Towards Optimization of Parallelized Mining of Subgraphs Sharing Common Items Using a Task-Parallel Language

Jing Xu[†1] Tasuku Hiraishi[†2] Shingo Okuno[†1*] Masahiro Yasugi[†3] Keiichiro Fukazawa[†1]

†1: Kyoto University (*: Presently with Fujitsu Limited) †2: Kyoto Tachibana University †3: Kyushu Institute of Technology

Graph Mining for Finding Subgraphs with Common Itemsets

Probrem definition

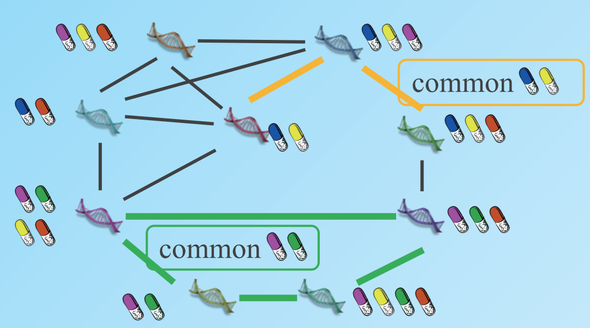
- Input: graph $G = (V, E)$, set of items I, items associated with each vertex $\mathfrak{I}(v) \in \mathfrak{P}(I)$, and threshold θ
- Output: all connected subgraphs $G' = (V', E')$ of G that satisfies the following conditions:

(1) $\left| \bigcap_{v \in V'} \mathfrak{I}(v) \right| \geq \theta$

(2) $\left| \bigcap_{v \in V' \cup \{v'\}} \mathfrak{I}(v) \right| < \left| \bigcap_{v \in V'} \mathfrak{I}(v) \right|$

for any $v'(\notin V')$connected to G'

- Application: gene network
  - Vertex: gene
  - Edge: protein-protein interaction
  - Item: reactional drugs

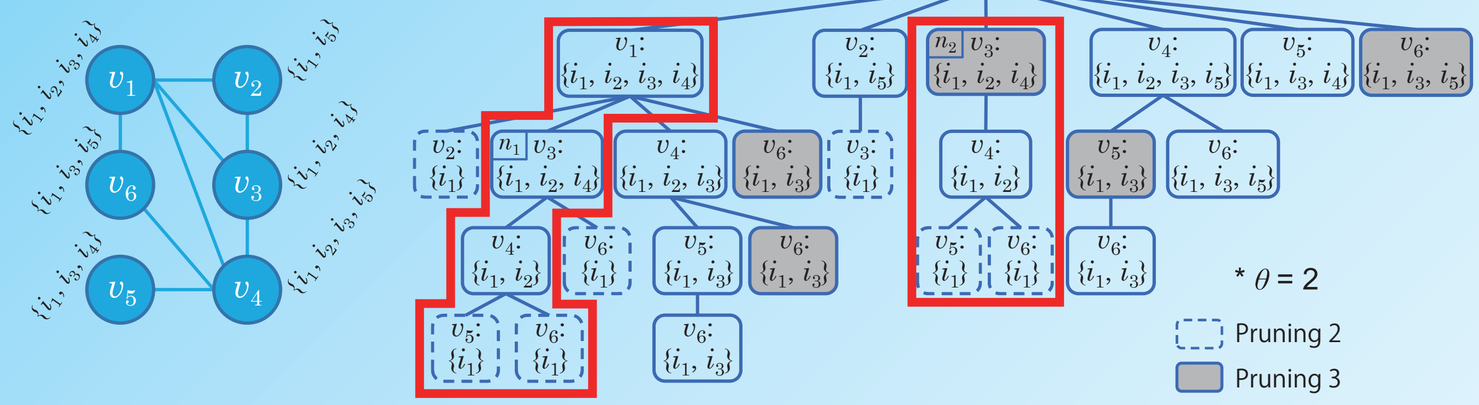


COPINE Algorithm [J. Sese et al., 2010]

- A depth-first tree algorithm for fining subgraphs with common itemsets employing the pruning for three kinds of subtrees:
  1. subgraph that has been already visited
  2. subgraph whose itemset is smaller than the threashold θ
  3. subgraph not being closed since one of its supergraphs has already been visited and their itemsets are identical

Parallel COPINE Algorithm [S. Okuno et al., 2017]

- In a parallel search (where a unique set of subtrees is assigned to each worker), a certain constraint is put on a worker for Pruning 3 [S. Okuno et al., JIP 2014].
- Problems in existing COPINE implementation:
  1. Right-to-Left (RTL) pruning is not allowed to avoid excessive pruning. Search space in parallel executions enlarges compared to sequential ones.
  2. Checking and updating an information table for pruning at every search step brings considerable overheads.



Itemset table for Pruning 3

- When adding a vertex to a current subgraph during a search, the common itemset of the resulting subgraph is added to the entry corresponding to the added vertex.
- On this occasion, if the table entry contains a super-itemset of the itemset being added, the search of the descendants of the current search tree node can be skipped.

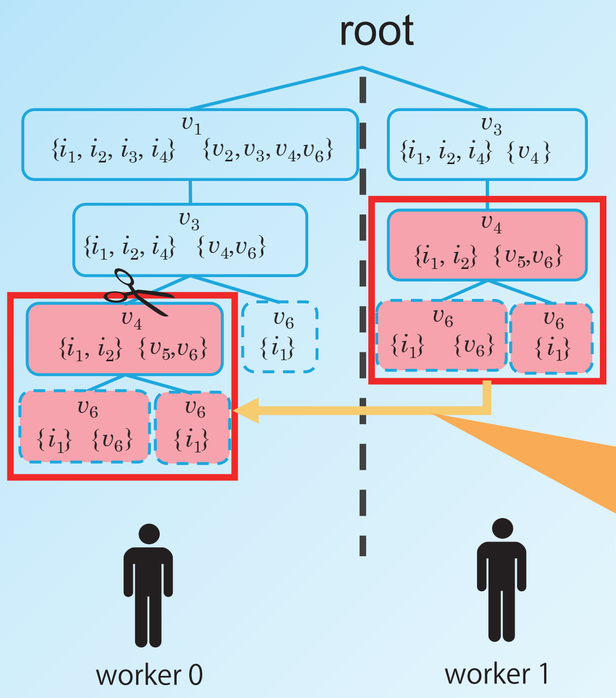
Table before traversing n_2

vertex	itemsets
v_1	$\{i_1, i_2, i_3, i_4\}$
v_2	$\{i_1, i_5\}$
v_3	$\{i_1, i_2, i_4\}$
v_4	$\{i_1, i_2, i_3\}$
v_5	$\{i_1, i_3\}$
v_6	$\{i_1, i_3\}$

Optimization of Parallelized COPINE using Task Parallel Language Tascell

Optimization

Right-to-Left (RTL) Pruning



A worker performs pruning "from right to left" when certain conditions are satisied, which is not allowed in the existing implementation.

Reducing the number of itemset table references

Given a threshold $d = 2$ for example, table access for Pruning 3 is performed only at search steps when the degree of the last added vertex is not less than 2.

Implementation

- We implemented these mechanisms by modifying the existing parallel COPINE implementation using the Tascell task-parallel language.

Performance evaluation

- Intel Xeon Broadwell 2.1GHz 18-core x 2
- Input: a real protein network, $\theta = 5$

  $|V| = 15227$, $|E| = 225458$, $|I| = 158$, avg. degree $= 29.2$, diameter $= 12$, each node has 9.42 items in average