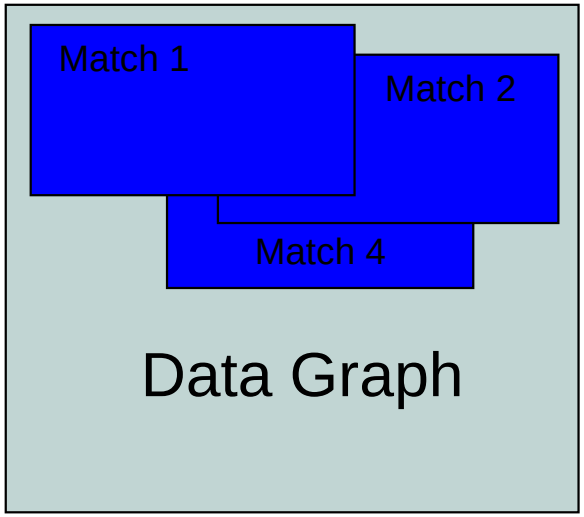
Ranking of Patterns of Interest



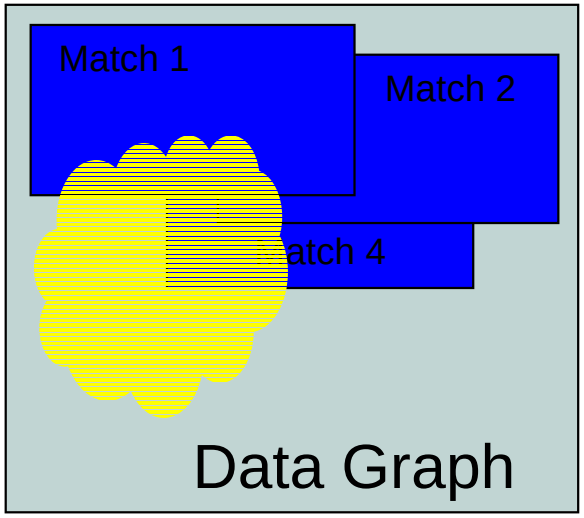
Clustering Patterns of Interest



Intensification

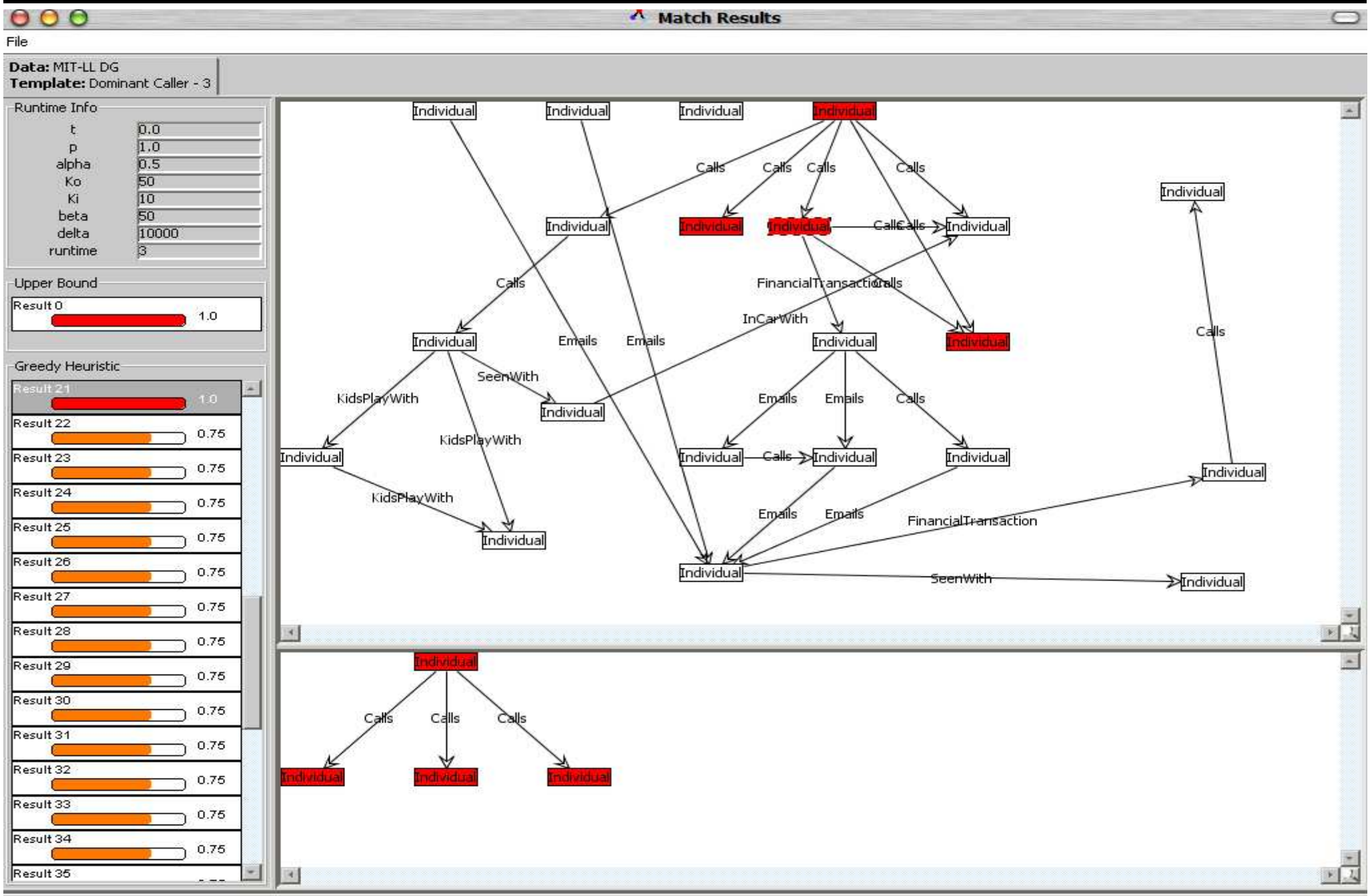


Diversification

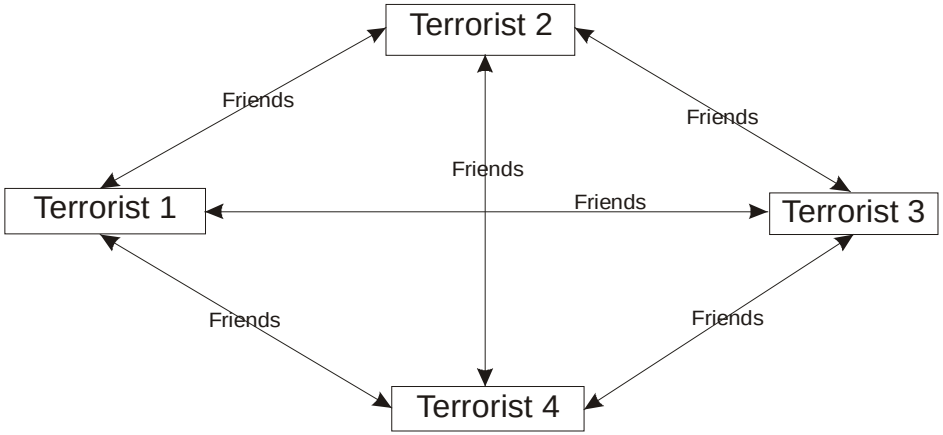


Pattern Discovery

TruST Implementation

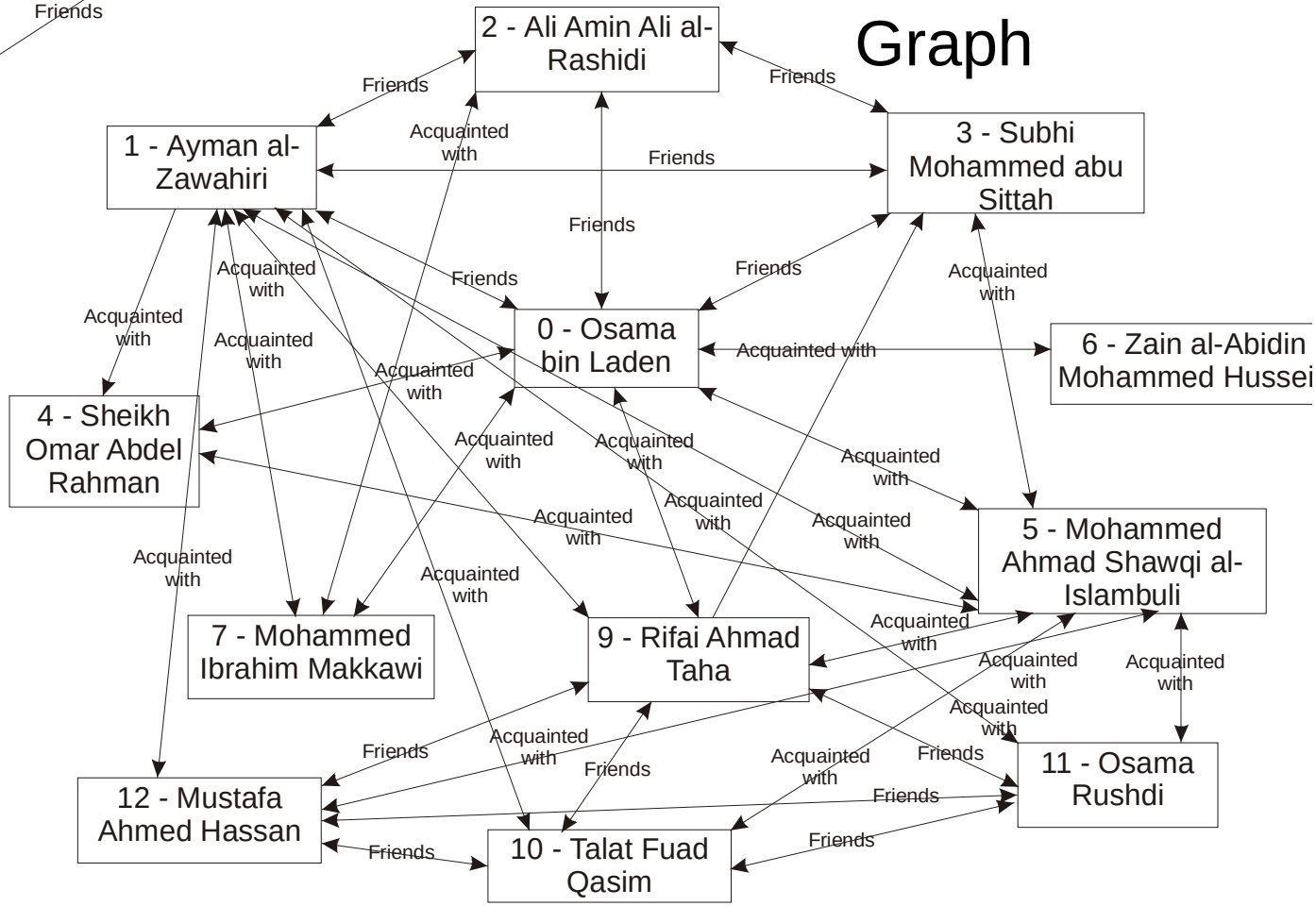


Example with Marc Sageman Data



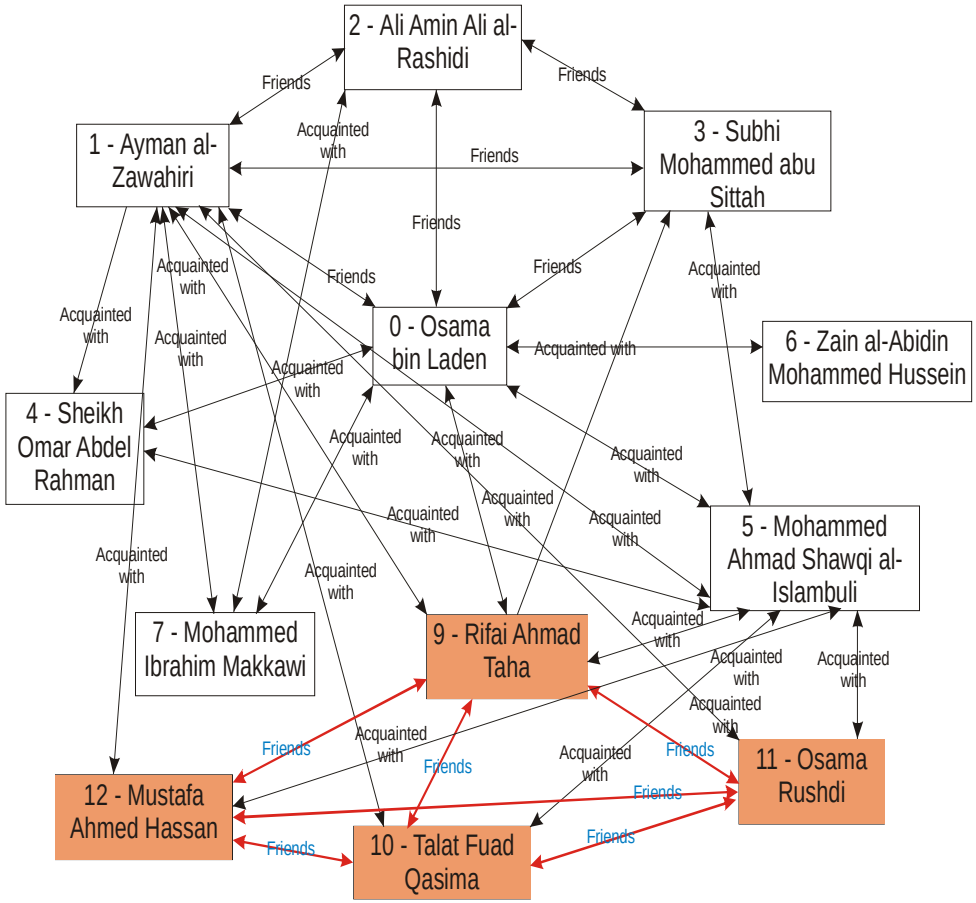
Template Graph

Data Graph

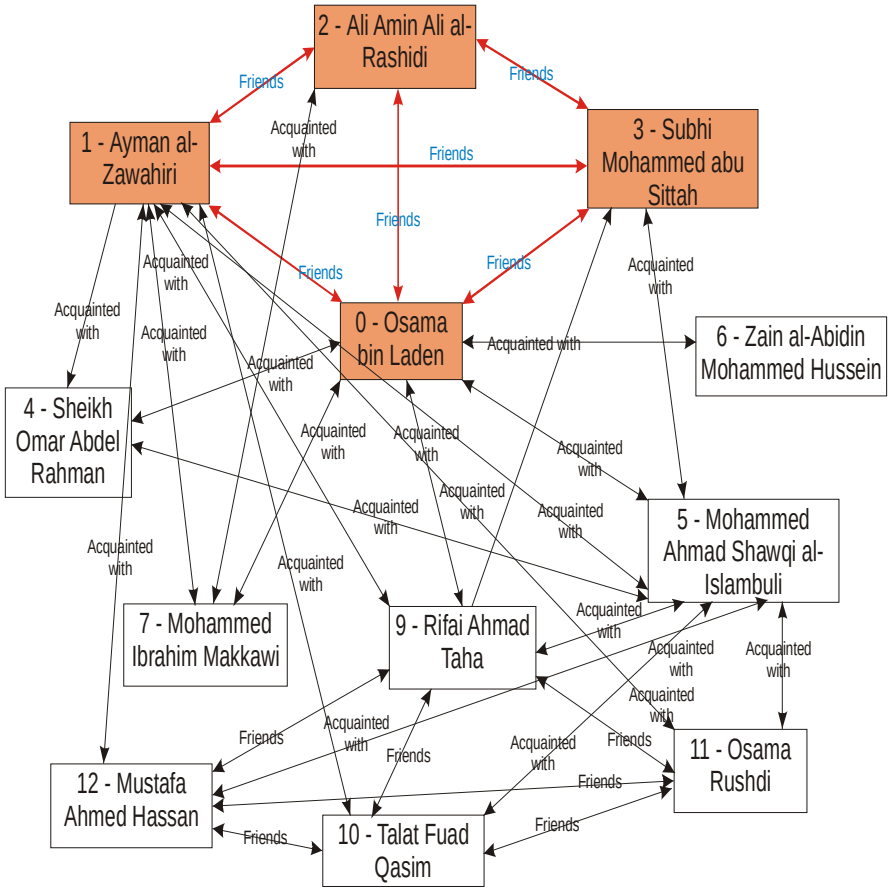


TruST matches

Match 1

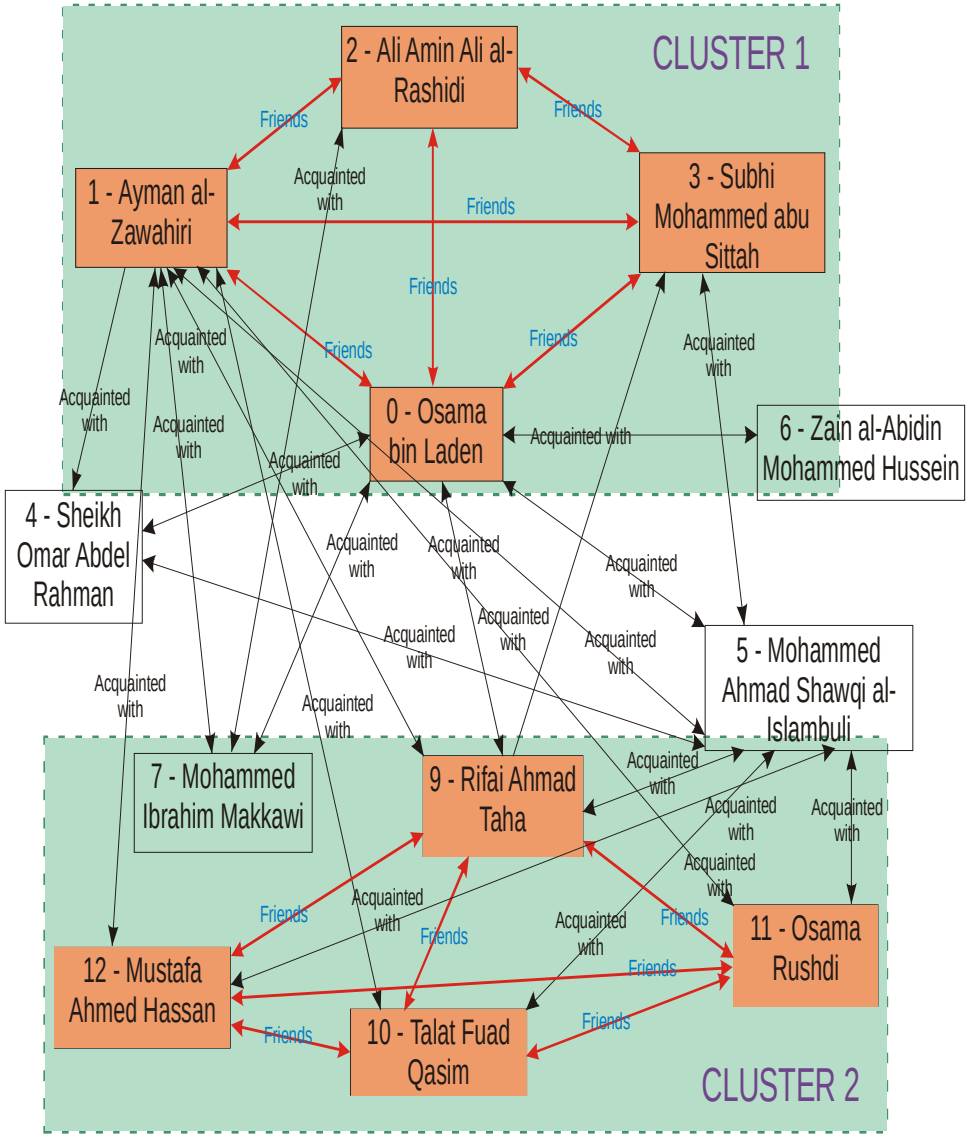


Match 2

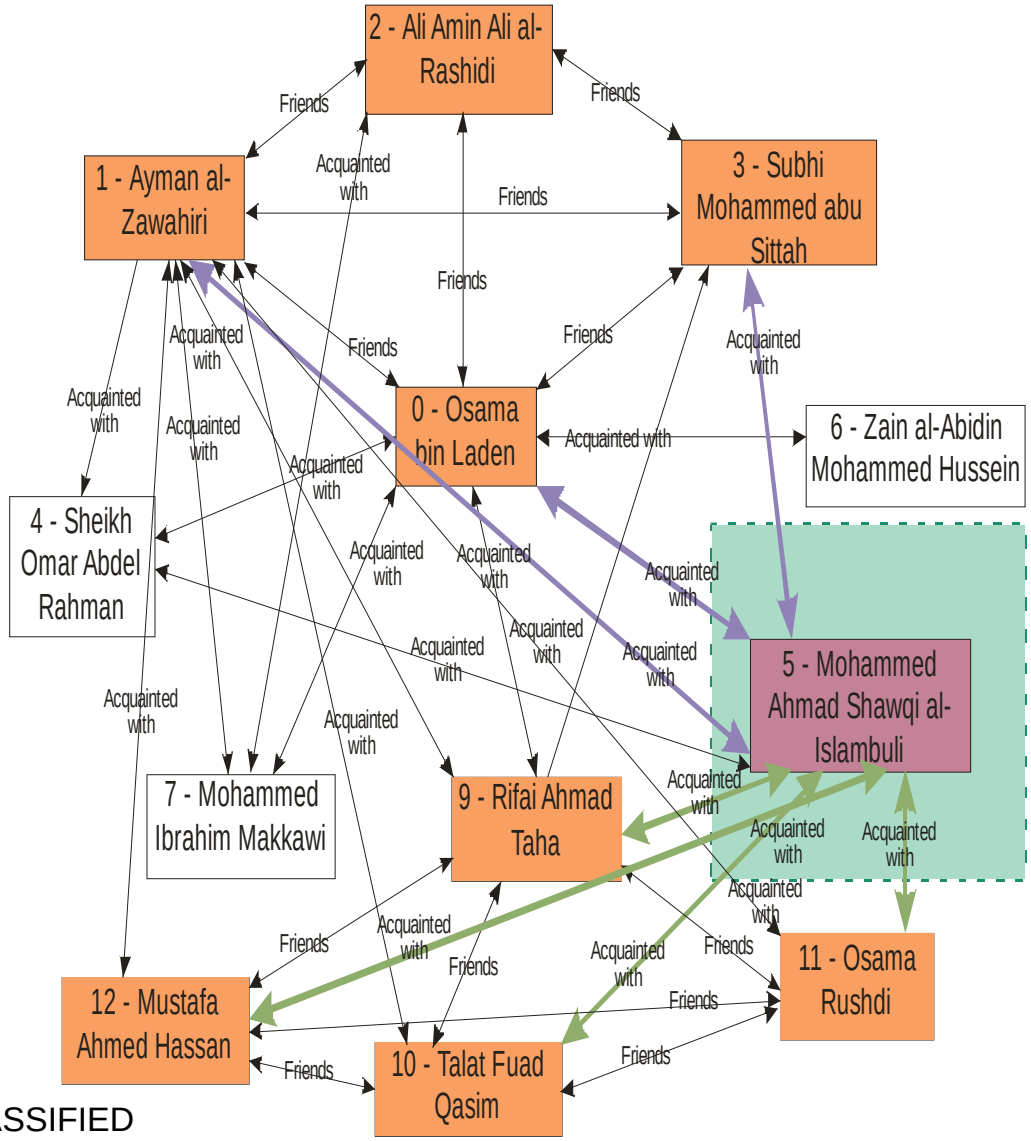


Clustering and Neighborhood Information

Hypercube Distance Clustering



Discovery of Neighborhood Information



Conclusion and Issues to Overcome

- Millions of possible Templates (Hypothesis) of interest
  - INformation Fusion Engine for Real Time Decision-making (INFERD)
- Ad-hoc choice of TruST parameters
  - We have done design of experiments to bound the best choice
  - We are working on trying to Characterize problem structure
- Graph Matching requires initial similarity values
  - We are working on automated process using the semantic features of the nodes and edges (Conceptual Spaces)
- Graph Matching could take to long for some domains
  - We span temporal decision-making process with INFERD and TruST
- Performance on multiple domains
  - Implemented in Asymmetric Warfare, Chem/Bio Warfare, Maritime Domain, Cyber Security, Sensor Management COA, etc.