
# Subgraph Isomorphism Graph Challenge
# - draft -

**Siddharth Samsi, Vijay Gadepally, Jeremy Kepner, Albert Reuther**

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

http://GraphChallenge.org

# Outline

- Introduction
- Data Sets
- Static Graph Isomorphism Challenge
- Metrics
- Summary

# Introduction

- Previous challenges in machine learning, High Performance Computing and visual analytics include
  - YOHO, MNIST, HPC Challenge, ImageNet, VAST
- GraphChallenge encourages community approaches, such as DARPA HIVE, to develop new solutions for analyzing graphs derived from social media, sensor feeds, and scientific data to enable relationships between events to be discovered as they unfold in the field
- GraphChallenge organizers will provide specifications, data sets, data generators, and serial implementations in various languages
- GraphChallenge participants are encouraged to apply innovative hardware, software, and algorithm techniques to push the envelop of power efficiency, computational efficiency, and scale
- Submissions will be in the form of full conference write ups to IEEE HPEC which will allow participants to be evaluated on their complete solution

# Graph Challenge

- GraphChallenge seeks input from diverse communities to develop graph challenges that take the best of what has been learned from groundbreaking efforts such as GraphAnalysis, Graph500, FireHose, MiniTri, and GraphBLAS to create a new set of challenges to move the community forward

- Initial Graph Challenges
  - Static Graph Challenge: Sub-Graph Isomorphism
    - This challenge seeks to identify a given sub-graph in a larger graph
  - Streaming Graph Challenge: Stochastic Block Optimization
    - This challenge seeks to identify optimal blocks (or clusters) in a larger graph

# Static versus Streaming Mode

- Static graph processing
  - Given a large graph G
  - Evaluate $f(G)$
- Two classes of streaming: stateless and stateful
  - Stateless: process data g as it goes by $f(g)$
  - Stateful: add data g to a larger corpus G and then process corpus $f(G + g)$
- Graph processing is often in the stateful category
  - Easy to simulate by partitioning G into streaming pieces g

# Labels and Filtering

- Filtering on edge/vertex labels is often used when available to reduce the search space
  - Filtering can be applied at initialization, during intermediate steps, or at the vertex level
  - Very problem dependent
- Some Graph Challenge data sets have labels and some data sets are unlabeled
  - Some participants will want to filter on labels
- Initial example implementations will work without labels
  - Labels can be used if they choose
- GraphChallenge is judged by a panel
  - Panel will value the variations that are submitted

# Outline

- Introduction
- Data Sets
- Static Graph Isomorphism Challenge
- Metrics
- Summary

# Publicly Available Datasets

## Initial Datasets in Blue

### Stanford Large Network Dataset Collection snap.stanford.edu/data

- Social networks (10 data sets up to 4.8M vertices & 69M edges)
  - Online social networks, edges represent interactions between people
- Networks with ground-truth (6 data sets up to 66M vertices & 1.8B edges, e.g. Friendster)
  - Ground-truth network communities in social and information networks
- Communication networks (3 data sets up to 2.3M vertices & 5M edges)
  - Email communication networks with edges representing communication
- Citation networks (3 data sets up to 3.7M vertices & 16M edges)
  - Nodes represent papers, edges represent citations
- Collaboration networks (5 data sets up to 23K vertices & 198K edges)
  - Nodes represent scientists, edges represent collaborations
- Web graphs (4 data sets up to 875K vertices & 5.1M edges)
  - Nodes represent webpages and edges are hyperlinks
- Amazon networks (5 data sets up to 548K vertices & 3.4M edges)
  - Nodes represent products and edges link commonly co-purchased products
- Internet networks (9 data sets up to 62K vertices & 147K edges)
  - Nodes represent computers and edges communication
- Road networks (3 data sets up to 1.9M vertices & 2.8M edges)
  - Nodes represent intersections and edges roads connecting the intersections
- Autonomous systems (5 data sets up to 26K vertices & 106K edges)
  - Graphs of the internet
- Signed networks (10 data sets up to 4.8M vertices & 69M edges)
  - Networks with positive and negative edges (friend/foe, trust/distrust)
- Location-based online social networks (2 data sets up to 198K vertices & 950K edges)
  - Social networks with geographic check-ins
- Wikipedia networks, articles, and metadata (7 data sets up to 3.5M vertices & 250M edges)
  - Talk, editing, voting, and article data from Wikipedia
- Twitter and Memetracker (4 data sets up to 96M vertices & 476M edges)
  - Memetracker phrases, links and Tweets
- Online communities (3 data sets up to 2.3M images)
  - Data from online communities such as Reddit and Flickr
- Online reviews (6 data sets up to 34M product reviews)
  - Data from online review systems such as BeerAdvocate and Amazon

# Publicly Available Datasets

## Initial Datasets in Blue

### AWS Public Data Sets aws.amazon.com/public-data-sets

- Astronomy (1 data set 180 GB)
  - Sloan Digital Sky Survey SQL MDF files
- Biology (4 data sets up to 200 TB)
  - Genome sequence data
- Climate (3 data sets size growing daily)
  - Satellite imagery data
- Economics (10 data sets up to 220 GB)
  - Census, transaction, and transportation data
- Encyclopedic (10 data sets up to 541 TB - Common Crawl Corpus)
  - Various online encyclopedia data
- Geographic (3 data sets up to 125 GB)
  - Street maps
- Mathematics (1 data sets 160 GB)
  - University of Florida Sparse Matrix Collection

### Graph500.org Data Generator

- Used to generate world's largest power law graphs
- Can be modeled with Kronecker Product of a Recursive MATrix (R-MAT): $G^{\otimes k} = G^{\otimes k-1} \otimes G$
  - Where "$\otimes$"denotes the Kronecker product of two matrices

### Yahoo! Webscope Datasets webscope.sandbox.yahoo.com

- Advertising and Market Data (4 data sets up to 3.7 GB)
  - Yahoo!'s auction-based platform for selling advertising space
- Competition Data (3 data sets up to 1.5 GB)
  - Data challenges run by Yahoo
- Computing Systems Data (5 data sets up to 8.8 GB)
  - Computer systems log data
- Graph and Social Data (3 data sets up to 5 GB)
  - Graph data from search, groups, and webpages
- Image Data (3 data sets up to 14 GB)
  - Flickr imagery and metadata
- Language Data (29 data sets up to 166 GB - Answers browsing behavior)
  - Wide range of question/answer data sets
- Ratings and Classification Data (10 data sets up to 83 GB — 1.5 TB dataset withdrawn)
  - Community preferences and data

**Tools to generate (at various scales and parameters) data sets will be provided as part of the challenge**




- R-MAT: A Recursive Model for Graph Mining, Chakrabarti, Zhan & Faloutsos (2004)
- Mathematically Tractable Graph Generation and Evolution, Leskovec, Chakrabarti, Kleinberg & Faloutsos, (2005)

# GraphChallenge Data Formats

Wikipedia Voting

Wikipedia Adminship

Road Network

Twitter Data

- **Public data is available in a variety of formats**
  - **Linked list, tab separated, labeled/unlabeled**
- **Requires parsing and standardization**
- **Proposed formats**
  - **Tab separated triples in ASCII file with labels removed**
  - **MMIO ASCII format: math.nist.gov/MatrixMarket**

**Amazon cloud services will be used to host particularly large data sets**



The Matrix Market Exchange Formats: Initial Design, Boisvert, Pozo & Remington (1996)

# Outline

- Introduction
- Data Sets
- Static Graph Isomorphism Challenge
- Metrics
- Summary

# Static Subgraph Isomorphism Problem

## Vertex-Based

- Given a sub-graph H and a larger graph G
- Is there a 1-to-1 mapping of the vertices in H to vertices in G such that every edge in H is also in G ?

## Array-Based

- Given sub-graph adjacency array **B** and a larger graph adjacency array **A**
- A subgraph isomorphism exists iff there is a permutation array **P** where

$$\mathbf{B} = \mathbf{P}^{\mathsf{T}} \mathbf{A} \mathbf{P}$$

**Bold uppercase** indicates a matrix, **Bold lowercase** indicates a vector, lowercase indicates scalars, sets, ...

# Scaling Observations

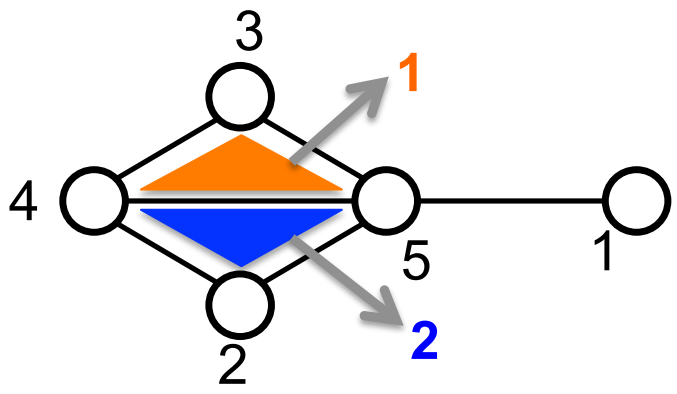
- **Two key parameters govern the complexity of subgraph isomorphism**
  - **Size of G and H**
  - **Nominal complexity is $|G|^{|H|}$**
- **Large G and H is may be too challenging**
- **Large G or H is more practical**

- **Small G and G and H are of same order is less interesting**
  - **Focus of program is large graphs**

- **Large G and small H (G is much larger than H) is more interesting**
  - **Example: H is a triangle (i.e., triangle finding algorithm)**
  - **Example: H is a k-truss (i.e., k-truss algorithm)**

# Triangle Counting

- **Triangles are the most basic non-trivial subgraph**
  - **Set of three mutually adjacent vertices in a graph**
- **Number of triangles in a graph is an important metric used in applications such as**
  - **Social network mining (e.g.: clustering coefficient calculation)**
  - **Link classification and recommendation**
  - **Cyber security**
  - **Intelligence**
  - **Functional biology**
  - **Spam detection**
- **Example of triangles in a Graph**
  - **Triangle 1 : Nodes 3,4,5**
  - **Triangle 2 : Nodes 2,4,5**

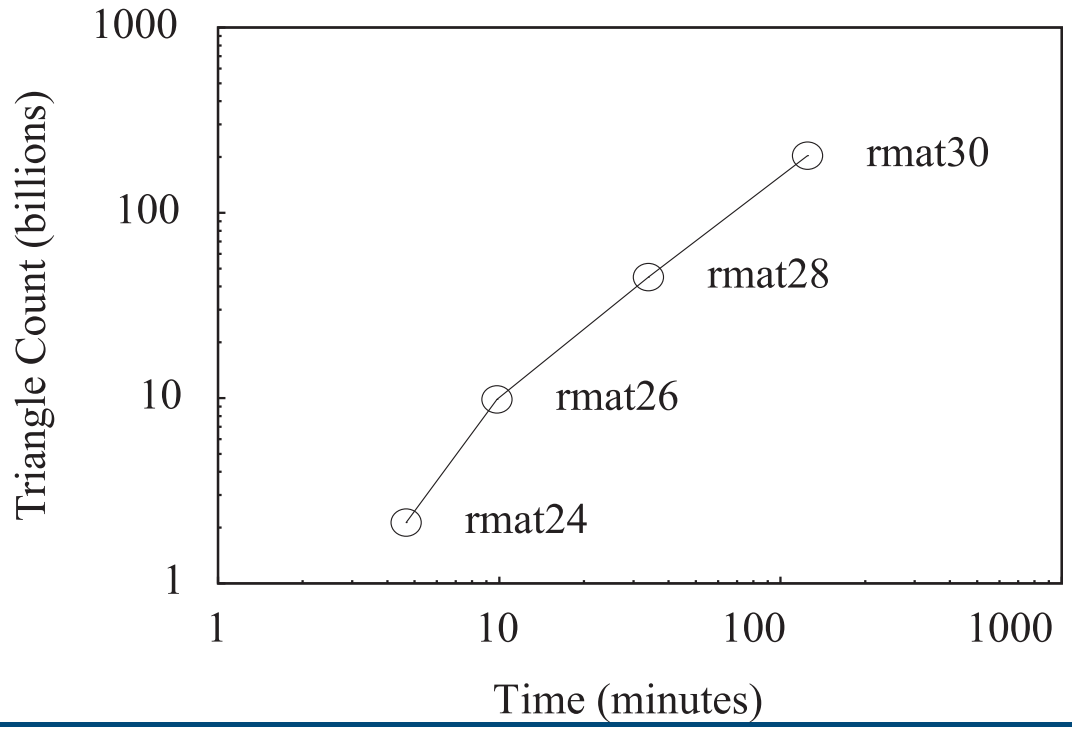




*Counting and Sampling Triangles from a Graph Stream*
A. Pavan, K. Tangwongsan, S. Tirthapura & Kun-Lung Wu, VLDB 2013

# Triangle Counting and Enumeration
## - Example Algorithm -

- **Lower complexity array approach**
- **Triangle enumeration has big I/O**
- **Triangle counting has minimal I/O**




Info Vis

*Article*

**Graphing trillions of triangles**

**Paul Burkhardt**

Information Visualization
1–10

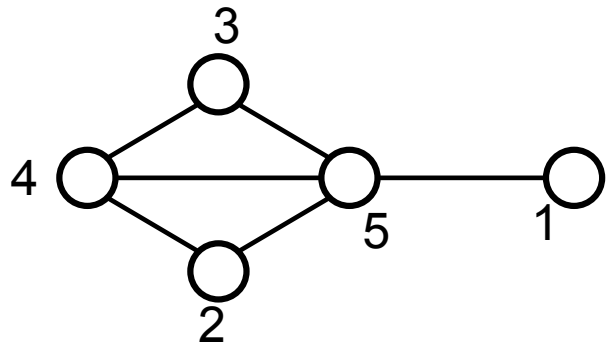
DOI: 10.1177/1473871616666393
ivi.sagepub.com
SAGE

**Theorem 1.** *Given $G$ and the Hadamard product $(A^2 \circ A)$, then $\Delta(v) = \frac{1}{2}\sum_v (A^2 \circ A)_v$ and $\Delta(G) = \frac{1}{6}\sum_{ij} (A^2 \circ A)_{ij}$.*

# Triangle Counting using miniTri
## - Example Algorithm -

- **Array based algorithm**
  - **Given a graph adjacency array A and incidence array E**

$$\mathbf{C} = \mathbf{AE}$$
$$n_T = \mathrm{nnz}(\mathbf{C})/3$$




- **Multiplication is overloaded such that**

$$\mathbf{C}(i,j) = \{i,x,y\} \text{ iff}$$
$$\mathbf{A}(i,x) = \mathbf{A}(i,y) = \mathbf{1} \text{ and } \mathbf{E}^T(x,j) = \mathbf{E}^T(y,j) = \mathbf{1}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad \mathbf{E}^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{C}=\mathbf{AE}^T = \begin{pmatrix} \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \{2,4,5\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \{3,4,5\} \\ \emptyset & \{4,2,5\} & \{4,3,5\} & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \{5,3,4\} & \{5,2,4\} & \emptyset \end{pmatrix}$$

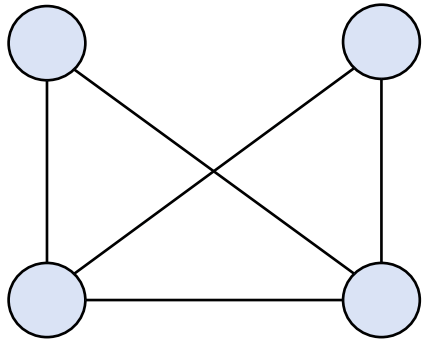
**miniTri represented in compact linear algebra-based operations**


*Kokkos/Qthreads Task-Parallel Approach to Linear Algebra Based Graph Analytics*
Wolf, Edwards & Olivier, IEEE HPEC 2016

# K-Truss

- A graph is a k-truss if each edge is part of at least k-2 triangles
- A generalization of a clique (a k-clique is a k-truss), ensuring a minimum level of connectivity within the graph

- Traditional technique to find a k-truss subgraph:
  - Compute the support for every edge
  - Remove any edges with support less than k-2 and update the list of edges
  - When all edges have support of at least k-2, we have a k-truss



Example 3-truss

# K Truss: Array Formulation
## - Example Algorithm -

- **If E is the unoriented incidence array (rows are edges and columns are vertices) of graph G, and A is the adjacency array**

- **If G is a k-truss, the following must be satisfied**
  - (**E A** == 2) **1** > (k – 2)

**Algorithm**

**R** = **E A**

**x** = find( (**R** == 2 ) **1** < k − 2 )

while **x**

$\mathbf{E}_x = \mathbf{E}(\mathbf{x}, :)$

$\mathbf{E} = \mathbf{E}(\mathbf{x}_c, :)$

$\mathbf{R} = \mathbf{E}(\mathbf{x}_c, :)\ \mathbf{A}$

$\mathbf{R} = \mathbf{R} - \mathbf{E}\ (\ \mathbf{E}_x\mathbf{E}_x^T - \mathrm{diag}(\mathbf{E}_x\mathbf{E}_x^T)\ )$

$\mathbf{x} = \mathrm{find}(\ (\mathbf{R} == 2\ )\ \mathbf{1} < k - 2\ )$


*Graphulo: Linear algebra graph kernels for NoSQL databases*, Gadepally, Bolewski, Hook, Hutchison, Miller & Kepner, IPDPS GABB 2015

# Outline

- Introduction
- Data Sets
- Static Graph Isomorphism Challenge
- Metrics
- Summary

# Metrics

- **Correctness**
  - **Number of triangles is exact for a given graph**
  - **Enumerating all k-trusses is exact for a given graph**

- **Performance**
  - **Total number of edges in graph (edges)**
  - **Execution time (seconds)**
  - **Rate (edges/second)**
  - **Energy (Watts)**
  - **Rate per energy (edges/second/Watt)**

# Summary

- Static sub-graph isomorphism graph challenge consists of
  - Triangle finding
  - K-Truss
- An implementation will perform these algorithms on provided data sets and report the given metrics
- A submission to the IEEE HPEC Graph Challenge consists of a conference style paper describing the approach, implementation, innovations, and results
- Both hardware and software should be described in solution
  - Innovative hardware solutions are of interest (in addition to best algorithm for hardware)
  - Special interest in performance for very large scale data sets (1-100 trillion edges)