**Fast Parallel Algorithms for Graph similarity and matching**

B.Tech 4th Year 1st SEM Project Report **Project Team:**

Akshay Agarwal - 127203

Rahul Iragavarapu - 127226

Y.V Praveen Kumar - 127264

**Project Guide:**

Dr.Ch.Sudhakar

Associate Professor, CSE Dept



**National Institute of Technology**

**Warangal (A.P.)-506004**

**2015-2016**

**CONTENTS:**

1. Abstract…………………………………………………………………….………….3
2. Introduction………………………………………………………………………….4
   1. Definition…................................................................................4
   2. Motivation…..............................................................................4
3. Related Work…………………………………………………………………….....7
4. Algorithms……………………………………………………………………….…….9
   1. NSD Algorithm……………………………………………………………………….10

4.1.1 Notations and format…………………………………………………….10

4.1.2 Algorithm ……………………………………………………………………..10

4.1.3 Explanation ………………………………………………….…………….…10

* 1. Bipartite marching Algorithm………………………………………………….11

4.2.1 Notations and format…………………………………………………….10

4.2.2 Algorithm ……………………………………………………………………..10

4.2.3 Explanation ………………………………………………….…………….…10

1. Work Completed…………………………………………………………………….17
2. Plan for Next Semester …………………………………………………………..17
3. References ………………………………………………………………………………18

# ABSTRACT

Many real world problems (examples vary from the web graph of documents, to a social network graph of friends, to road-map graphs of cities) either include graphs structured datasets or can be modelled into graph datasets, there is increasing need for efficient ways of analysing such datasets. These analyses include conservation, alignment, differentiation, and discrimination, among others. When defined on general graphs, these problems are considerably harder than their well-studied counterparts on sets and sequences.

This project discusses on alignment of graphs and focuses on finding “how similar a node in first graph is similar to a node in second graph?” or “what is the best match for each node in first graph to nodes in the second graph?” using parallel algorithms.

**INTRODUCTION**

**2.1 DEFINITIONS:**

Graph similarity and matching is about finding the most similar pair of vertices from the two given graphs or networks

Graph alignment problem is finding out how similar each vertex of first graph is with each vertex of second graph or what is best match for a vertex in first graph to a vertex in second graph.

The similarity of two nodes is defined by elemental and topological similarity, where elemental similarity represents how two nodes are individually similar and topological similarity represents how two nodes are similar with respect to their neighbourhood.

The solution to these consists of two steps

Step 1: to find the similarity matrix (X) of size A x B where A and B are number of vertices in G1 and G2 respectively based on their topological similarity and initialized with elemental similarity matrix (H).H matrix is based on measures like distance, or other vertex characteristics.

Step 2: to find for each vertex in G1 a vertex in G2 which is most similar using maximum bipartite matching algorithm.

**2.2 MOTIVATION :**

Over the last two decades, the field of graph mining has grown rapidly, not only because the number and the size of graphs has been growing exponentially (with billions of nodes and edges), but also because we want to extract much more complicated information from our graphs (not just evaluate static properties, but infer structure and make accurate predictions). This leads to challenges on several fronts - proposing meaningful metrics to capture different notions of structure, designing algorithms that can calculate these metrics, and finally finding approximations or heuristics to scale with graph size if the original algorithms are too slow.

Consider a use case where datasets are from mobile network, it is highly beneficial for an organisation to know its customers from their call routines, i.e. it can apply similarity algorithms on previous month and current month’s to know a customer’s relations just through data (each node a customer and call to other customer makes an edge from it)

Consider a use case where datasets are from protein, it is highly beneficial for a practitioner to know it’s functioning from their structural routines, i.e. similarity algorithms application on known protein to unknown protein data can shed some light on what it could be.

**RELATED WORK:**

There are many algorithms designed for the matching of graph vertices (among it’s vertices or other graphs vertices)

ISORANK:

The algorithm aims to maximize the overall match across all input networks. The intuition behind this algorithm is that a protein in one PPI network is a good match for a protein in another network if the former's neighbours are good matches for the latter's neighbours.

IsoRank iteration kernel:

X ← αB ̃ X A ̃T + (1-α)H

PAGERANK and HITS:

Page-Rank models a web surfer using random walks with occasional new traversals initiated from newly selected pages (nodes). The rank of a page is determined by computing the steady-state distribution of

the consequent random process. The HITS model, on the other hand, distinguishes between “hubs” and “authorities,” and computes their ranks in a mutually reinforcing manner . The motivation behind HITS is that a good authority should be pointed to by many good hubs, and that a good hub should point to many good authorities. Each page has both a hub score and an authority score. These scores are updated iteratively by the HITS algorithm. The HITS method is closely related to the Page-Rank algorithm —it finds similarity scores using two separate random walks on the corresponding bipartite graph of hubs and authorities, based on two different transition matrices. It follows naturally that the final scores are also the equilibrium distributions of the respective random walks .

**ALGORITHMS:**

**NSD Algorithm**:

Notations and Format:

A & B are the 2 input networks/graphs adjacency matrices

A is the normalized version of the matrix AT ; formally,

( A )ij = aji / Σ aji for nonzero rows j of A and zero otherwise.

n is the no.of iterations for topological similarity

alpha is the factor for considering elemental and topological similarity (0<=alpha<=1)

s is no.of pairs of singular value triplets for matrix H which is minimum of no.of rows in graphs

H is the elemental similarity matrix

U,V,Sigma are the singular value decompositions of H

(sigmai,ui,vi ) are the singular value triplets of H

where

sigmai is the ith diagonal element of Sigma

ui is the ith column of U

vi is the ith column of V

wi is sigmai\*ui

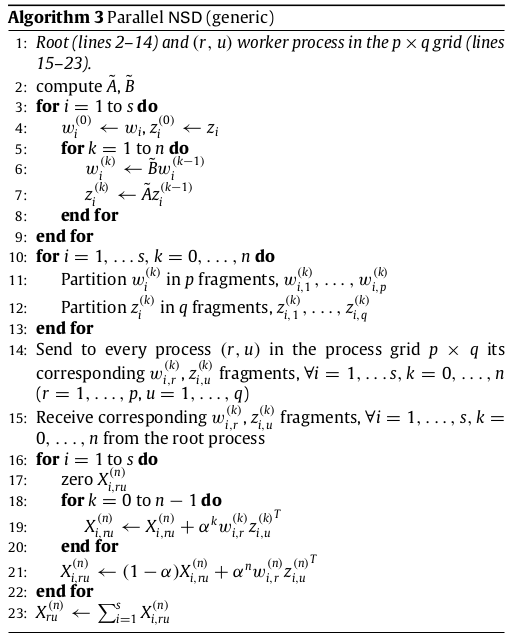
zi is vi

X(n) is the final similarity matrix

w(k)i,r is a partition of w(k)i in kth iteration

z(k)i,u is a partition of z(k)i in kth iteration

Algorithm:



Explanation:

line 2: Computation of tilde matrix for both graphs

tilde matrix gives how a particular vertex fits in the network

each vertex distributes it unity measure to all its adjacent vertices  
tilde matrix element aij contains how much jth vertex graph contributes to ith vertex

line 3-9: Calculate vectors iterates wi and zi for n iterations

line 10-13: Partition vectors wikinto p fragments and zikinto q fragments for all i’s and k’s

line 14: Send every fragment to corresponding process for all i’s and k’s

line 15: Receive corresponding fragments from  root process.

line 16-22: Calculate X(n)i,ru

line 23: Calculate X(n)ru, a part of similarity matrix.

**Bipartite Matching Algorithm(Auction based)**:

Notations and Format:

W is similarity matrix  
Va and Vb are the set of vertices of first and second graph respectively

na and nb are the number of vertices in first and second graph respectively

M is the current matching  
I is the set of unassigned vertices in first graph

p vector contains the price of all the vertices of second graph initialized to 0

is the update factor to ensure price is incremented in every iteration

Mlocal is the current matching in current process

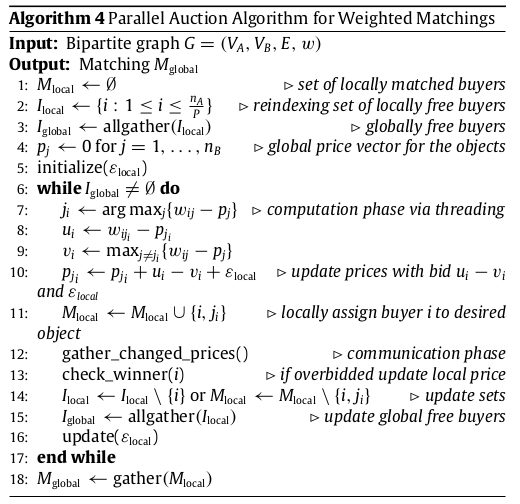
Mglobal is current matching from all processes

Ilocal is the set of unassigned vertices in first graph in current process

Iglobal is unassigned vertices in first graph from all processes

local is update factor to ensure price is incremented in every iteration in current process

Algorithm:



Explanation:

line 1-5: Initialization of matching set,unassigned vertices,price vectors and scaling factor

line 6,17: Loop segment

line 7-10: computation of the best matching vertex  based on similarity value and local price for that vertex and updation of prices

line 11: assign local buyer to vertex

line 12-13: gather all changed prices from all the processes and find the best buyer

line 14: update the local matching set and unassigned buyer set based on the best buyer,i.e when any other buyer has a higher bid on a particular vertex which has been locally assigned

line 15: update the global unassigned vertices set according to all local sets

line 16: update the scaling factor

line 18: gather all local matching sets to acquire global matching set

**Work Completed**

* Understood the working of serial and parallel algorithms in paper.
* Understood the Formulation from the problem statement.
* Implemented the matrix template library for random access operations of data in file and basic matrix operations.
* Implemented the serial versions of
  + NSD algorithm
  + Bipartite auction algorithm
* implemented the parallel versions of
  + NSD algorithm
  + Bipartite auction algorithm

With pthread library

**Percentage of Work Completed** –45%

**PLAN FOR THE NEXT SEMESTER**

* Implementation of algorithms by using Map-Reduce and OpenMP technologies.
* Deploying the problem statement code**.**
* Extension to current work by adding a new constraint.
* Analysing efficiency and correctness by adding a new constraint and comparing with existing problem.

**REFERENCES**

1. Kollias, Georgios; Sathe, Madan; Schenk, Olaf; and Grama, Ananth, "Fast Parallel Algorithms for Graph Similarity and Matching"
2. Madan Sathe: “Parallel Graph Algorithms for Finding Weighted Matchings and Subgraphs in Computational Science”
3. Giorgos Kollias, Shahin Mohammadi, and Ananth Grama: “Network Similarity Decomposition (NSD): A Fast and Scalable Approach to Network Alignment”