A. Arleo, W. Didimo, G. Liotta, F. Montecchiani

A Distributed Multi-Level Force-directed algorithm

MULTI-GILA

24th International Symposium On
Graph Drawing And Network Visualization,
September 19-21, Athens, Greece

Objective

Efficient graph drawing algorithm (graphs with more than 10M edges)

- Using emerging distributed computing platforms
- Running on inexpensive *PaaS* environments

Apache Giraph

- Runs on Hadoop
- *BSP* model
- *Think-Like-A-Vertex*

Think Like a Vertex

Iterative approach

```
for(Edge e in edges){
  Get s & t
  coordinates from e
  Compute their
  distance Δ
  Update s position
  using Δ and the
  force function
}
```

- A single entity knows the entire topology
- Hard to scale

TLAV approach

```
for(Vertex v in
neighbours){
   Get v coordinates
   Compute the
   distance Δ
   Update v position
   using Δ and the
   force function
}
```

- Whole topology is unknown
- Easier to scale

Distributed Design Challenges

- What does TLAV mean?
  - Vertex perspective
  - Messages exchange with neighbors
- Limitations
  - Vertices store a small amount of data
  - Light communication load (i.e. light messages)
  - Global variables are permitted, but expensive
- Infrastructure overhead

Related Work

Mueller et al., 2006

- Complex network infrastructure
  - Computing and rendering nodes
- Multi monitor visualization
- Tested on graph up to 8k nodes

Tikhonova and Ma, 2008

- 260,385 edges graph in 40 minutes
  - PSC's BigBen Cray XT3 cluster, 32 processors

Related Work

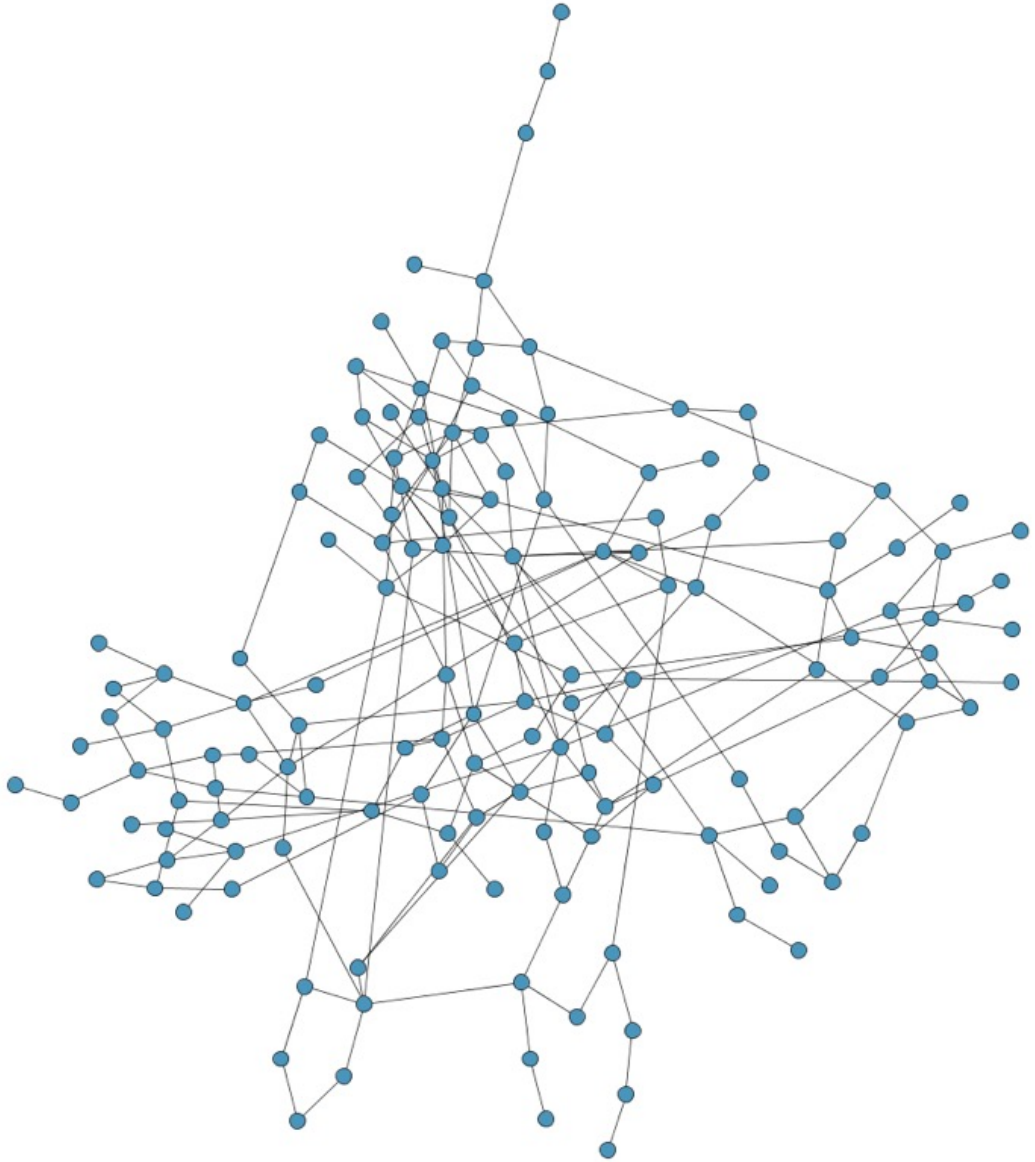
Hinge and Auber, 2015

- Force directed layout algorithm on GraphX
- Repulsive forces are approximated using centroids
- 8k vertices and 35k edges graph drawn in 5 hours
  - 16 machines, 24 cores and 48GB Ram each

Arleo et al., 2015

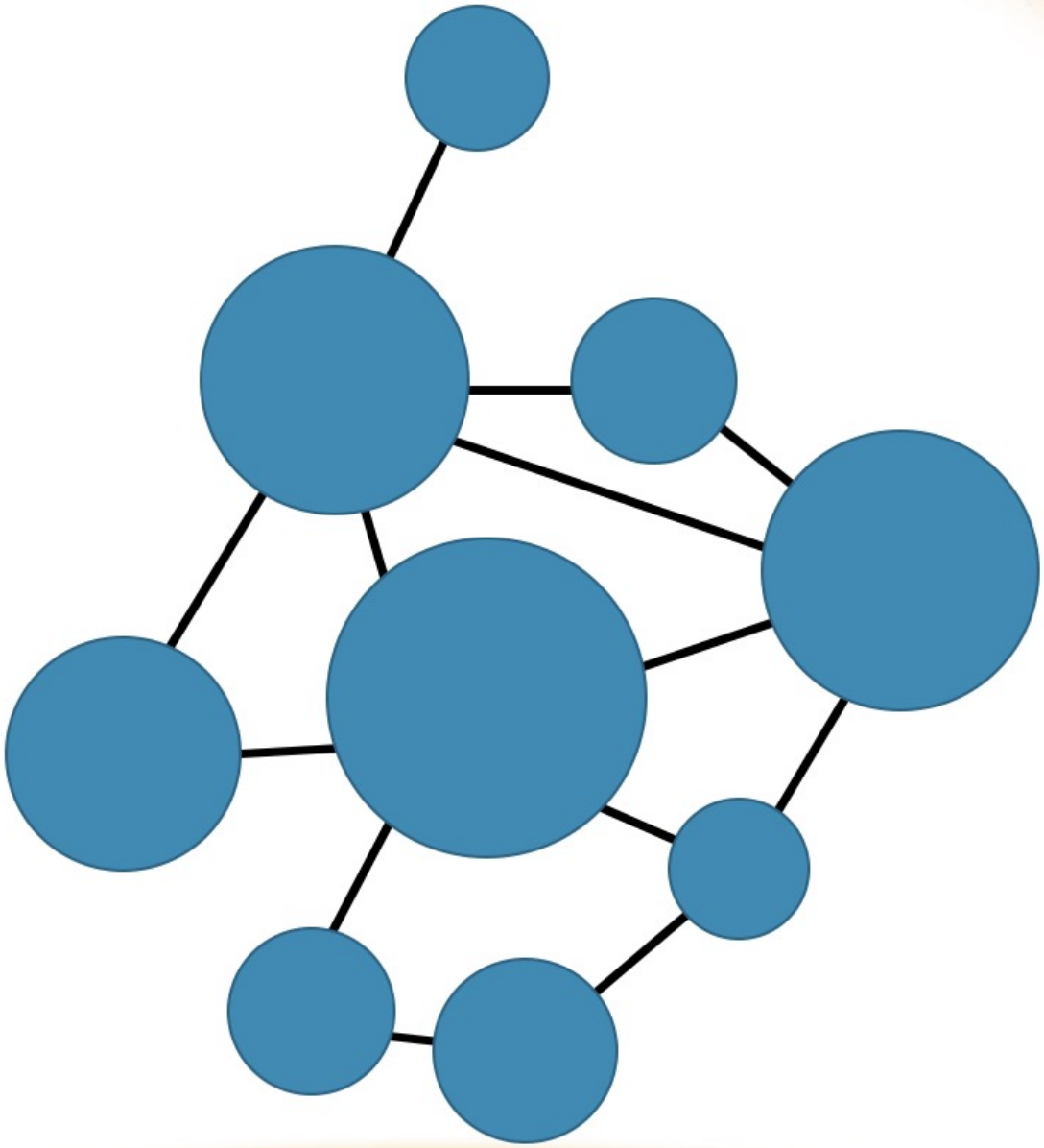
- "*Gila*" (former *Clint*)
- The first single-level FD layout algorithm on Giraph
- Performance was comparable to a centralized FD (Fruchterman-Reingold, 1991)
- 1.5M edges graph in less than an hour
  - 10 machines, 4 cores, 30GB Ram each

Multi-Level Layout



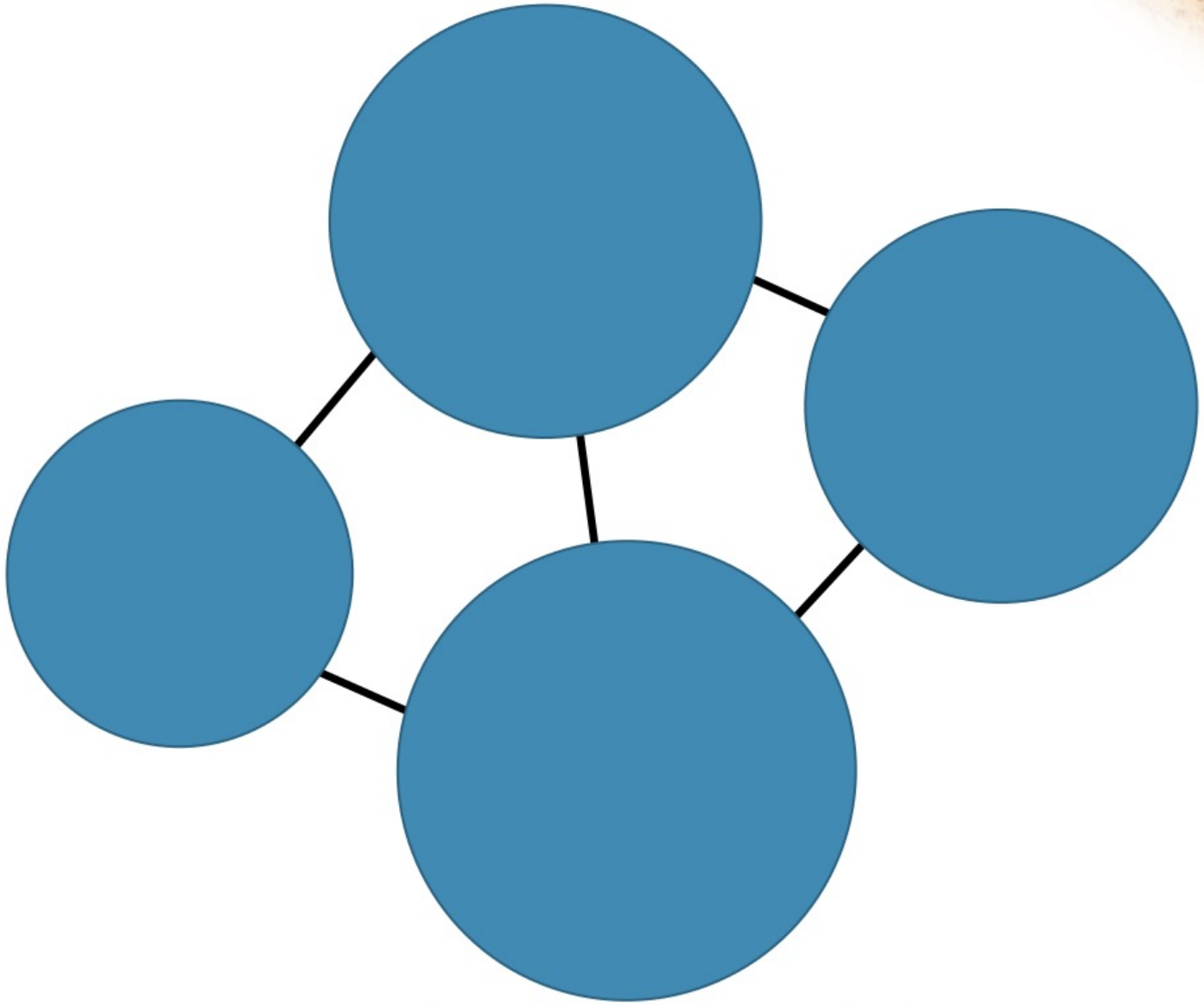
Coarsening

Multi-Level Layout



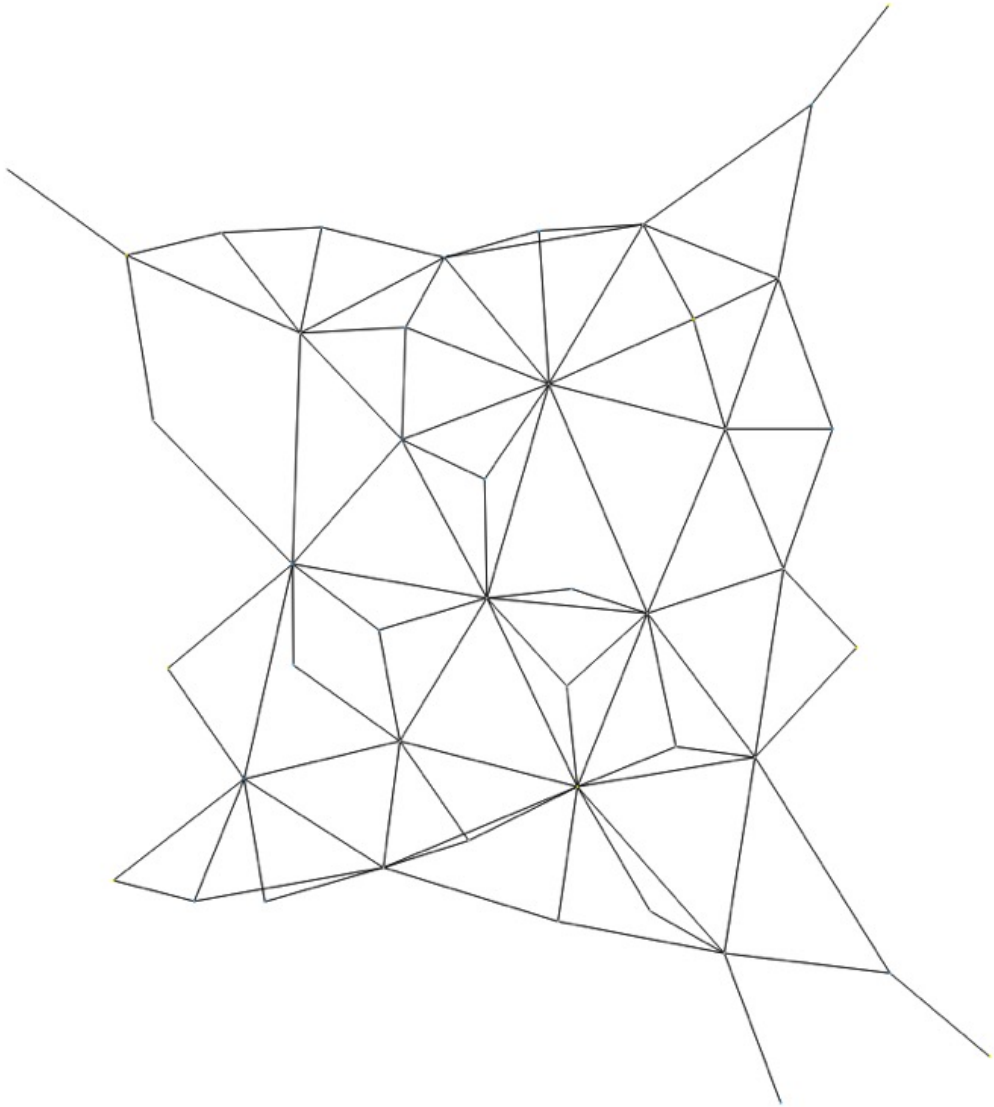
Coarsening

Multi-Level Layout

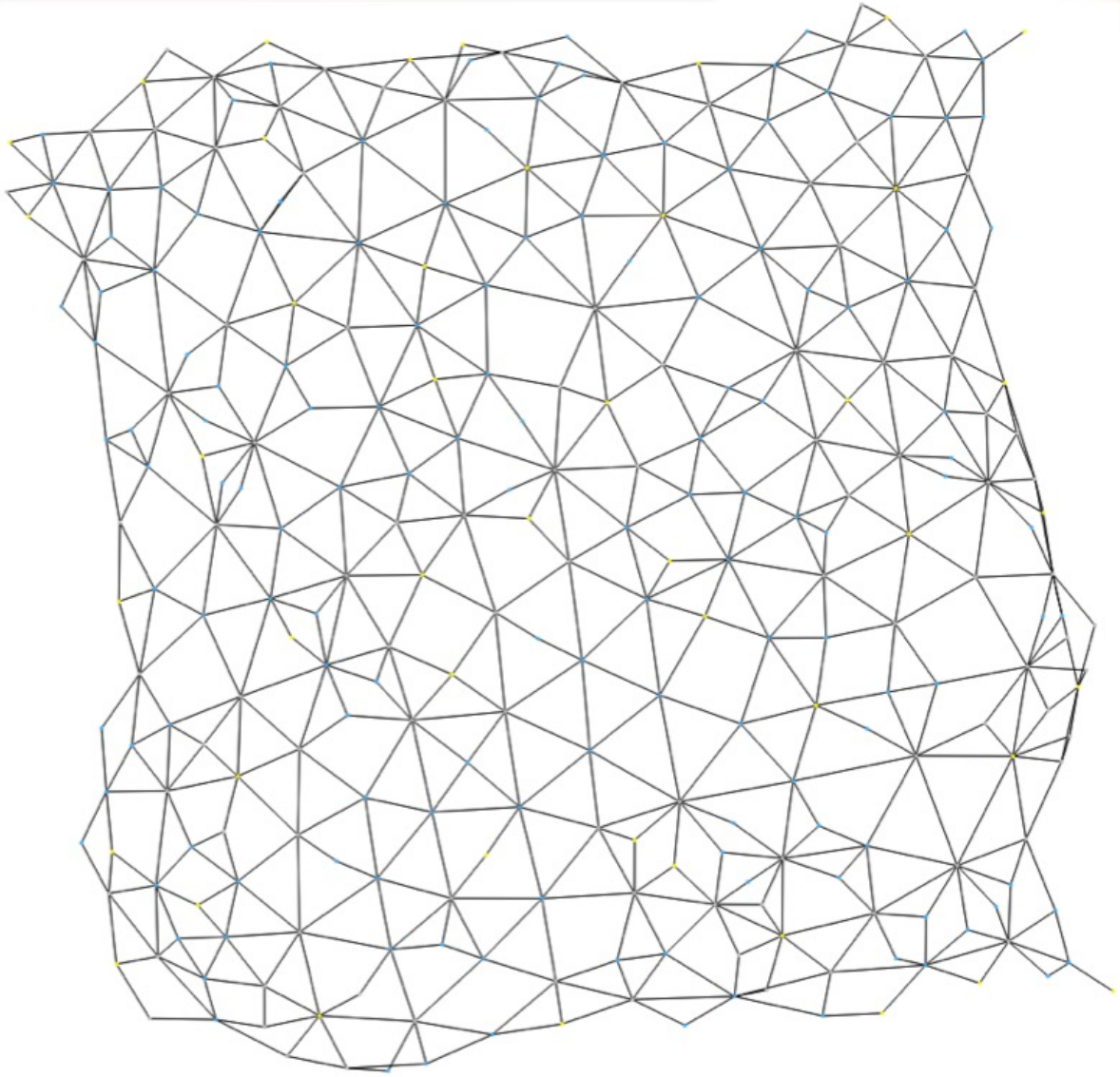


Coarsening

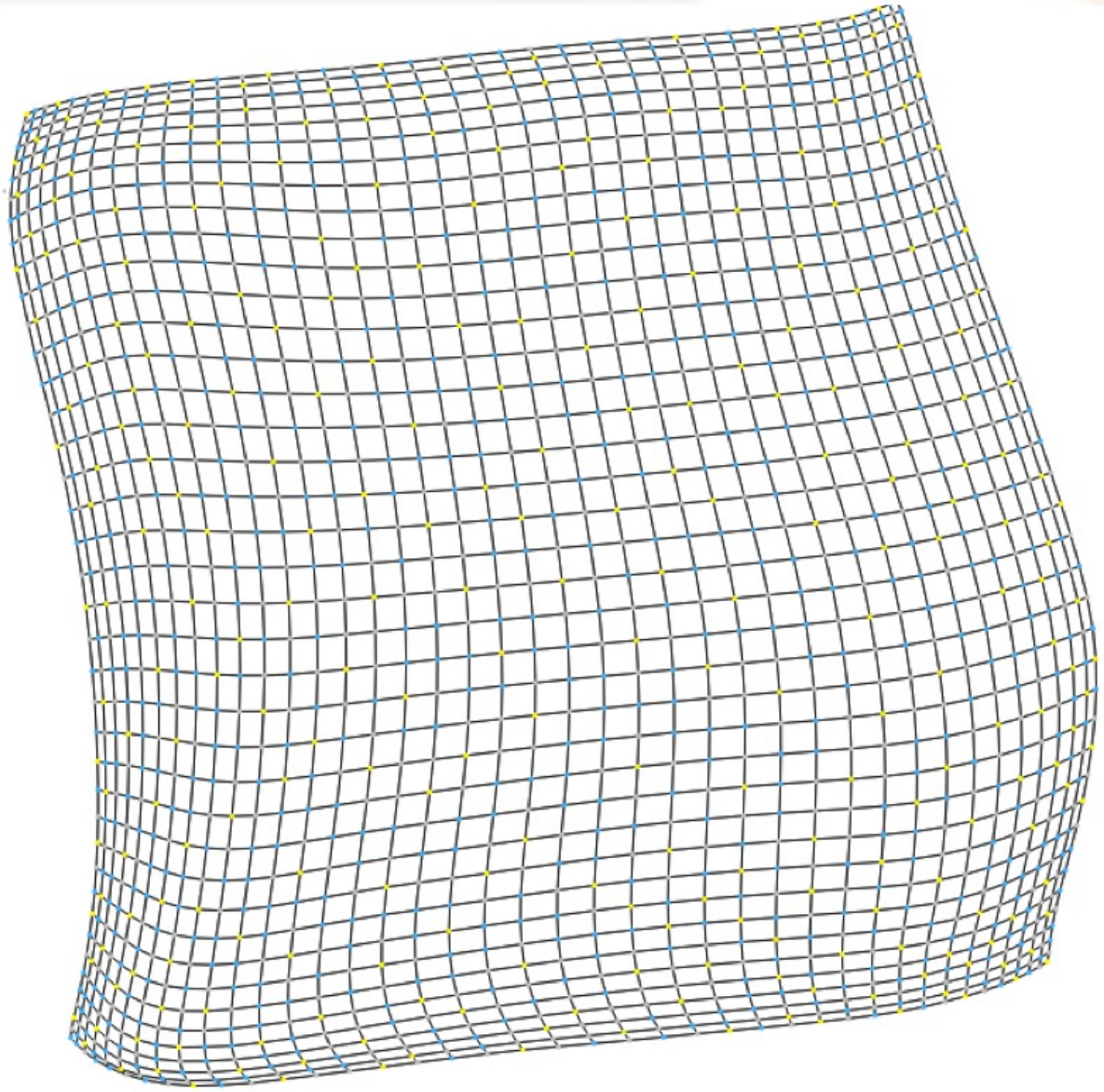
Multi-Level Layout



Multi-Level Layout



Multi-Level Layout



Placement & Layout

Our Contribution

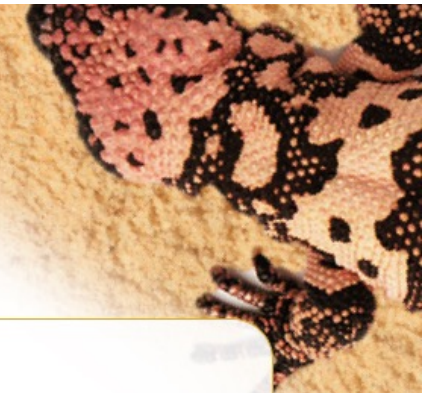
- First distributed multi-level force-directed layout algorithm
  - Implemented on *Giraph*
  - Single level layout provided by GILA
- Experimental evaluation
  - Small regular graphs
  - Medium sized graphs (up to 1.5M edges)
  - Large graphs (up to ~11M edges)

Algorithm Overview

Key-Ideas

- Coarse hierarchy generation inspired by FM^3 (Hachul-Junger, 2004)
  - Distributed Merger
  - Distributed Placer
  - Designed to have a low impact on total running time
- Dynamic single-level layout tuning
  - Coarse levels will have more accurate drawings
  - Allows us to scale to bigger graphs

Multi-Gila Pipeline



Pre-process

- Pruning
- Partitioning
- Component Discovery

Multi-Level layout

- Distributed Solar Merger
- Distributed Solar Placer
- Gila (single level layout)

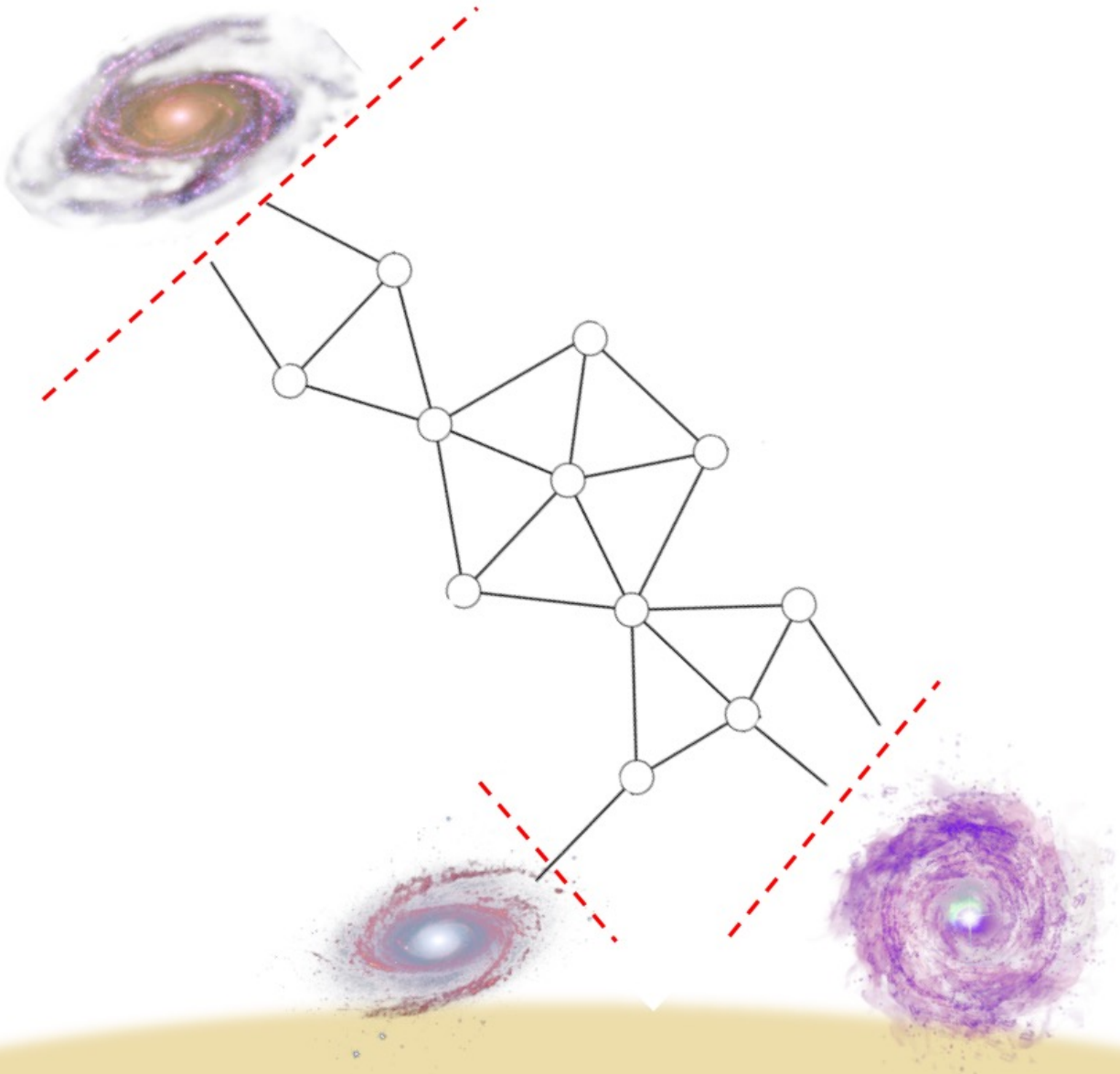
Post-Process

- Graph reconstruction
- Connected components arrangement

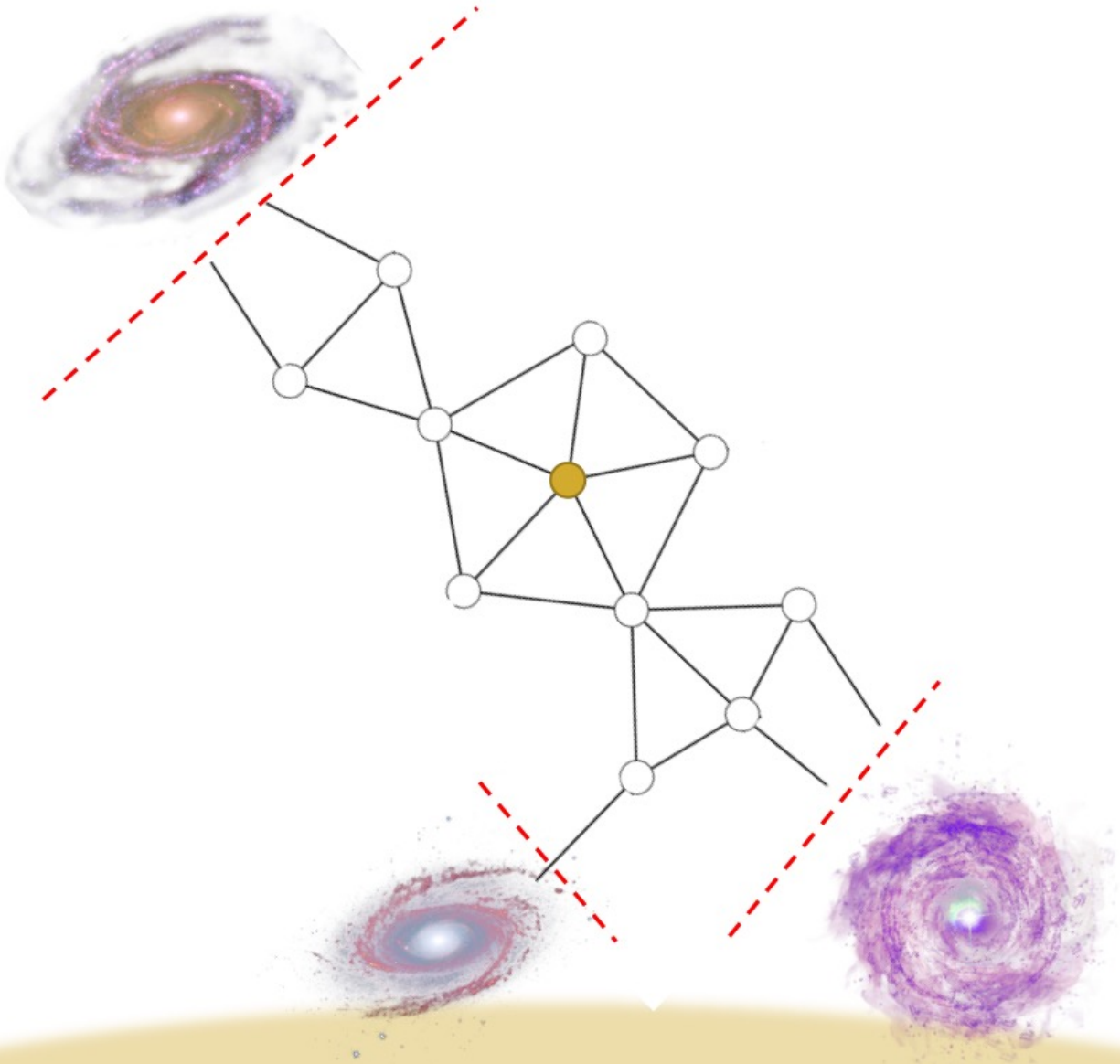
Coarsening Phase

- Partitioning of the input graph into subgraphs
- These subgraphs are called *Galaxies*:
  - There is a single *Sun*
  - *Sun's* neighbors are called *Planets*
  - If any, *Planets'* neighbors are called *Moons*
  - Max GTD between any two of its nodes: 4
- *Galaxies* are collapsed onto their sun

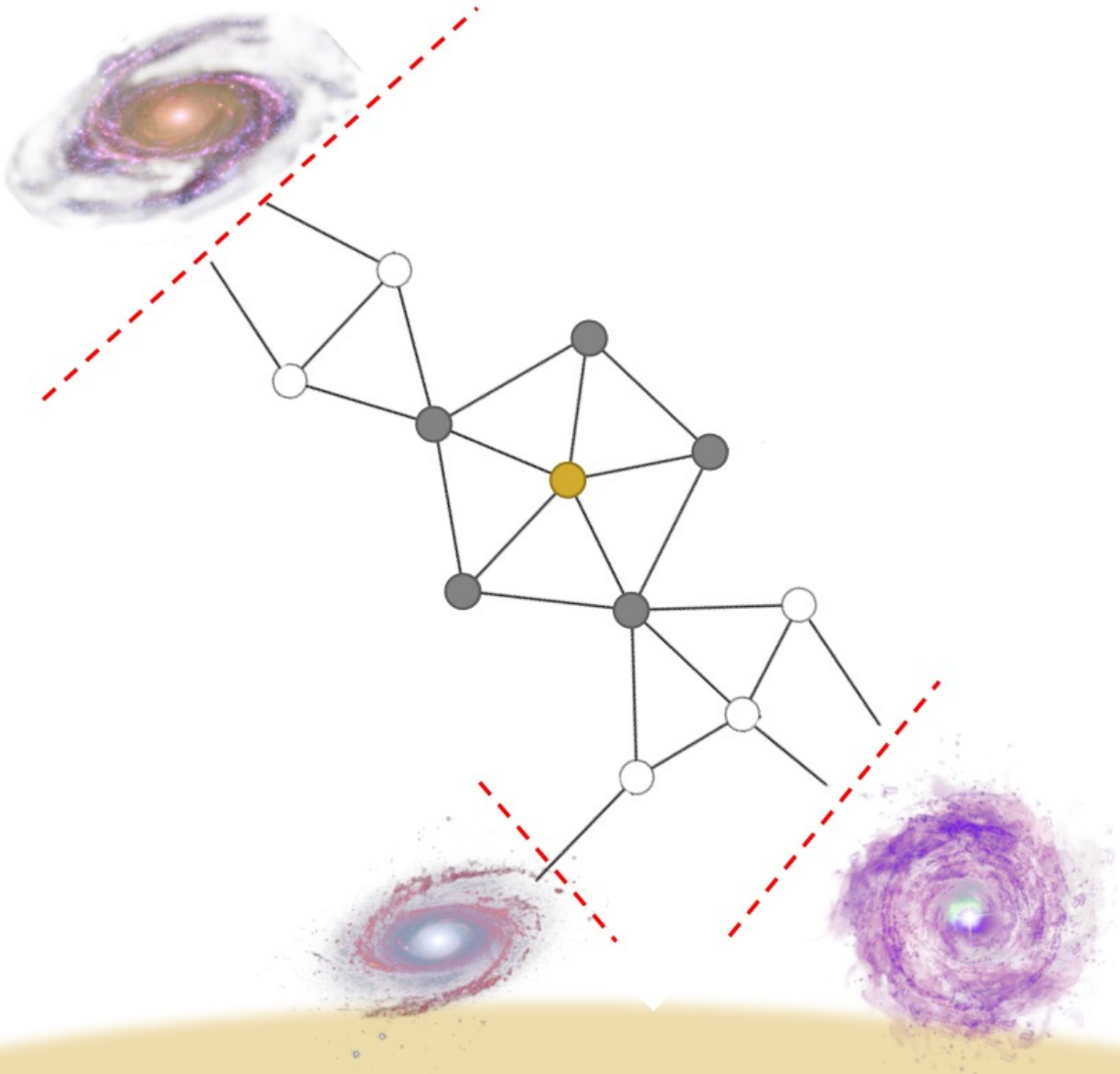
FM³ Solar Merger (FSM)



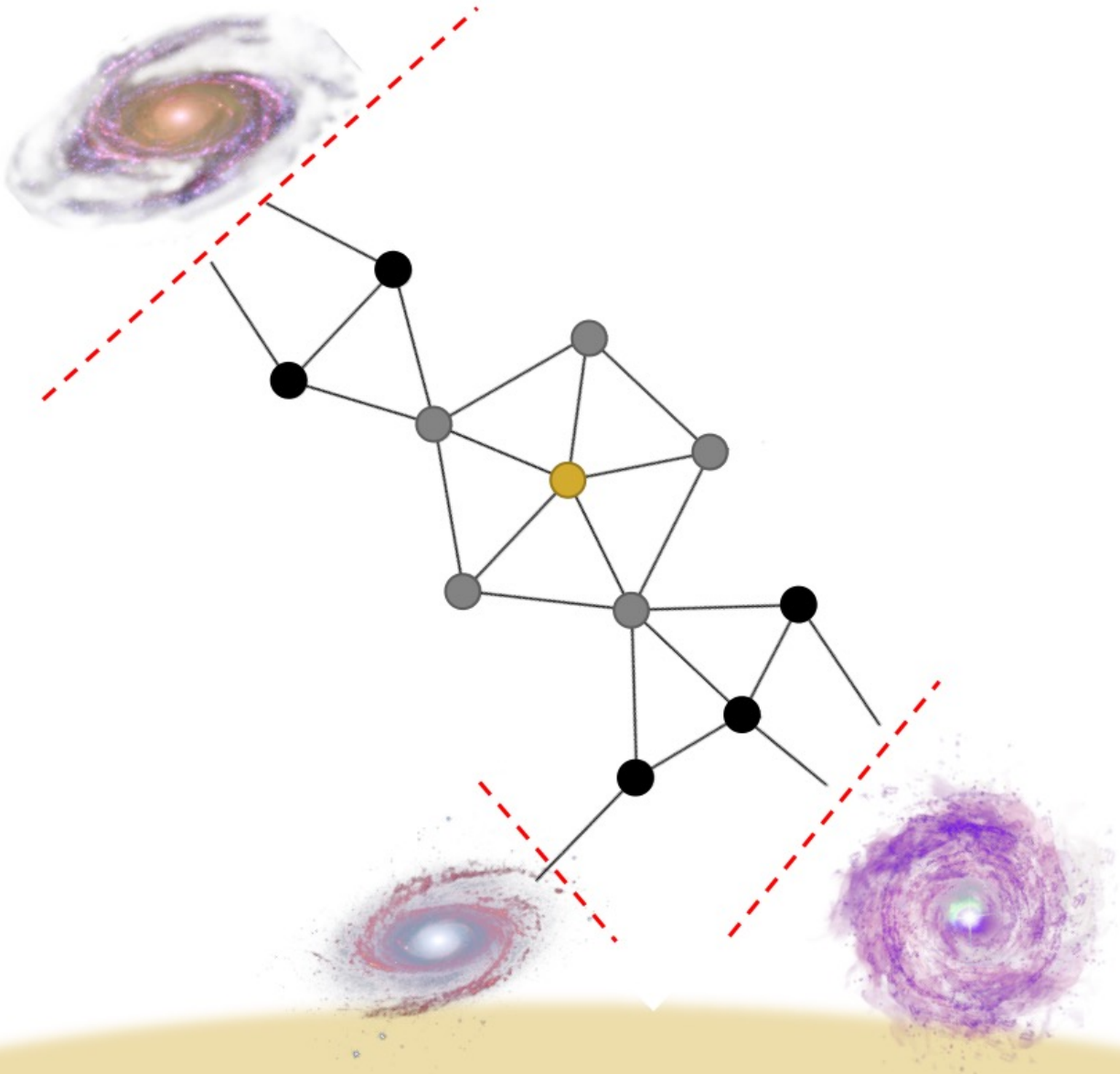
FM³ Solar Merger (FSM)



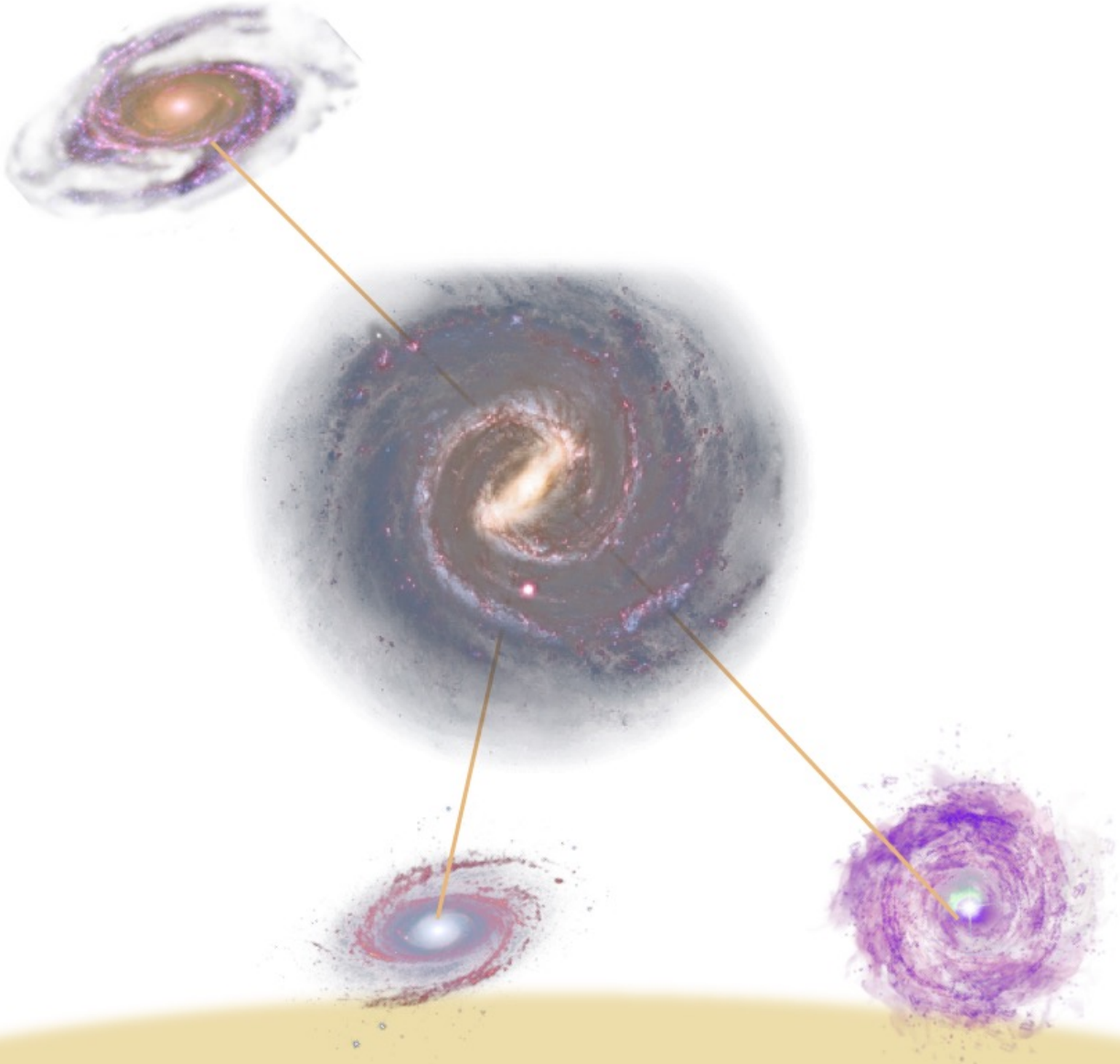
FM³ Solar Merger (FSM)



FM³ Solar Merger (FSM)



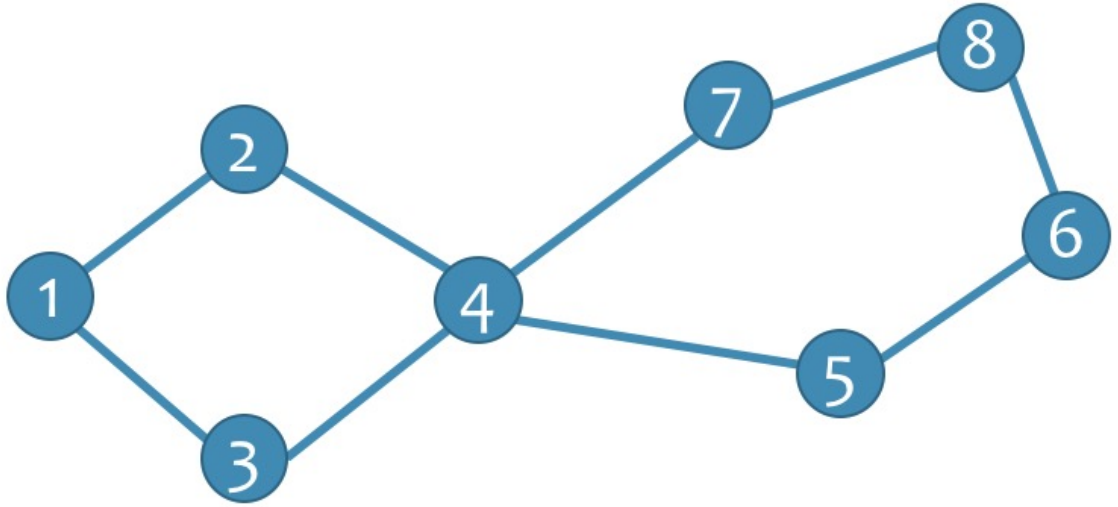
FM³ Solar Merger (FSM)



Challenges

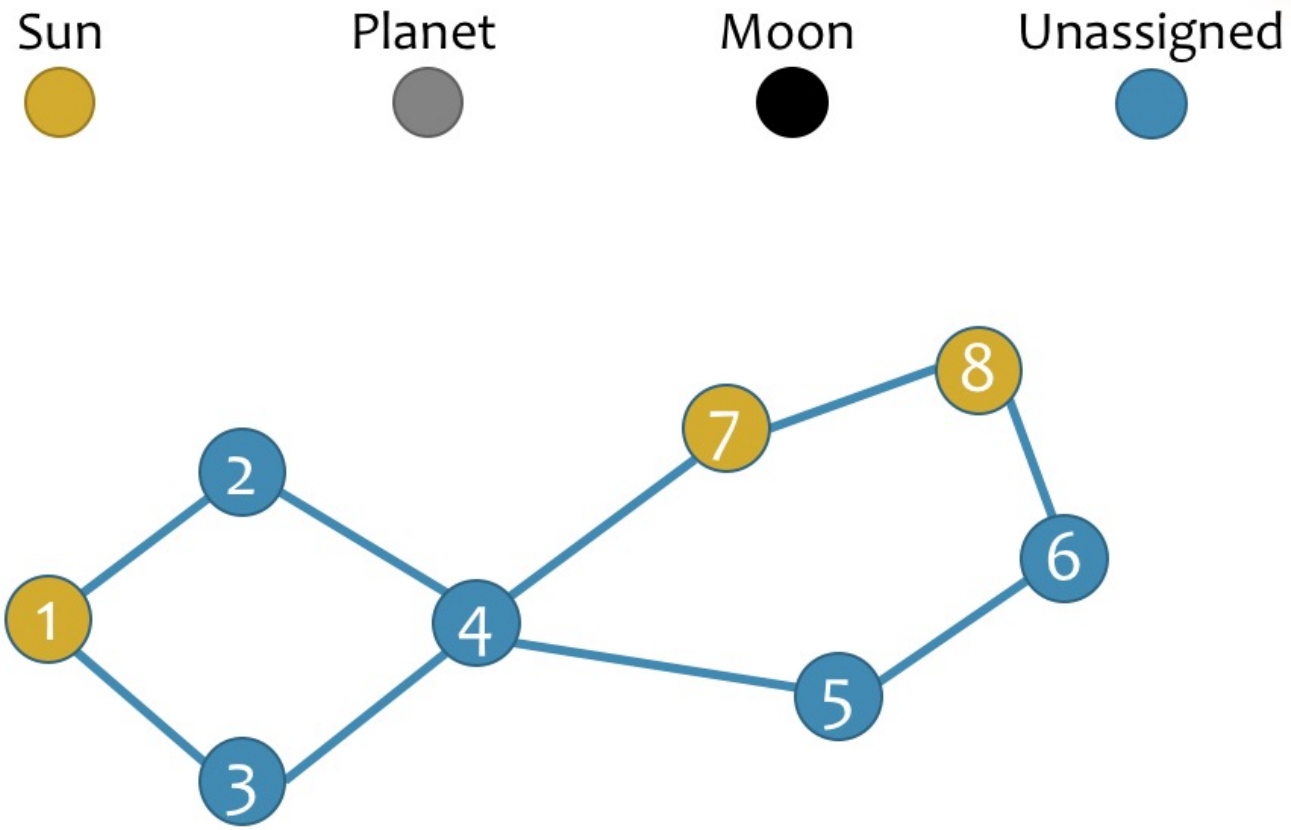
- No knowledge of the entire graph topology
- Path information must be preserved
- One-vertex pickup not feasible
- It is necessary a network discovery protocol
  - Network exploration using messages

Distributed Solar Merger (DSM)



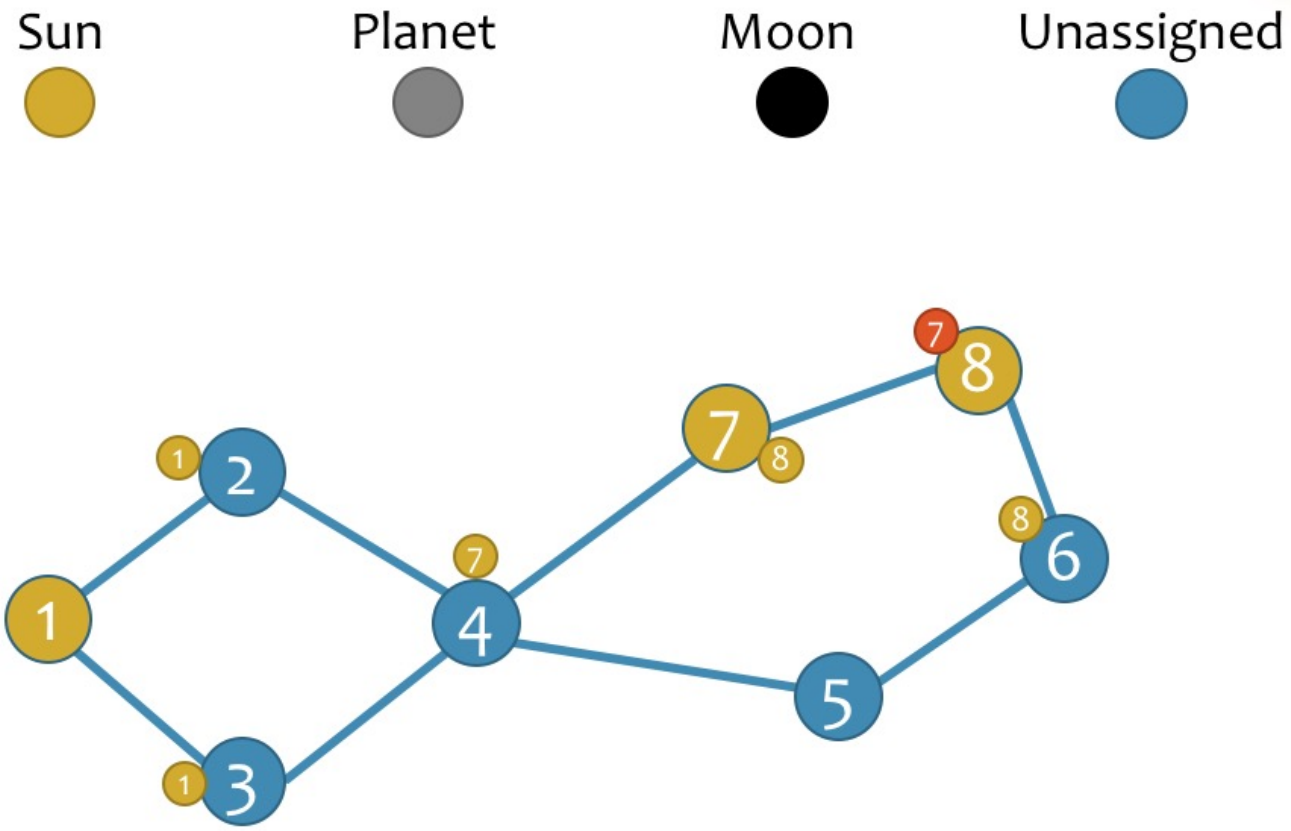
Sun Generation

Distributed Solar Merger (DSM)



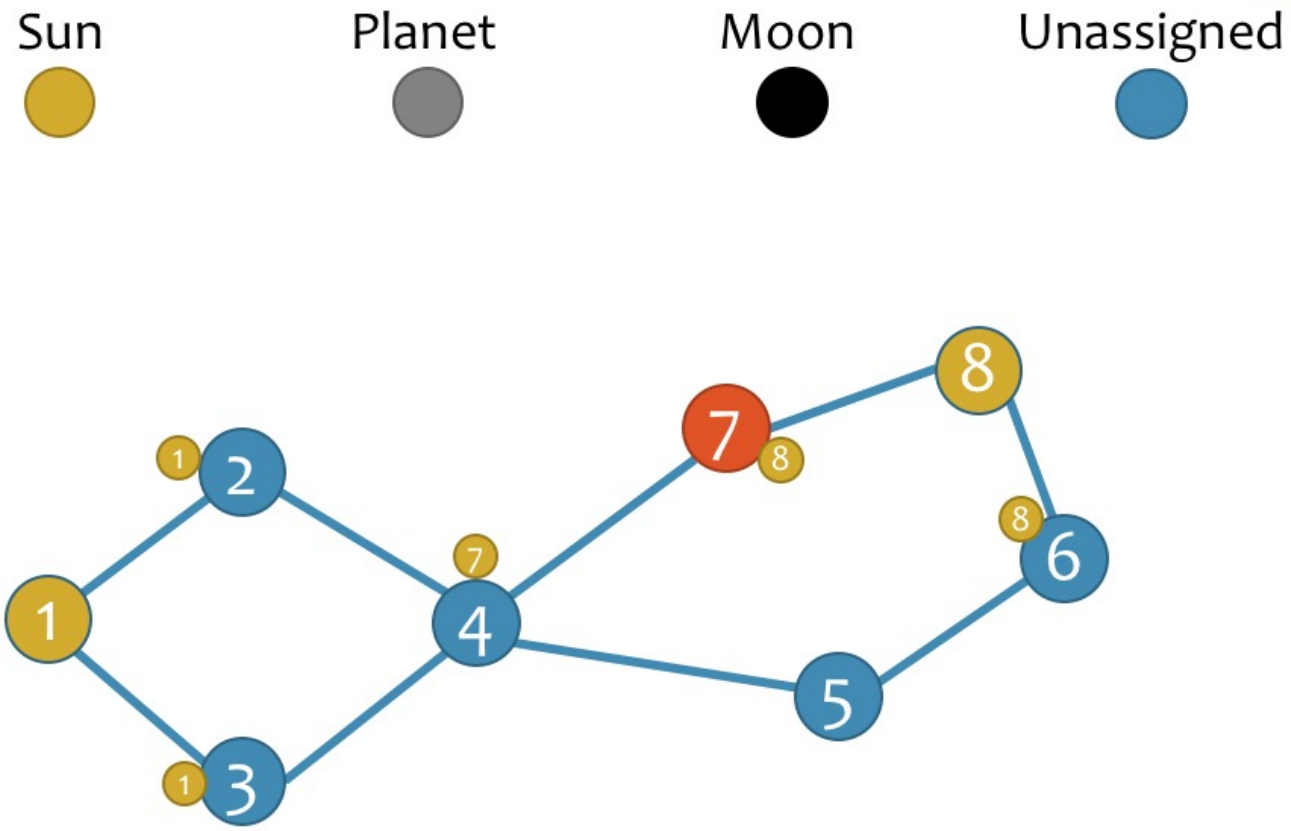
Sun Generation

Distributed Solar Merger (DSM)



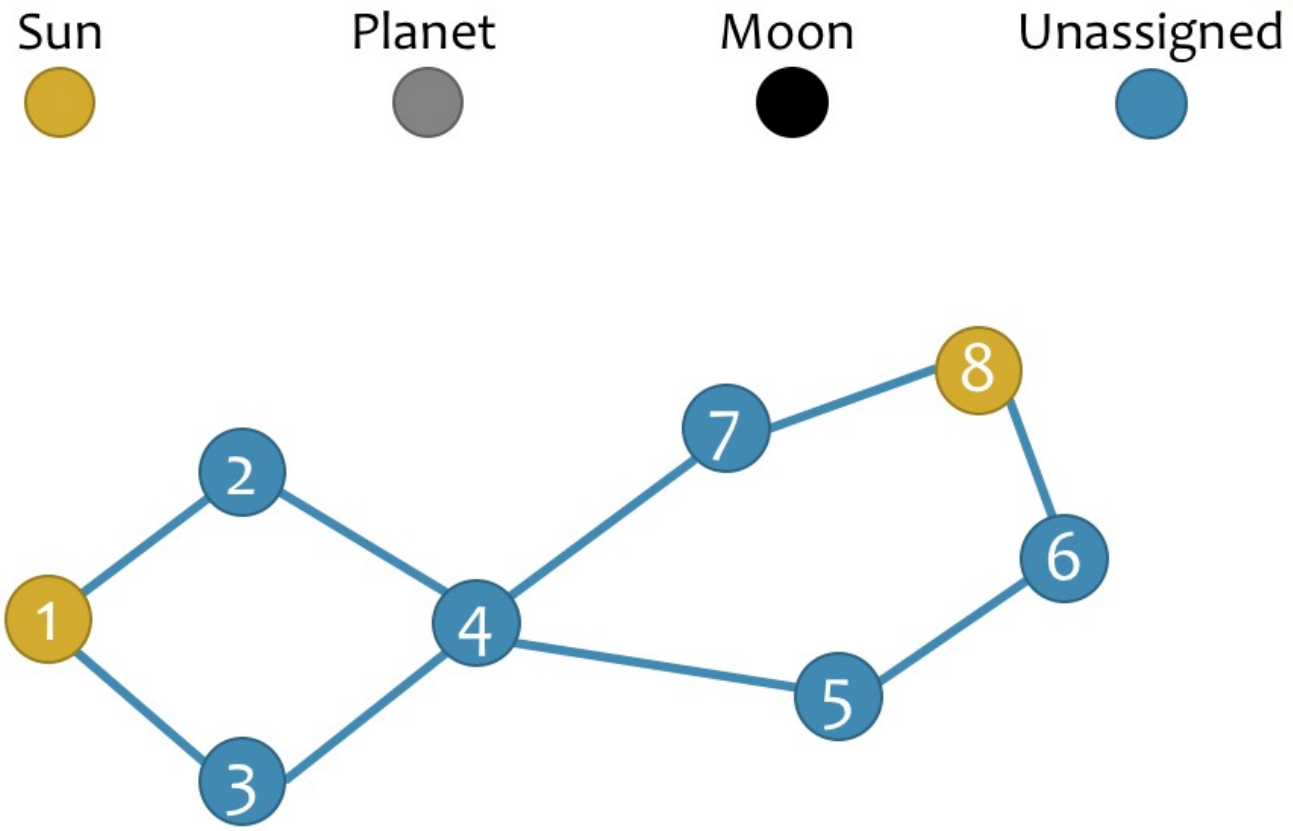
Correction procedure

Distributed Solar Merger (DSM)

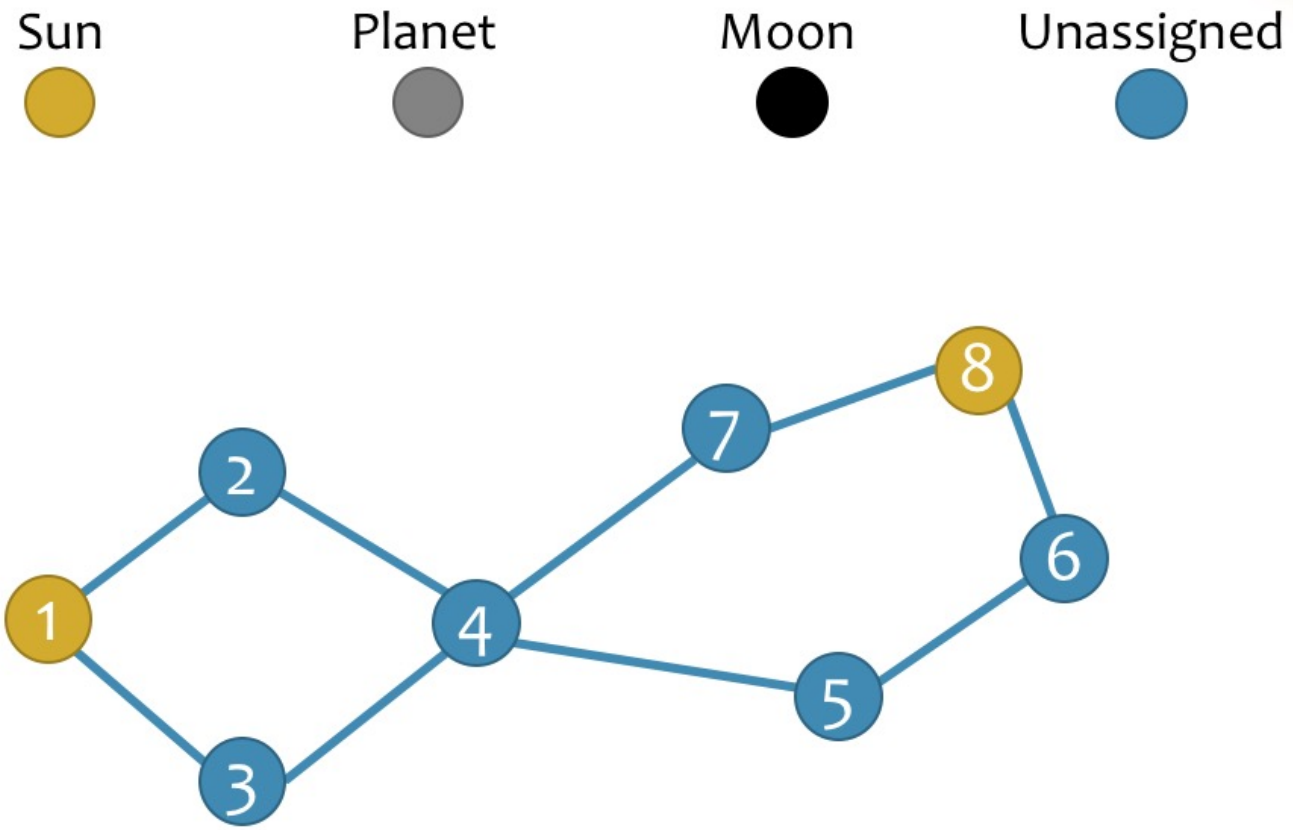


Correction procedure

Distributed Solar Merger (DSM)

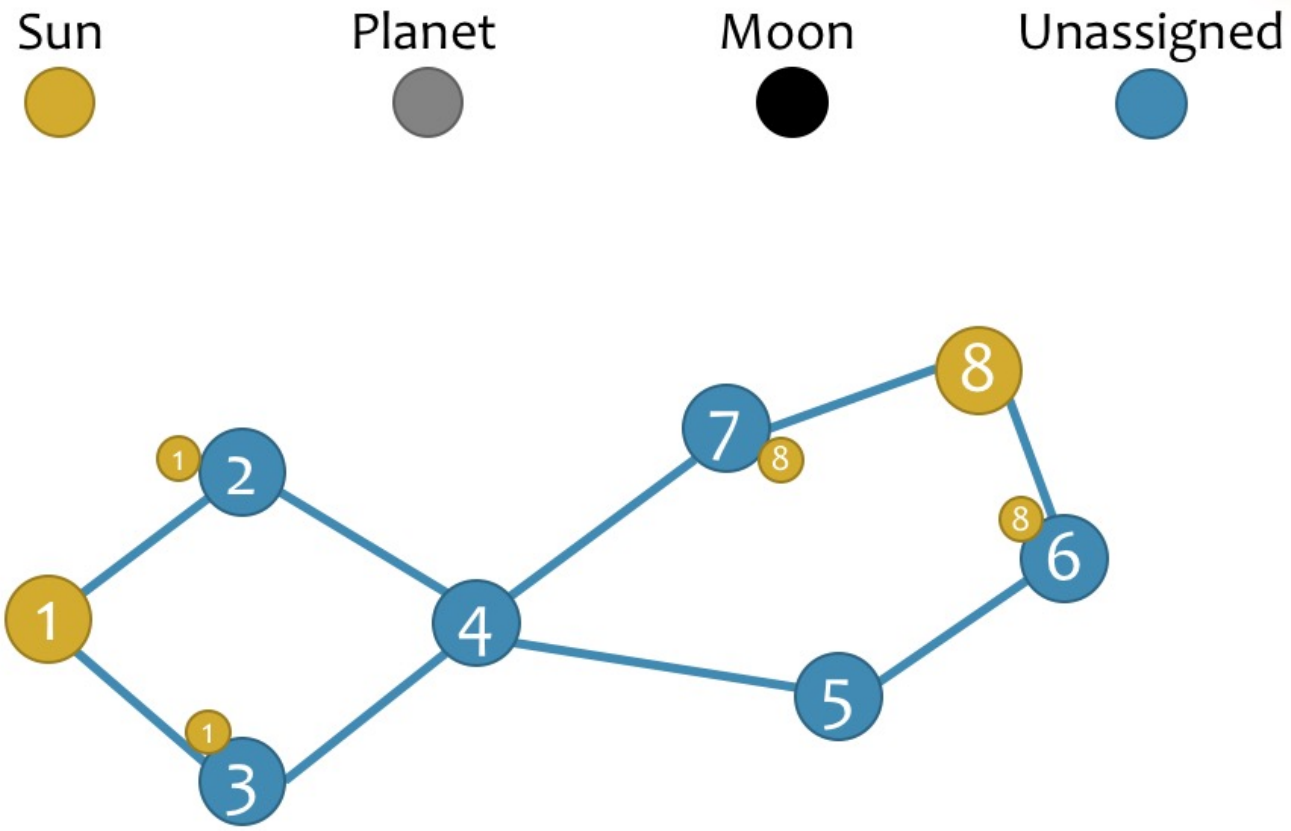


Distributed Solar Merger (DSM)



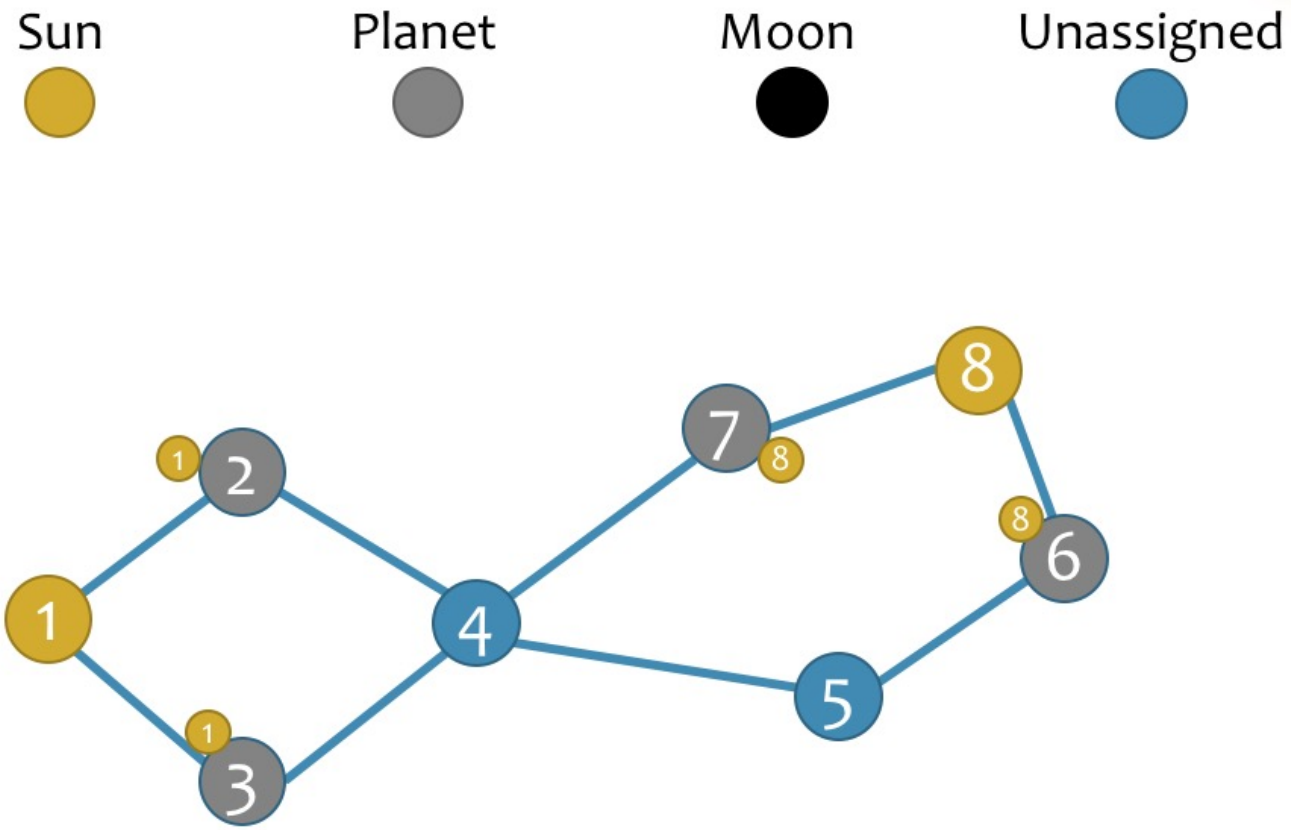
Sun Offer

Distributed Solar Merger (DSM)



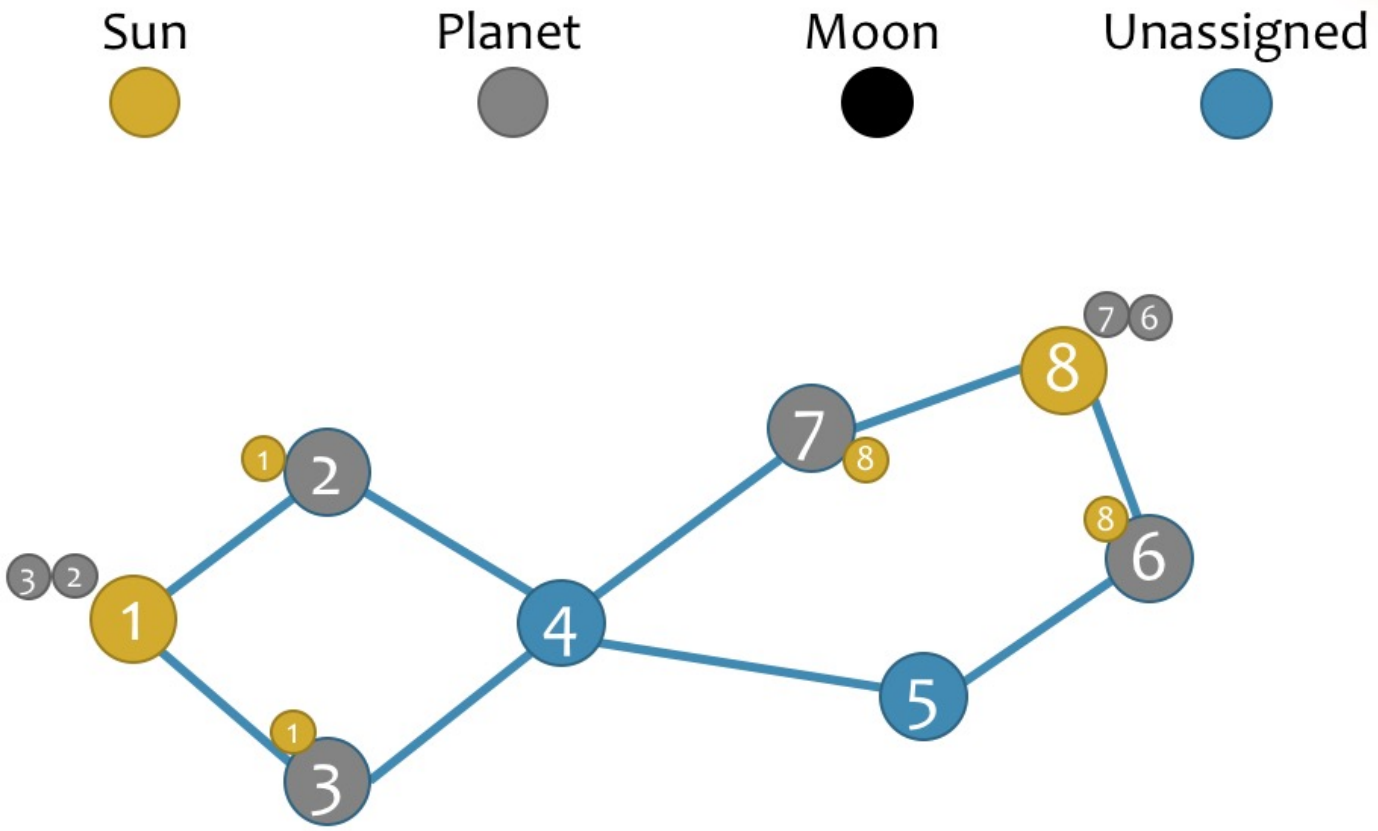
Planet Response

Distributed Solar Merger (DSM)



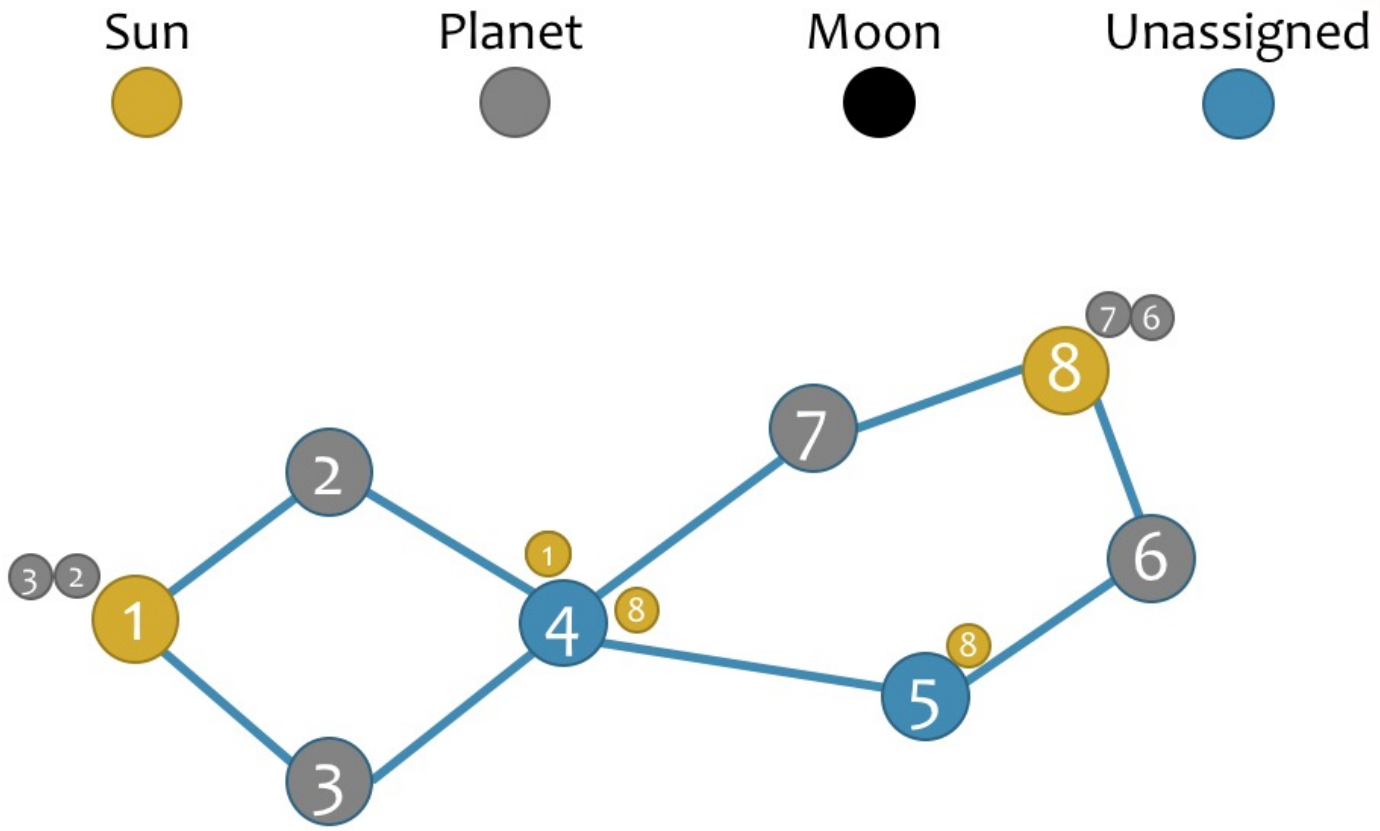
Planet Response

Distributed Solar Merger (DSM)



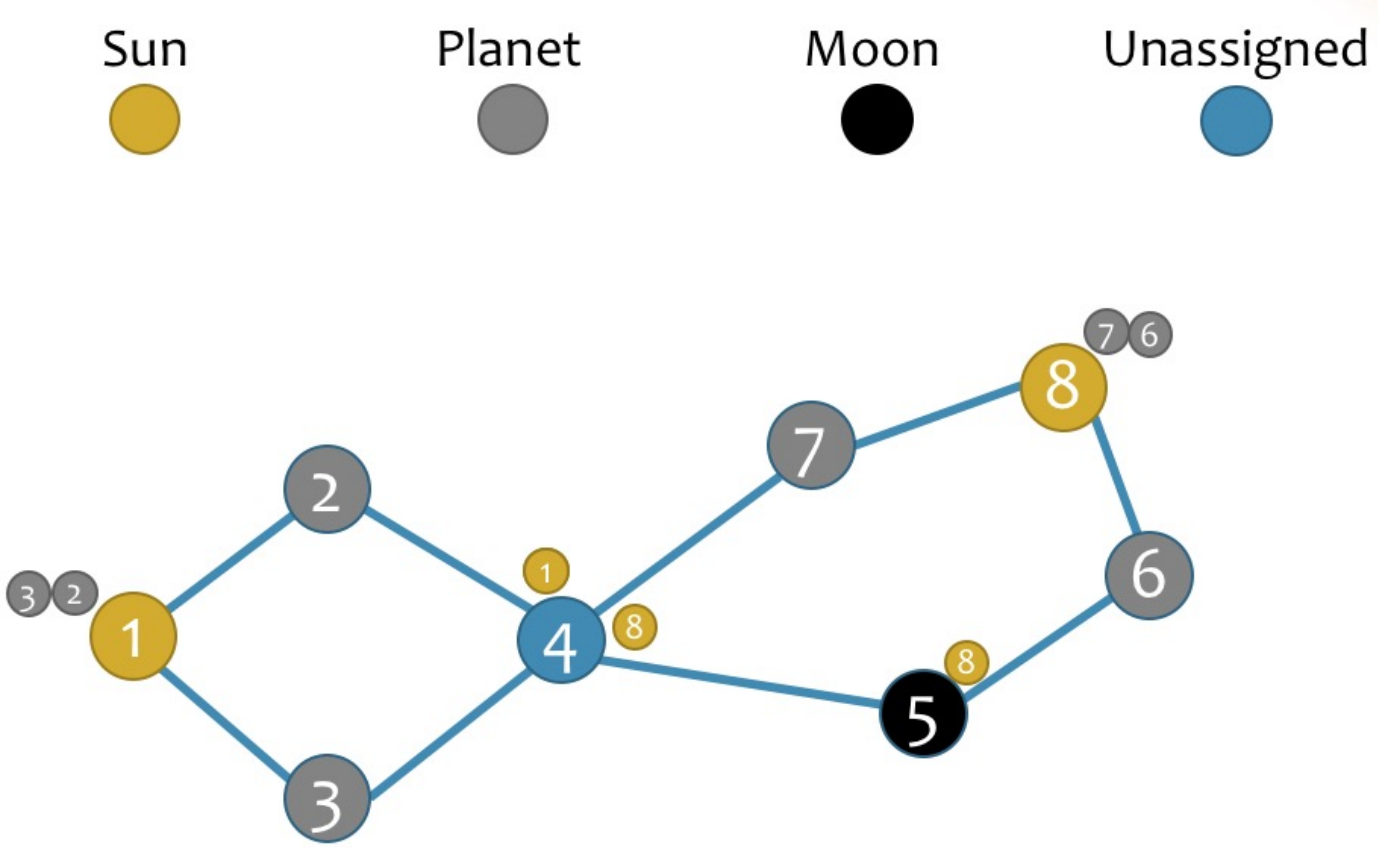
Regime Merger

Distributed Solar Merger (DSM)



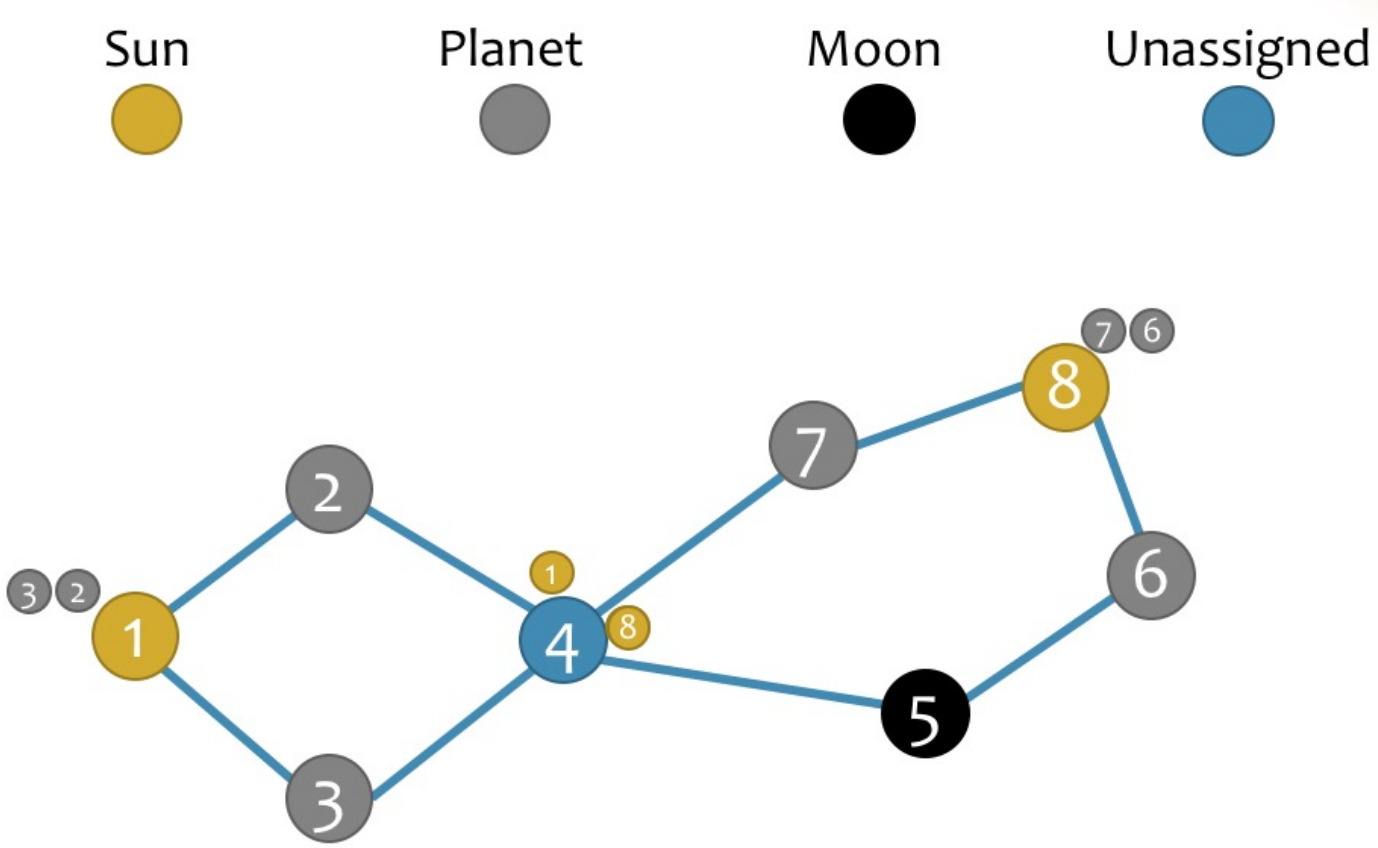
Regime Merger

Distributed Solar Merger (DSM)



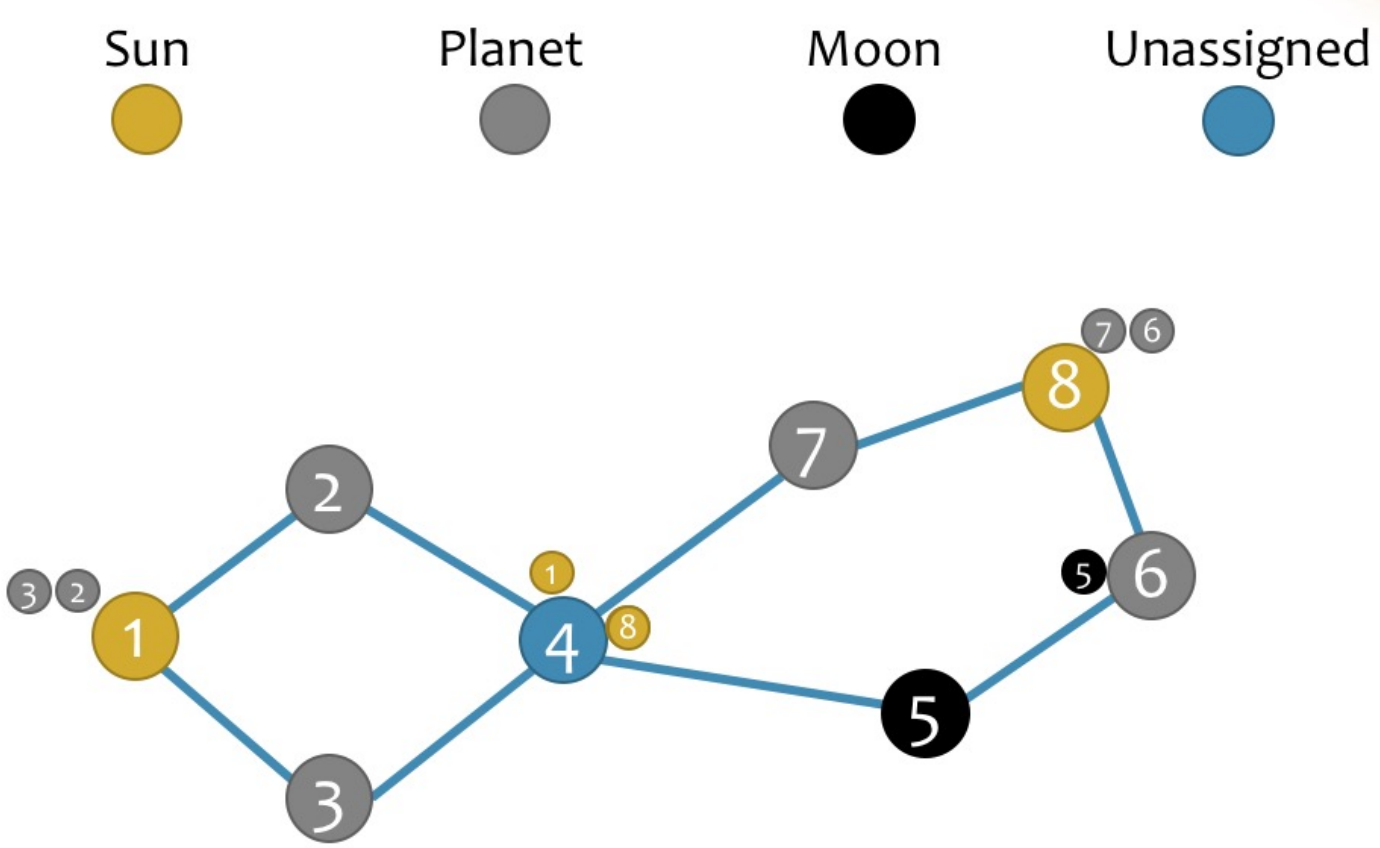
Regime Merger

Distributed Solar Merger (DSM)



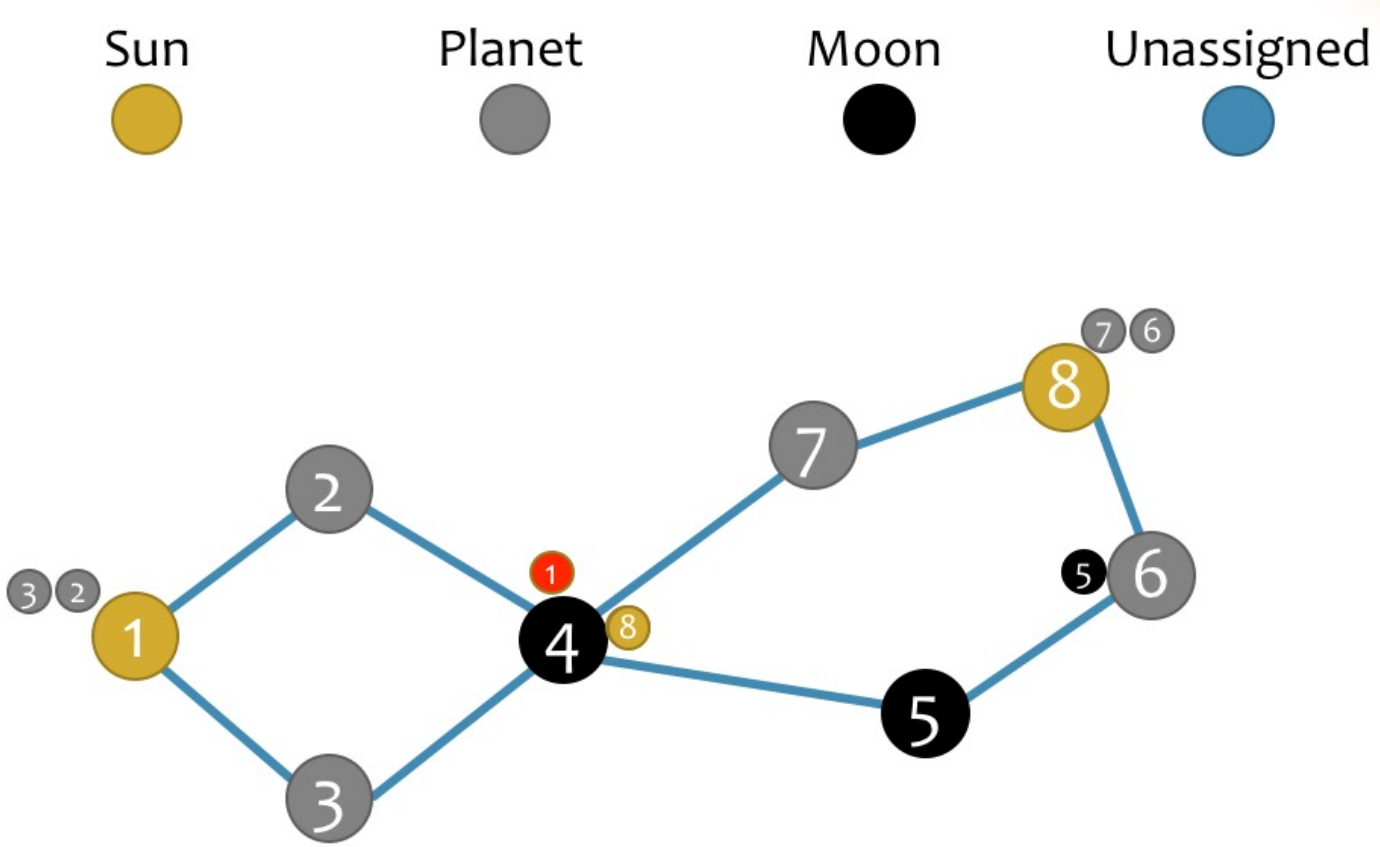
Regime Merger

Distributed Solar Merger (DSM)



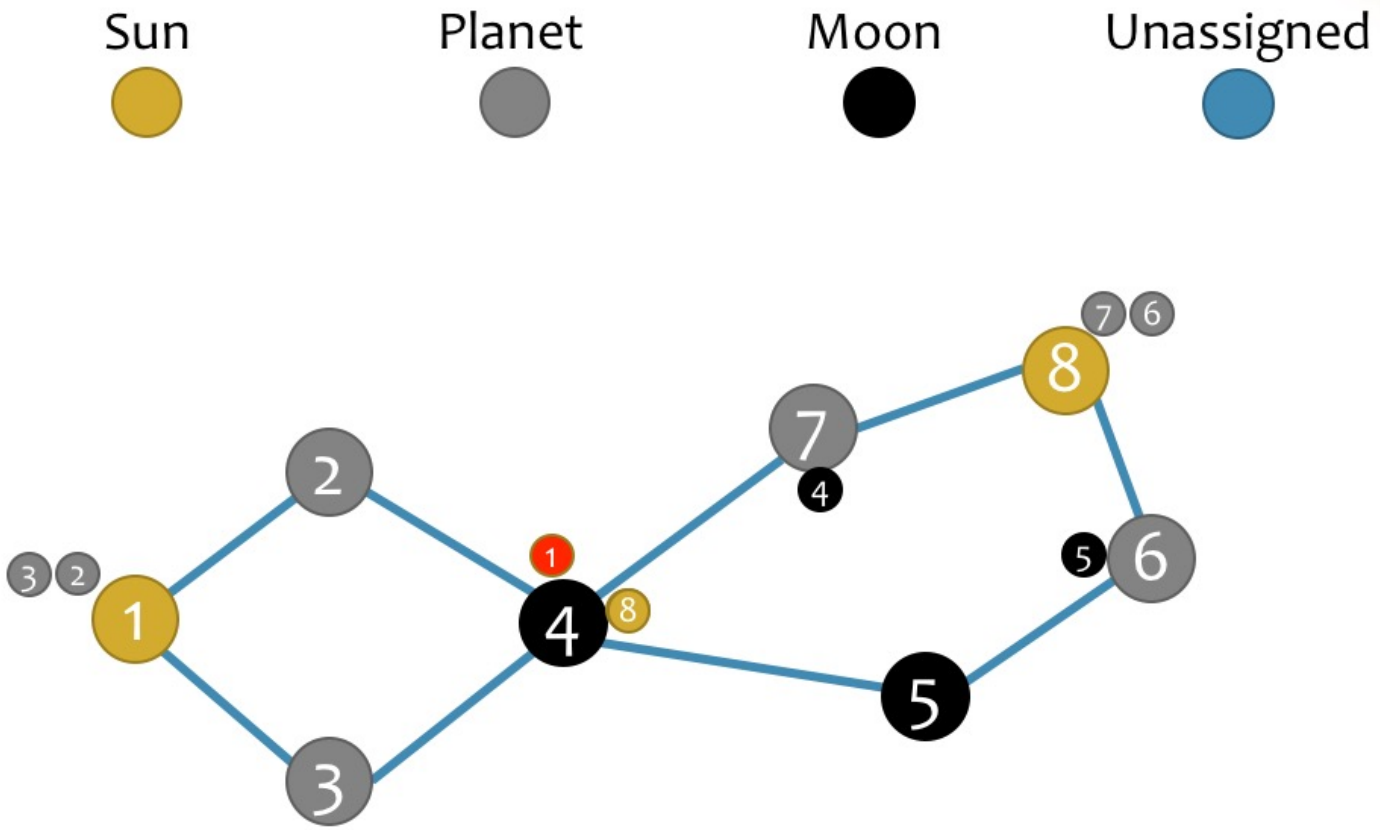
Regime Merger

Distributed Solar Merger (DSM)



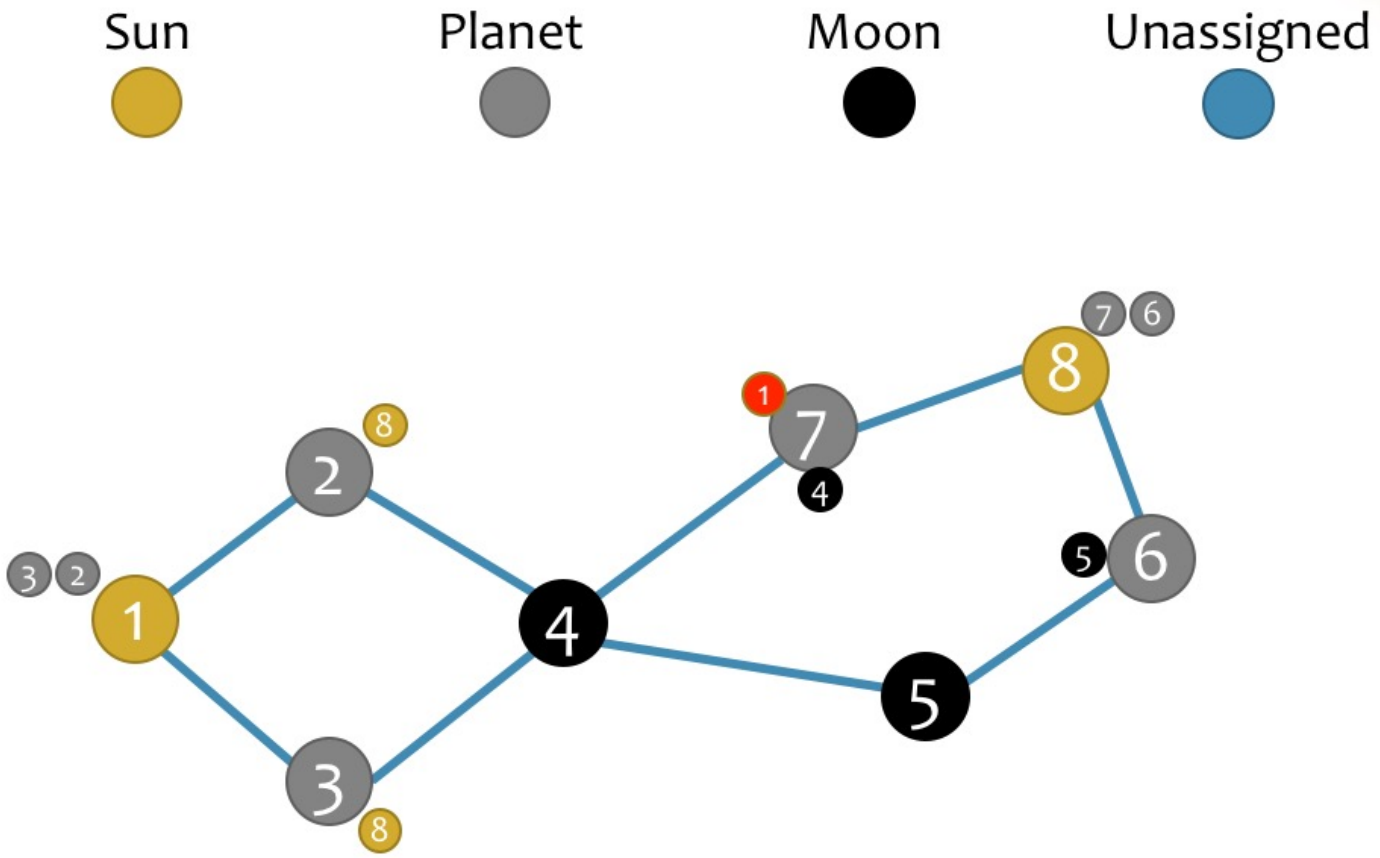
Regime Merger

Distributed Solar Merger (DSM)



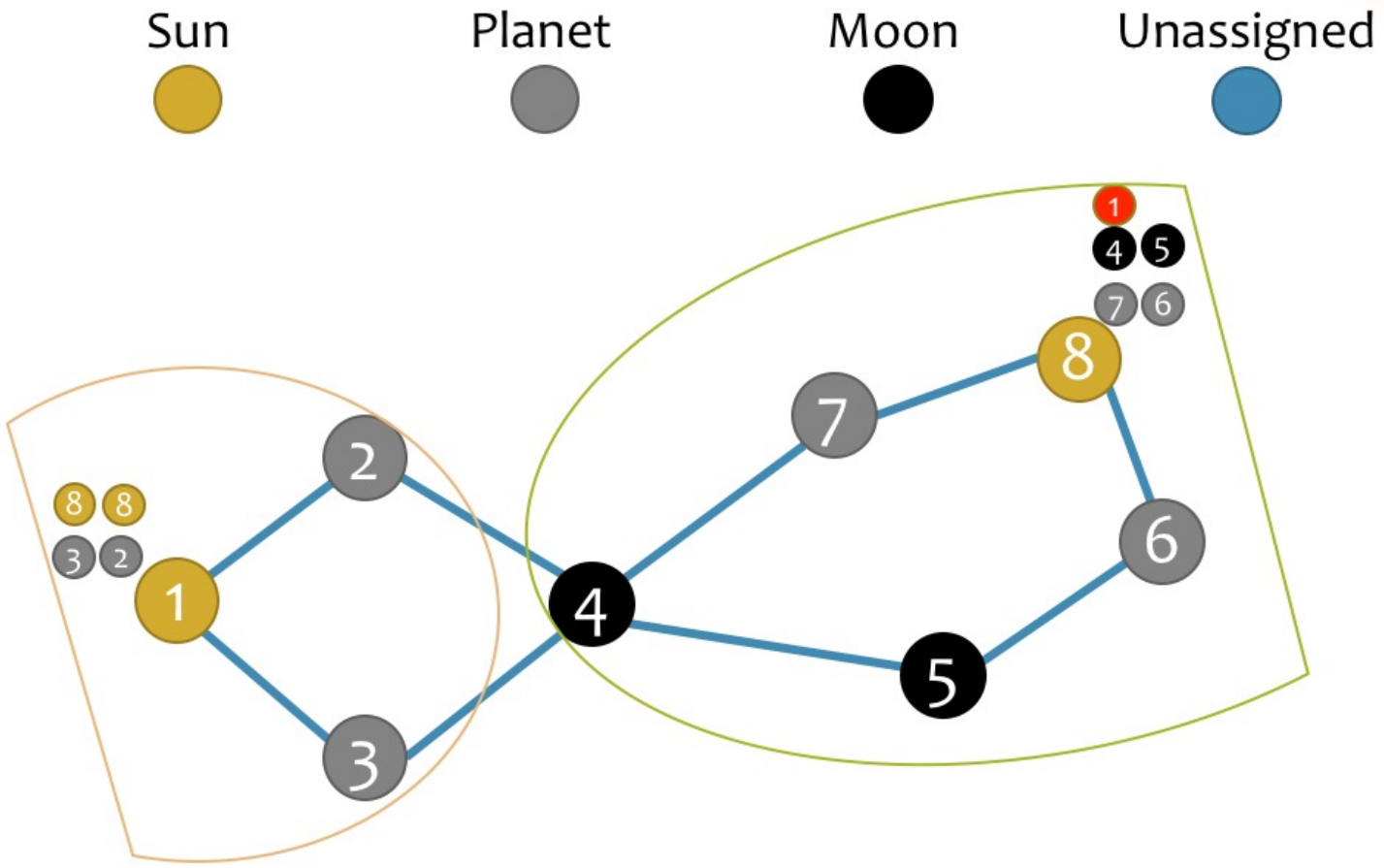
Regime Merger

Distributed Solar Merger (DSM)



Message Delivery

Distributed Solar Merger (DSM)



Distributed Solar Merger (DSM)

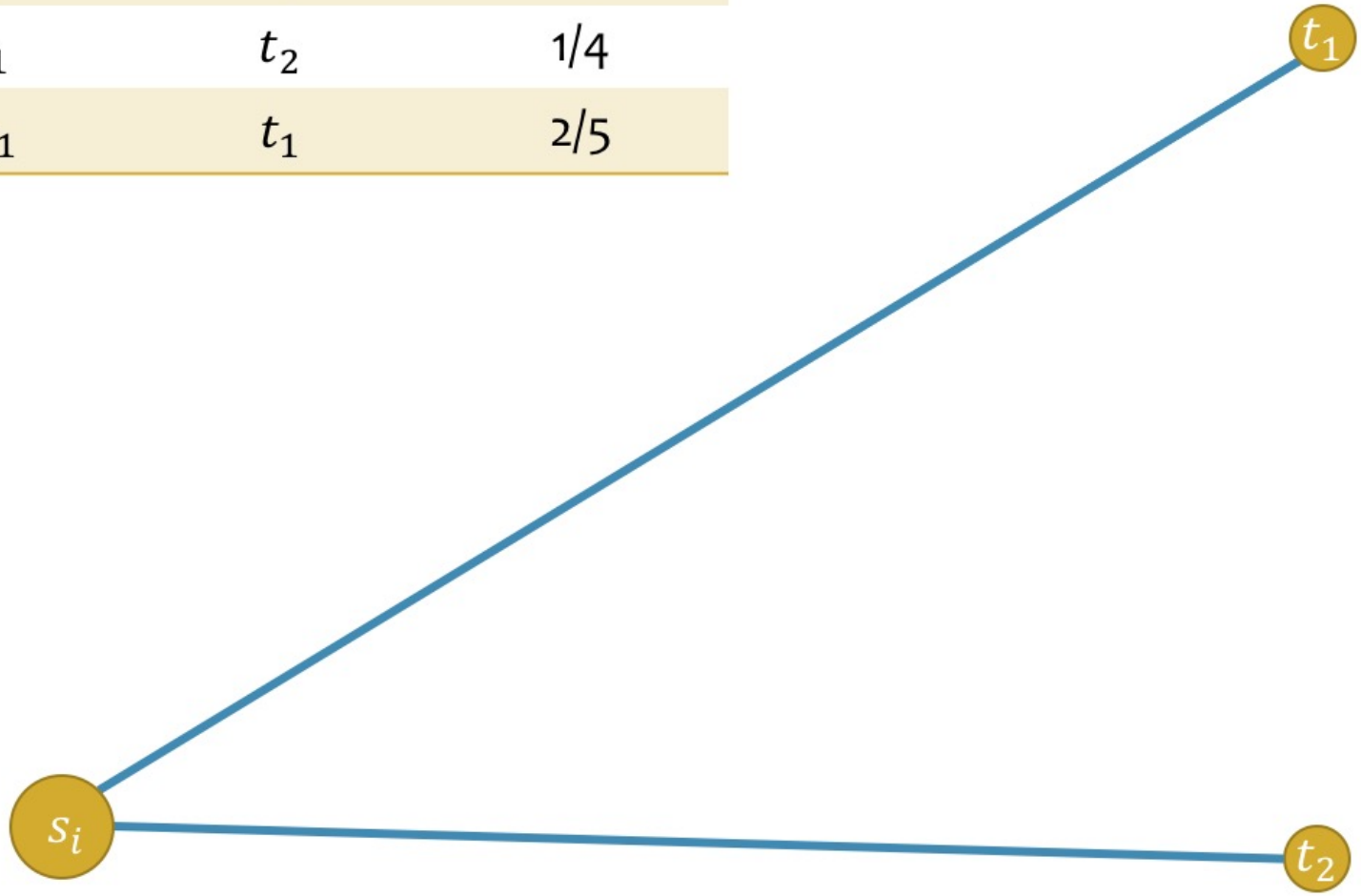
- Process repeated until 100% coverage is achieved
- All paths are found using sun offers
  - Conflict resolutions allow neighbor discovery
- The new level (vertex and edges) is created
- Control passes to the new level

Distributed Solar Placer (DSP)

- For each of its planets/moons it knows:
  - On which path they are
  - Where they are on the path
    - A path between two suns is at most of length 5
- Each sun knows the coordinates of its neighbors

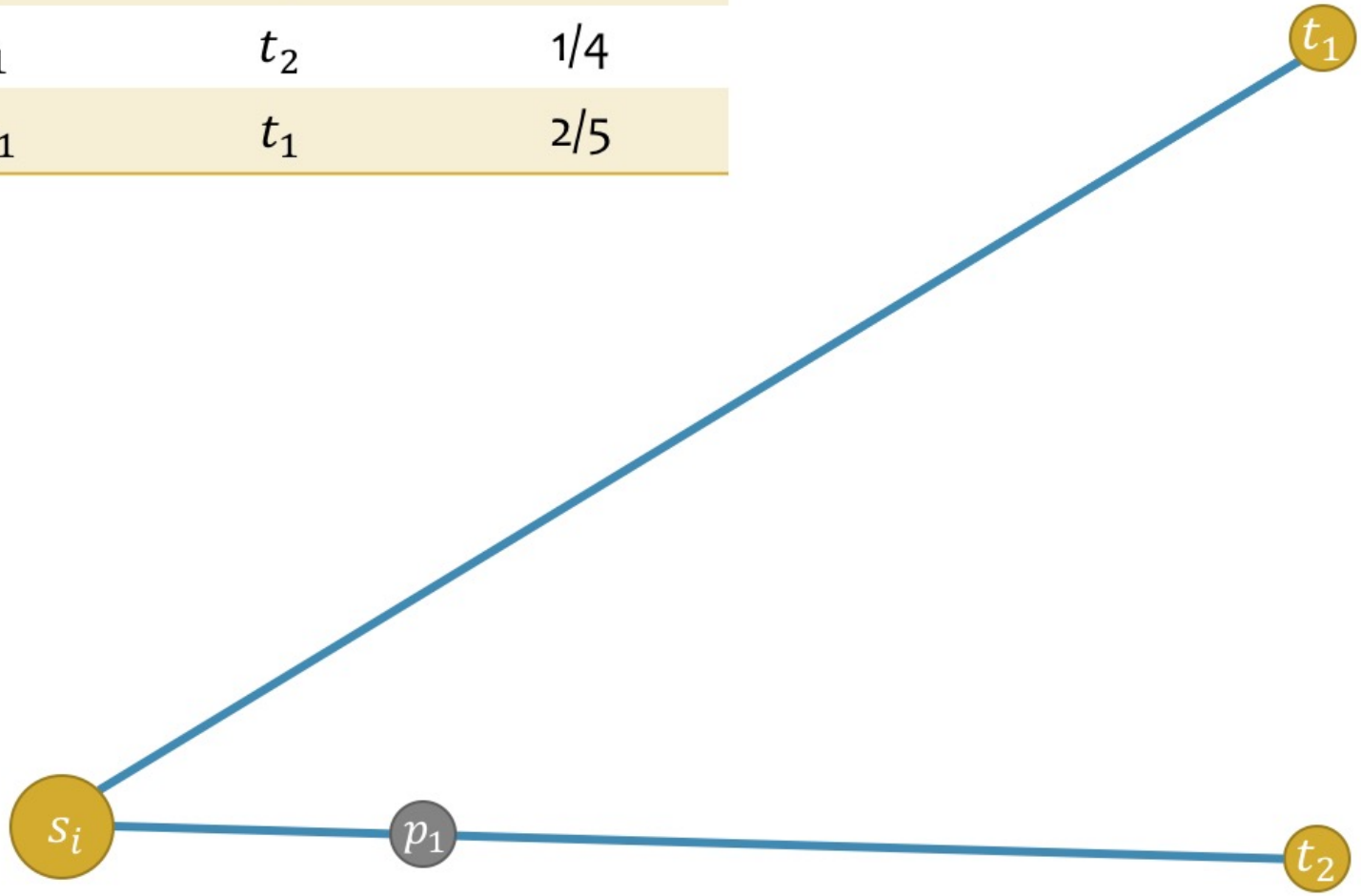
Distributed Solar Placer (DSP)

Planets/M.	Path	Position
p_1	t_1	1/5
p_1	t_2	1/4
m_1	t_1	2/5



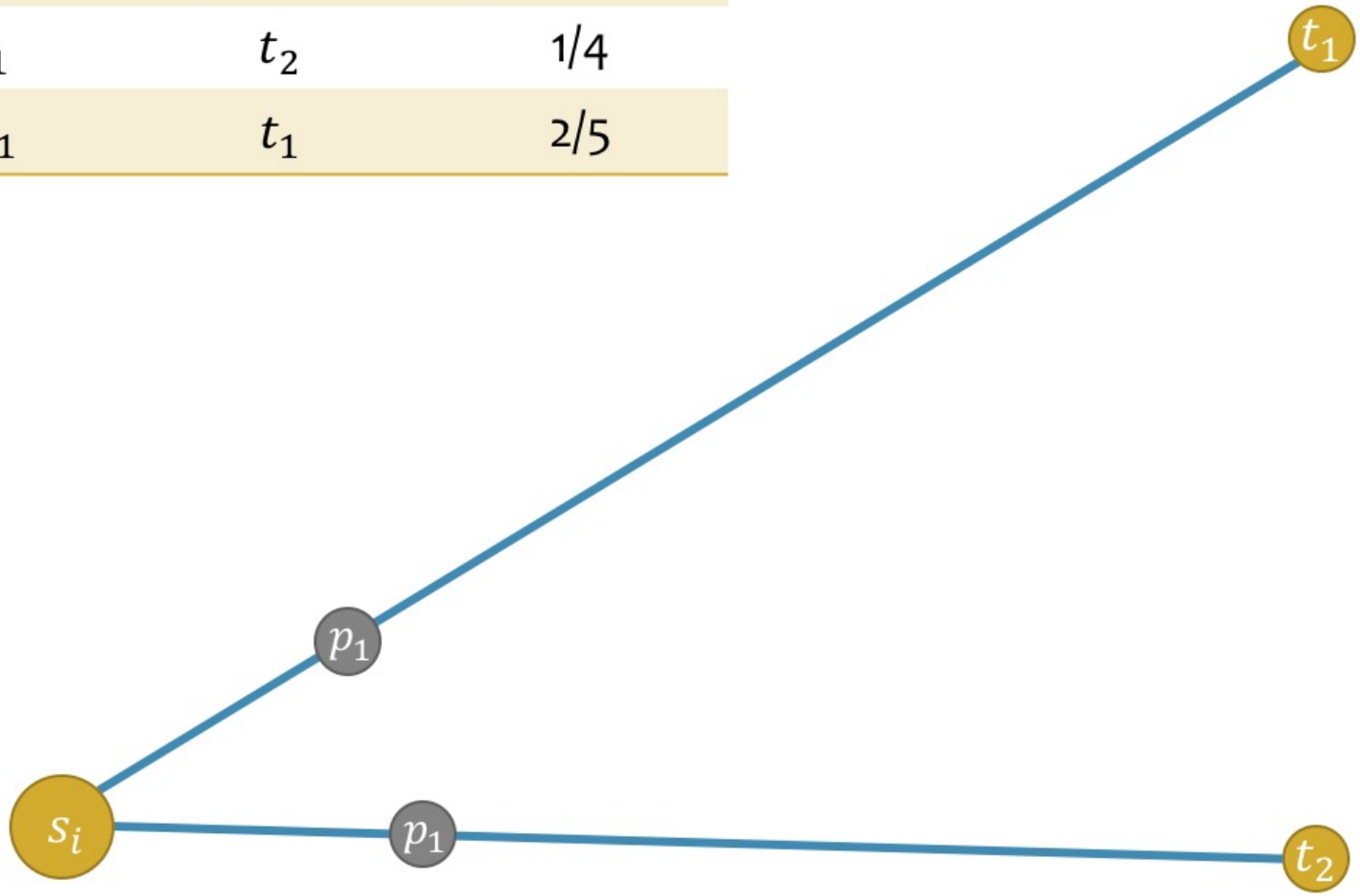
Distributed Solar Placer (DSP)

Planets/M.	Path	Position
p_1	t_1	1/5
p_1	t_2	1/4
m_1	t_1	2/5



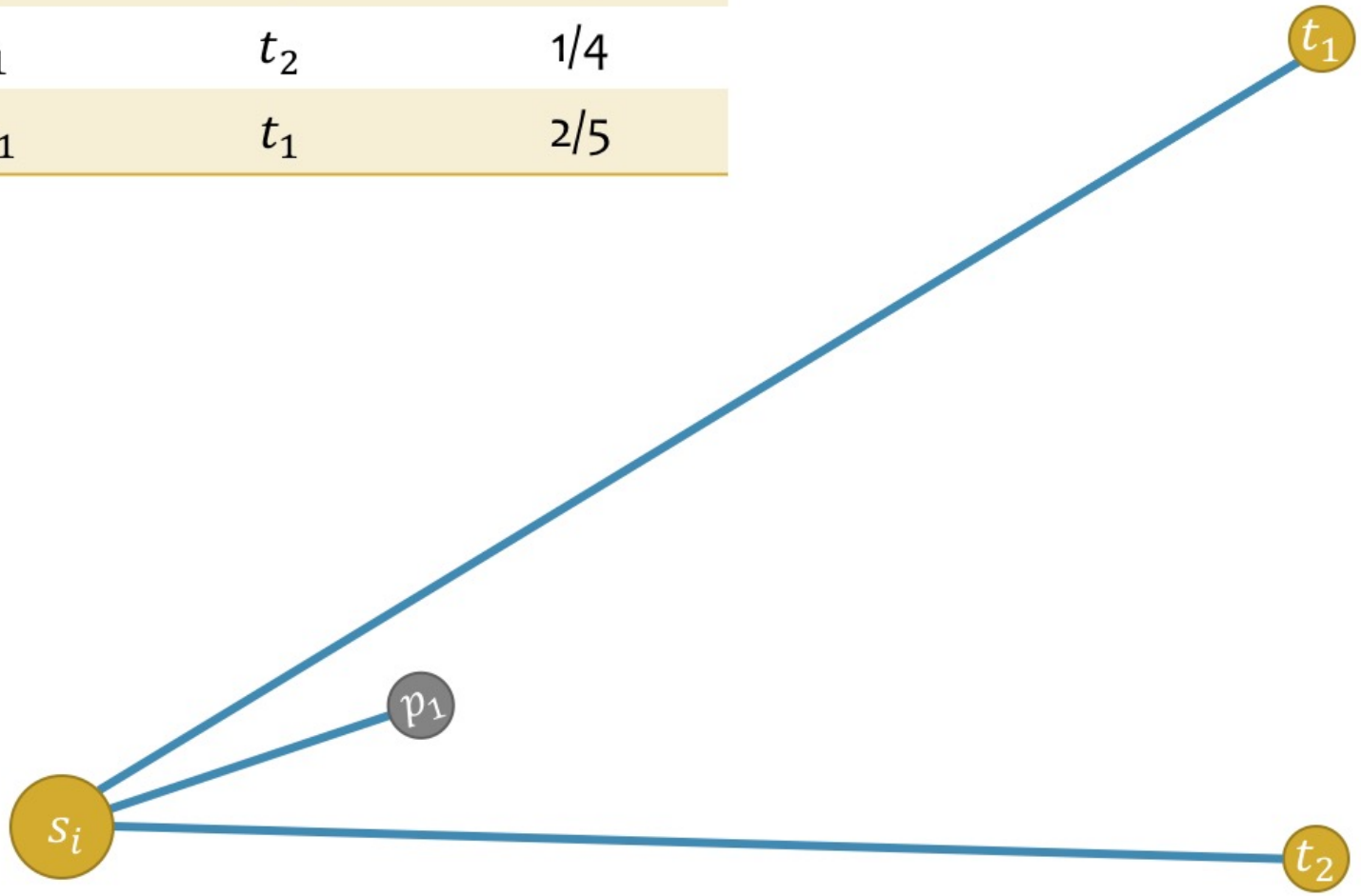
Distributed Solar Placer (DSP)

Planets/M.	Path	Position
p_1	t_1	1/5
p_1	t_2	1/4
m_1	t_1	2/5



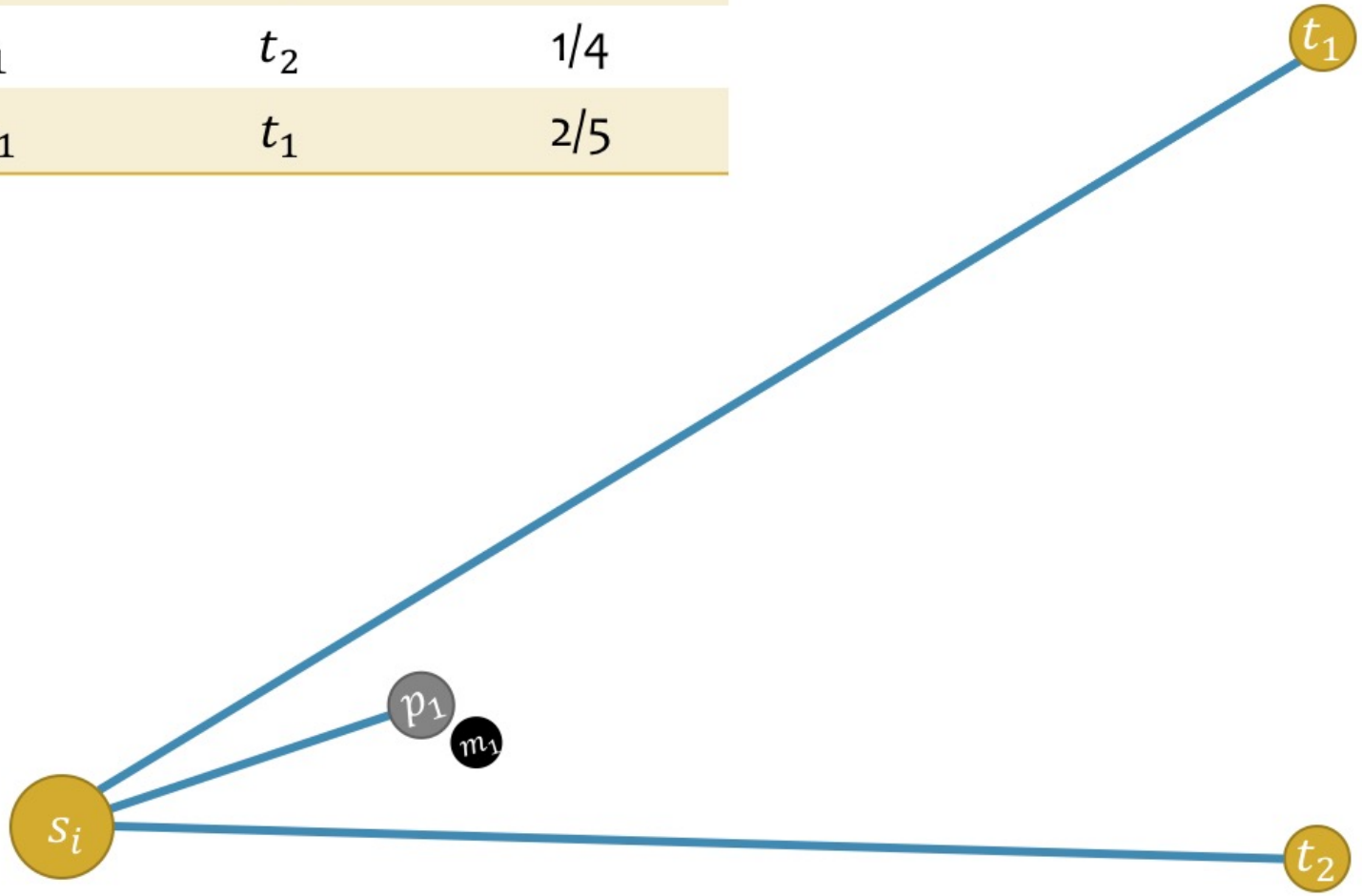
Distributed Solar Placer (DSP)

Planets/M.	Path	Position
p_1	t_1	1/5
p_1	t_2	1/4
m_1	t_1	2/5



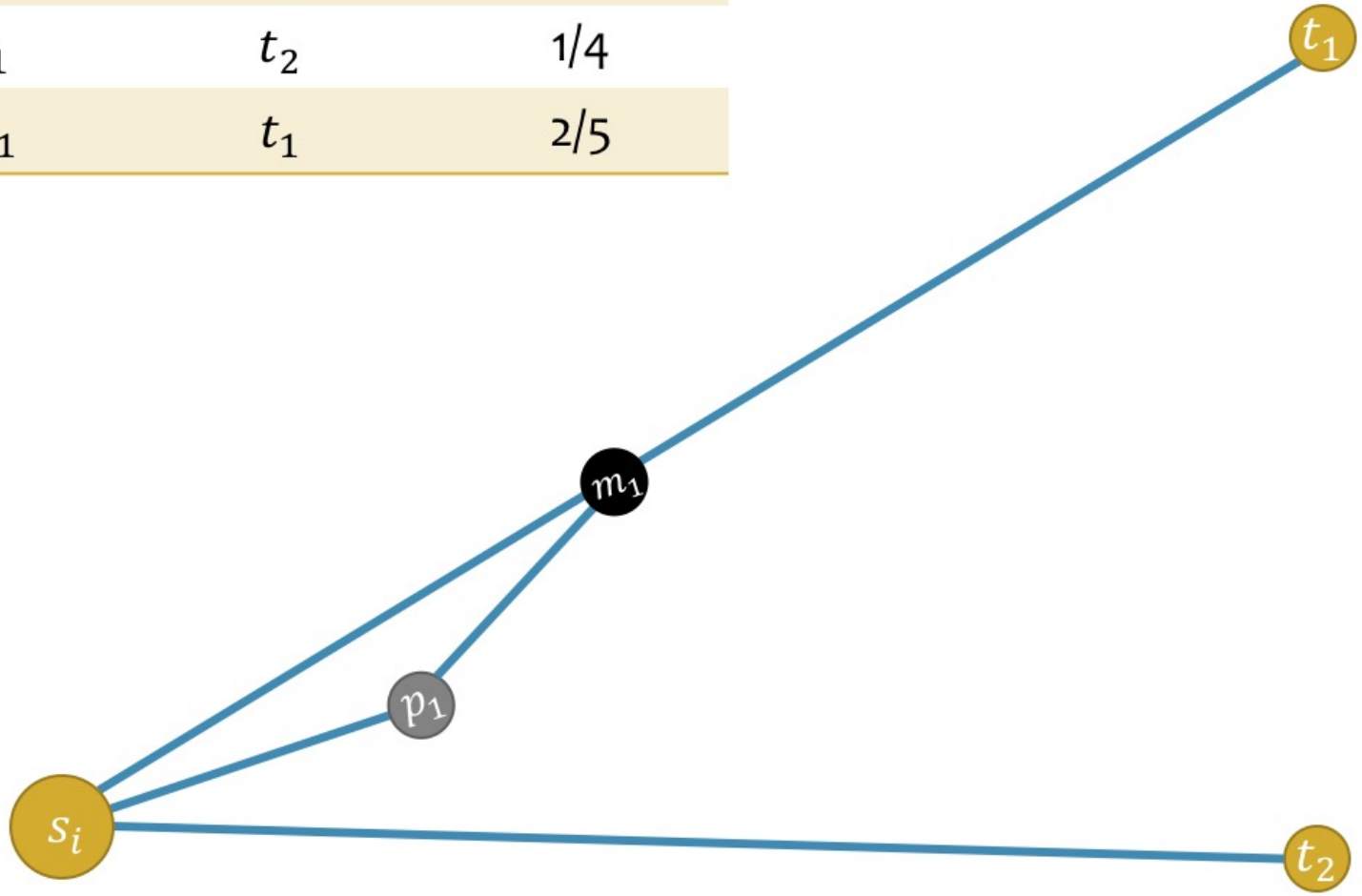
Distributed Solar Placer (DSP)

Planets/M.	Path	Position
p_1	t_1	1/5
p_1	t_2	1/4
m_1	t_1	2/5



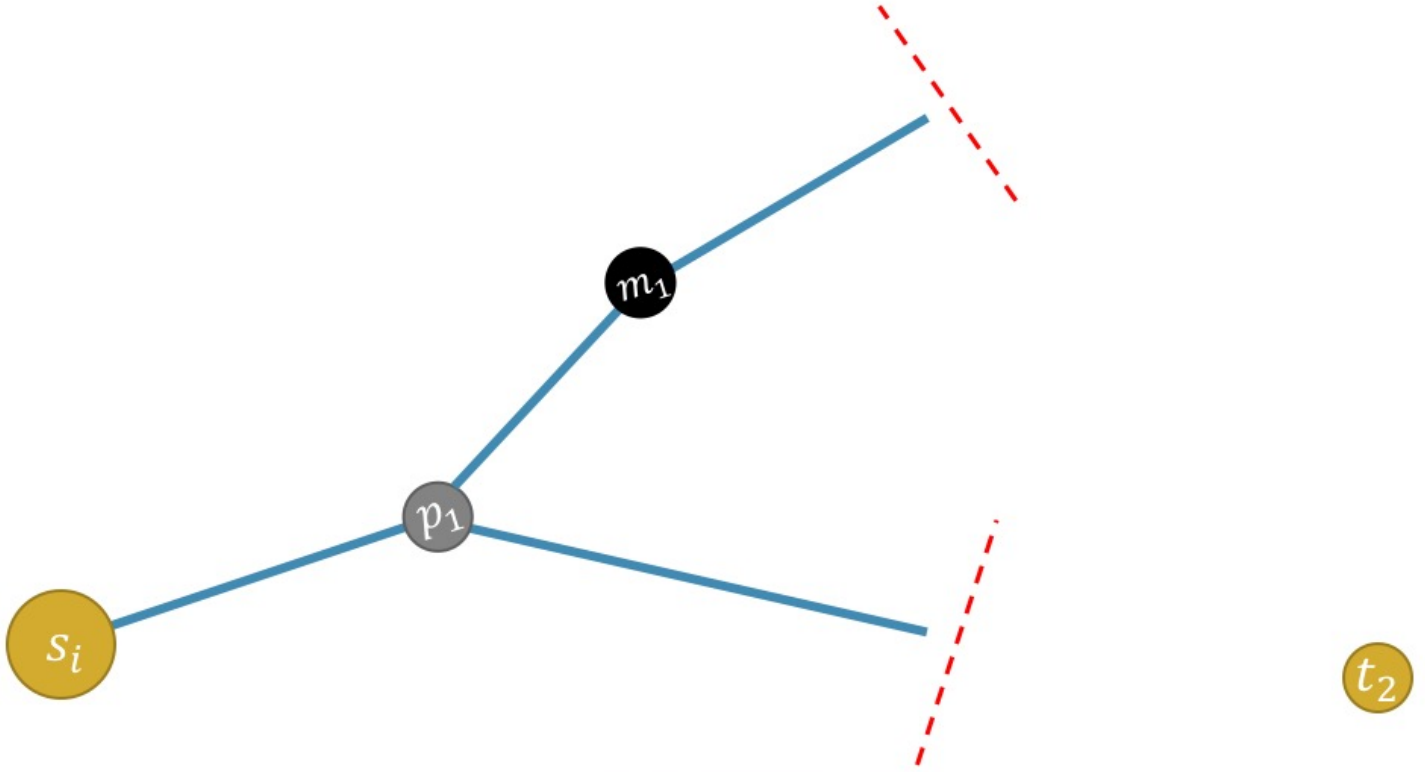
Distributed Solar Placer (DSP)

Planets/M.	Path	Position
p_1	t_1	1/5
p_1	t_2	1/4
m_1	t_1	2/5



Distributed Solar Placer (DSP)

Planets/M.	Path	Position
p_1	t_1	1/5
p_1	t_2	1/4
m_1	t_1	2/5



Gila – Single Level Layout

- Based on FR algorithm
- Repulsive forces computed by means of a controlled flooding scheme
  - Spatial decomposition techniques unfeasible
  - GTD as an approximation of geometric distance
- Layout process takes several iterations
  - Each iteration is split in two phases

Gila – Single Level Layout

Seeding

Vertices broadcast their coordinates

Propagation

Forces computation

Messages broadcast

N° of times a message is propagated depends on its *time to live (k)*

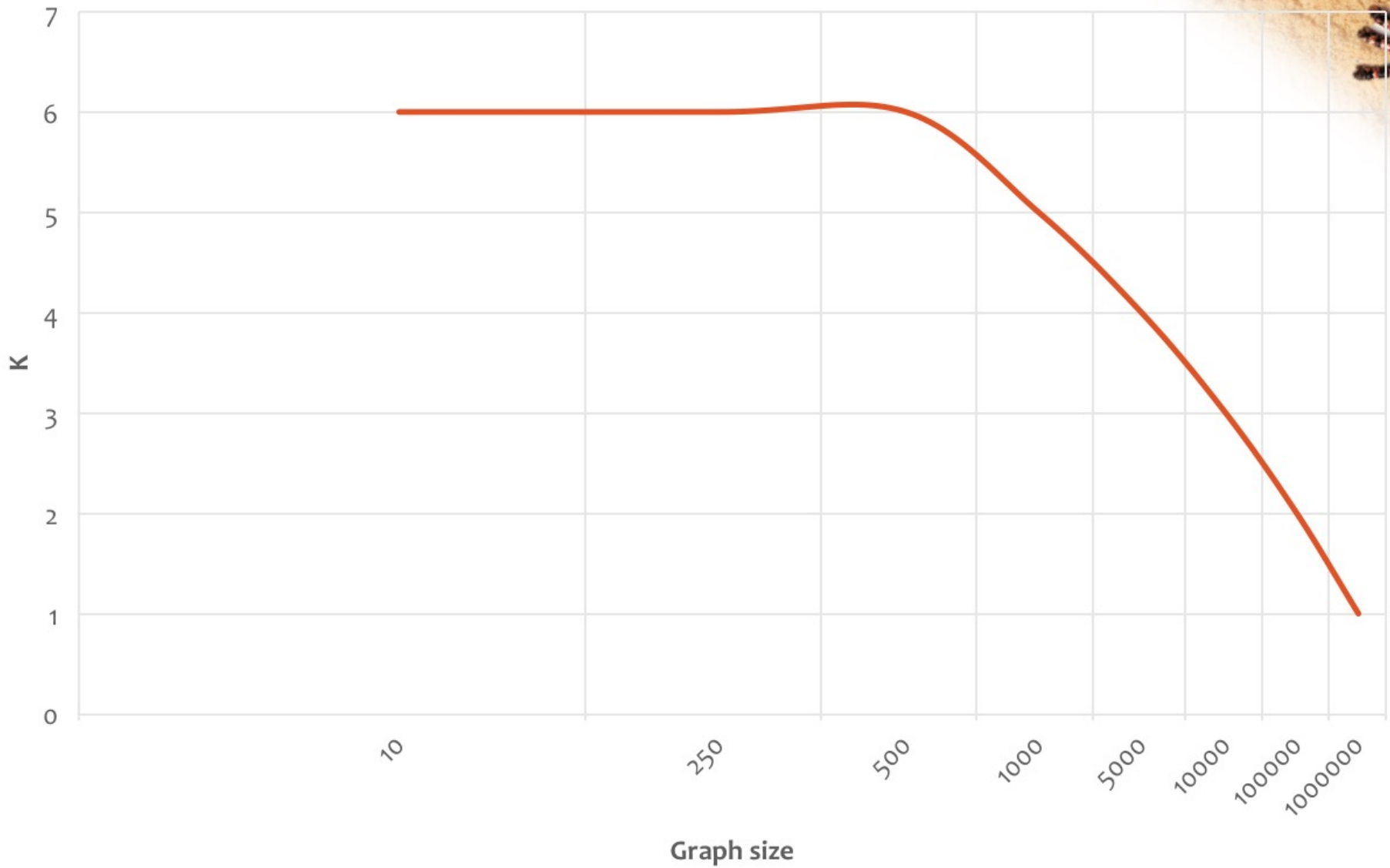
Gila – Single Level Layout

- Overall # of messages: $O(m^k)$
- *A lot* of traffic (and potential overhead)
- Messages are expensive…
- … but a bigger *k* implies better quality

Choose *k*

as a tradeoff between quality and cost

Dynamic *Gila* tuning



Experimental results

Test Suite

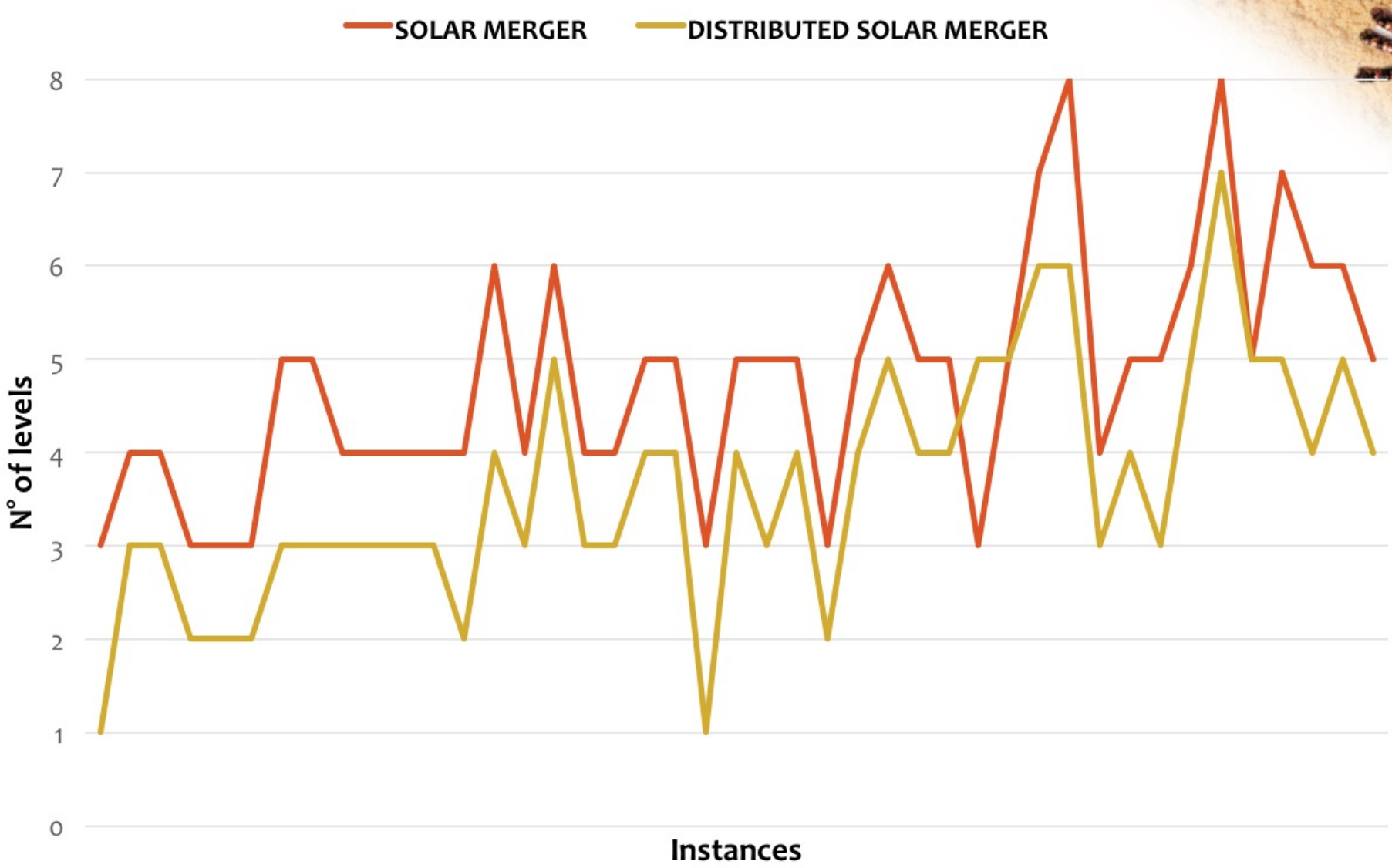
- 43 Small, synthetic graphs
  - *Bartel et al., 2011*
- 5 Medium sized graphs
  - Up to 1.5M edges
- 3 Large graphs
  - Up to ~10M edges

Experimental Environment

- Small, synthetic graphs
  - Single machine pseudo-cluster, MacBook laptop
- Medium sized graphs
  - AWS cluster, 5-15 r3.xlarge instances
- Large graphs
  - AWS cluster, 20-30 r3.xlarge instances

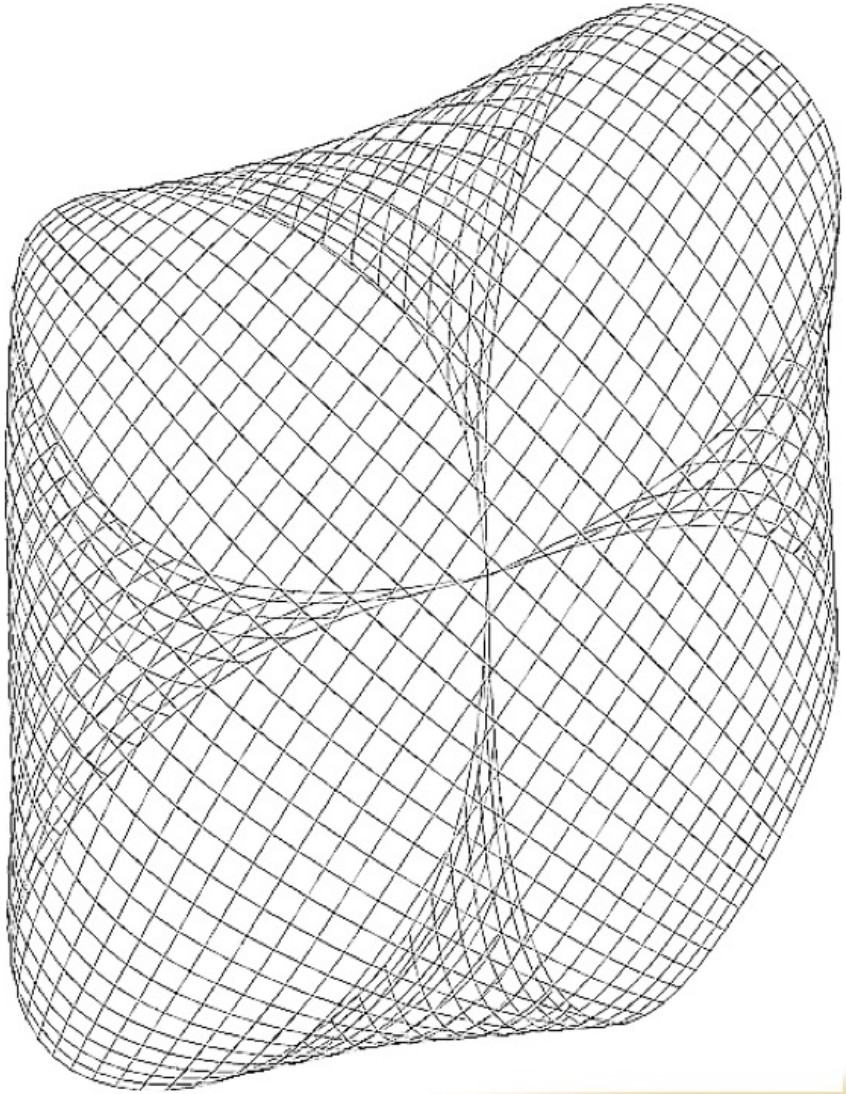
4 cores,
30GB Ram

Number of Levels

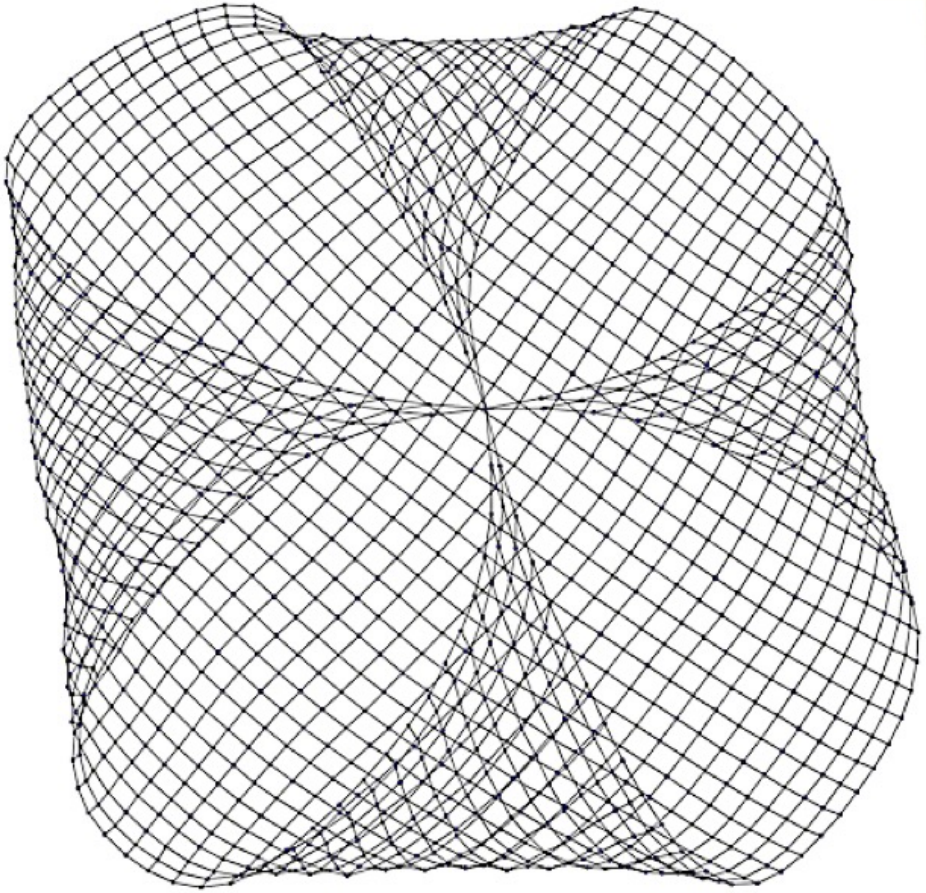


Grid 40x40 Double Folded

Multi-Gila

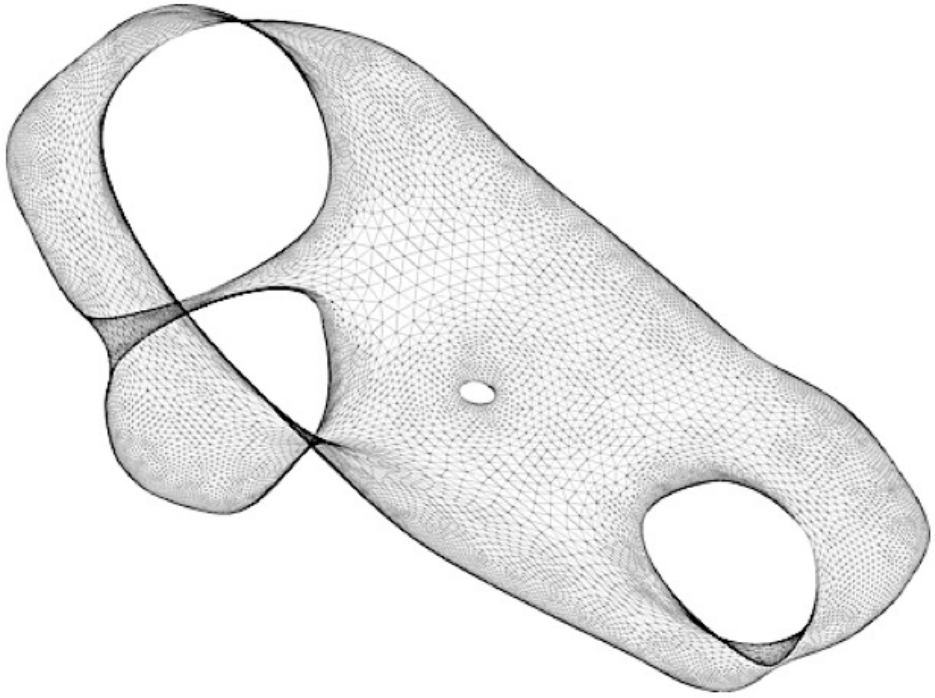


FM³

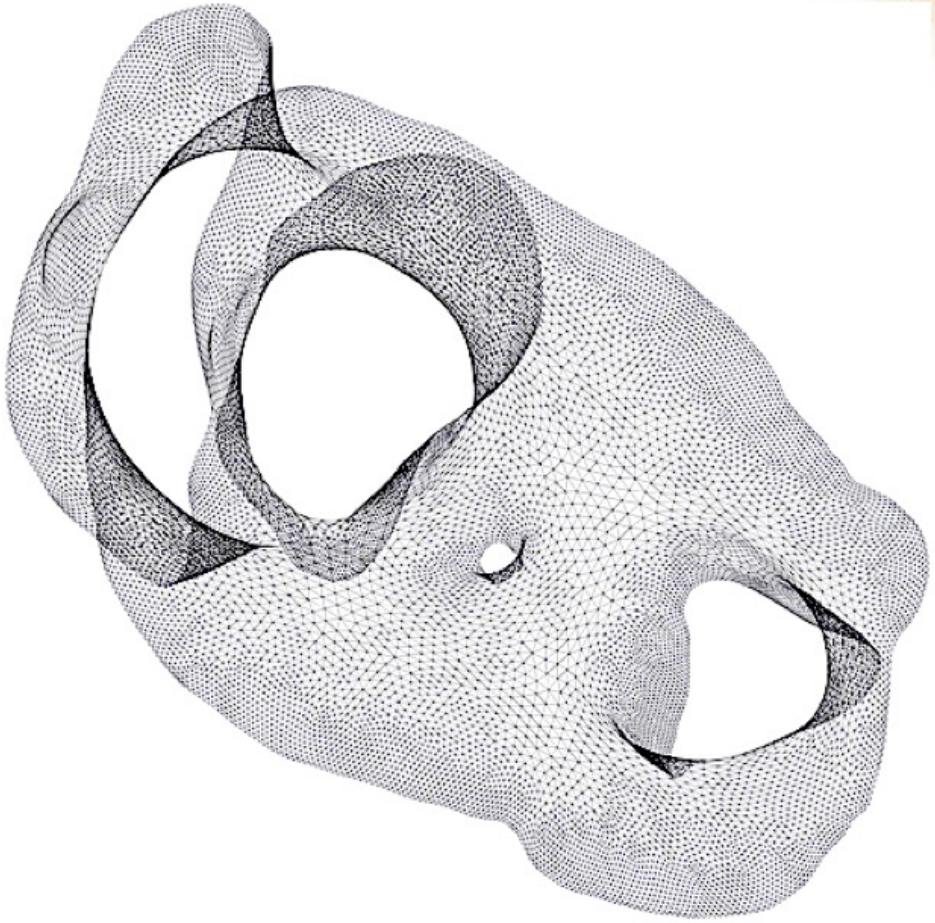


"4elt" Mesh

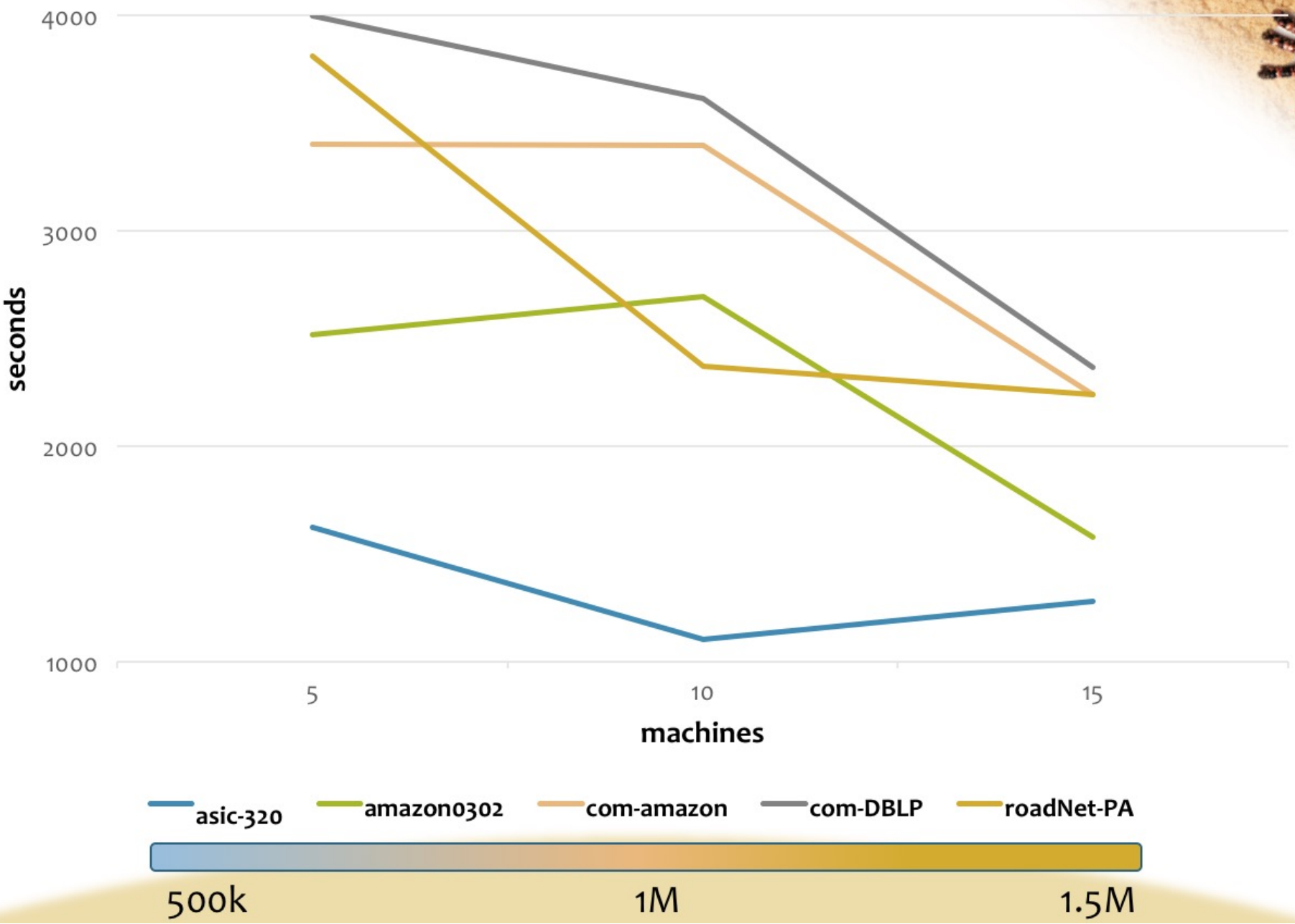
Multi-Gila



FM³

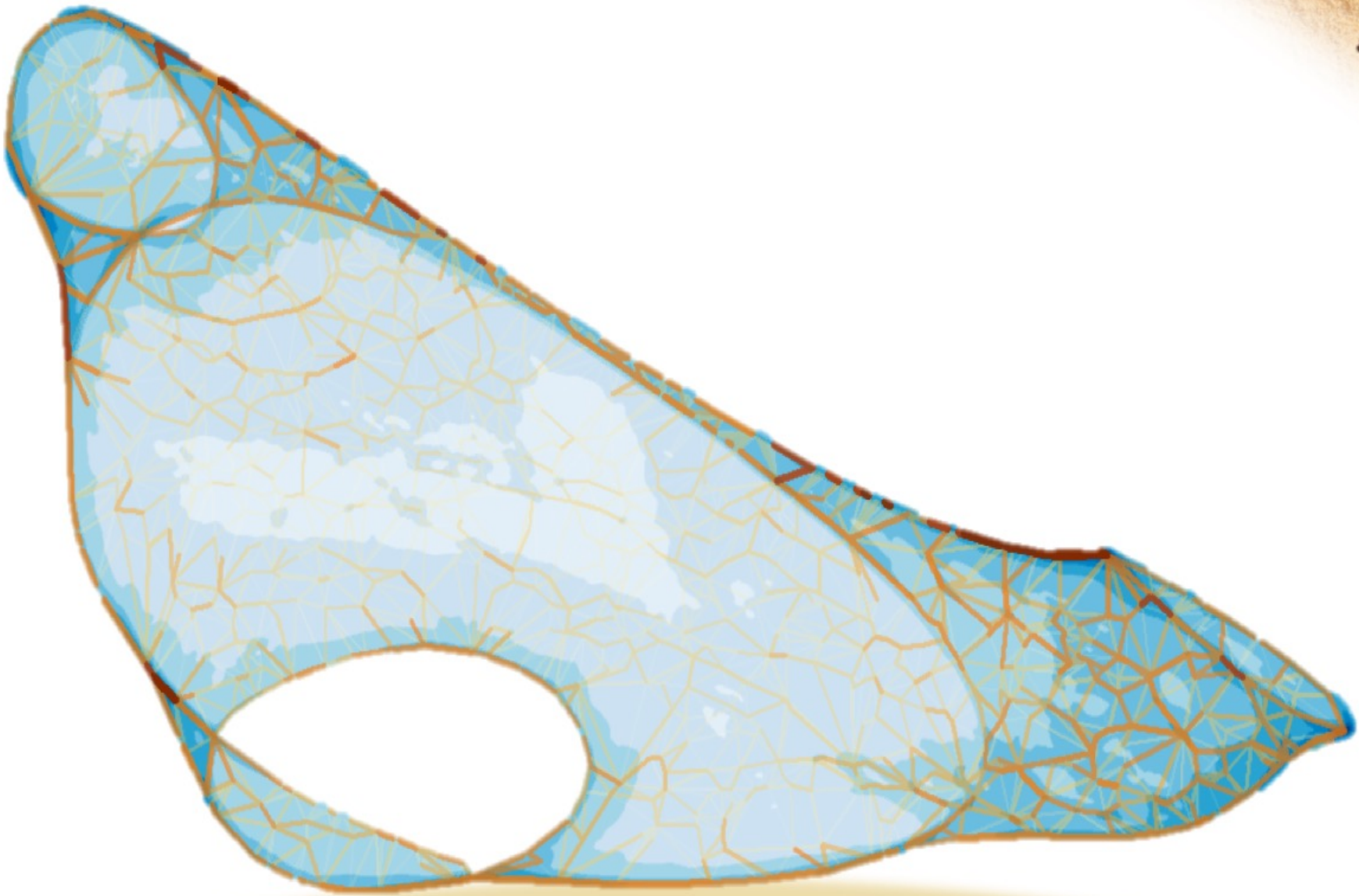


Medium-Sized Graphs Scalability



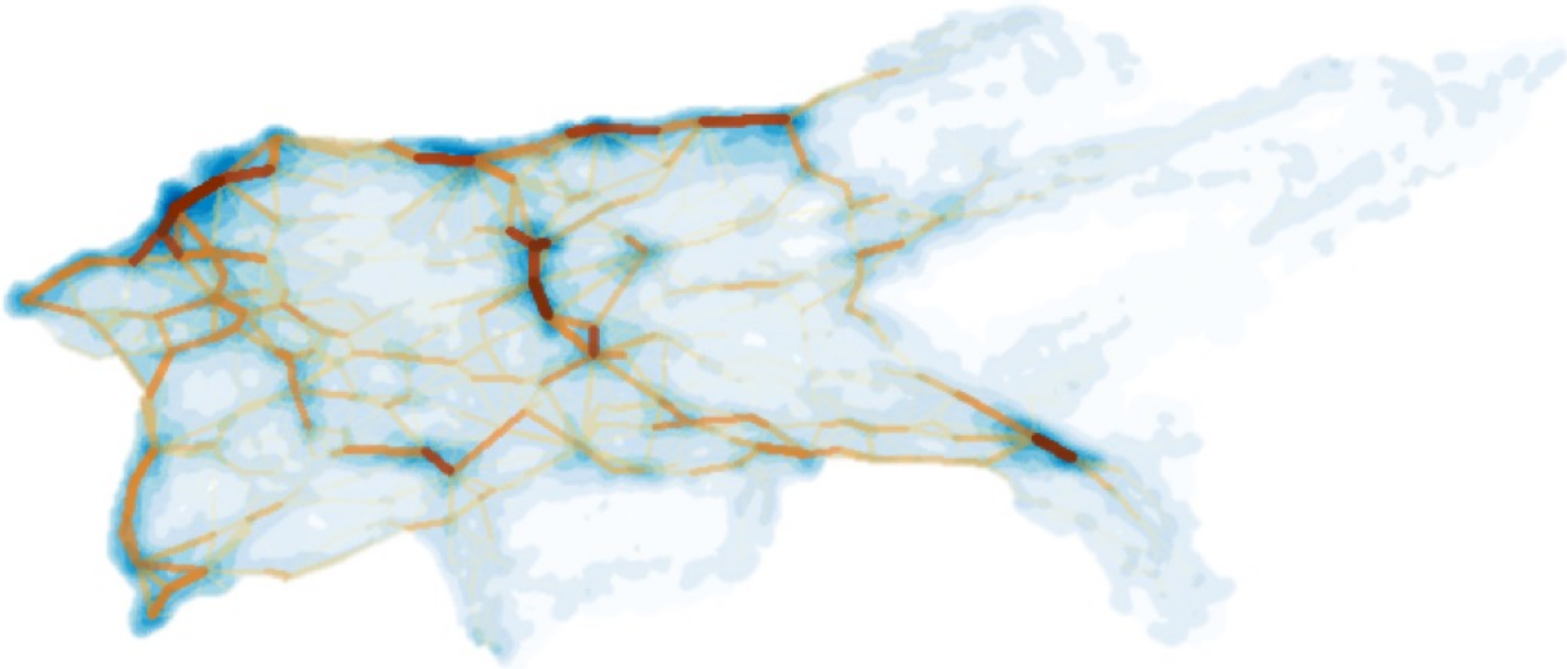
Applications: *Multi-Gila* and *Lago* ❖

❖ Brandes et al., 2012



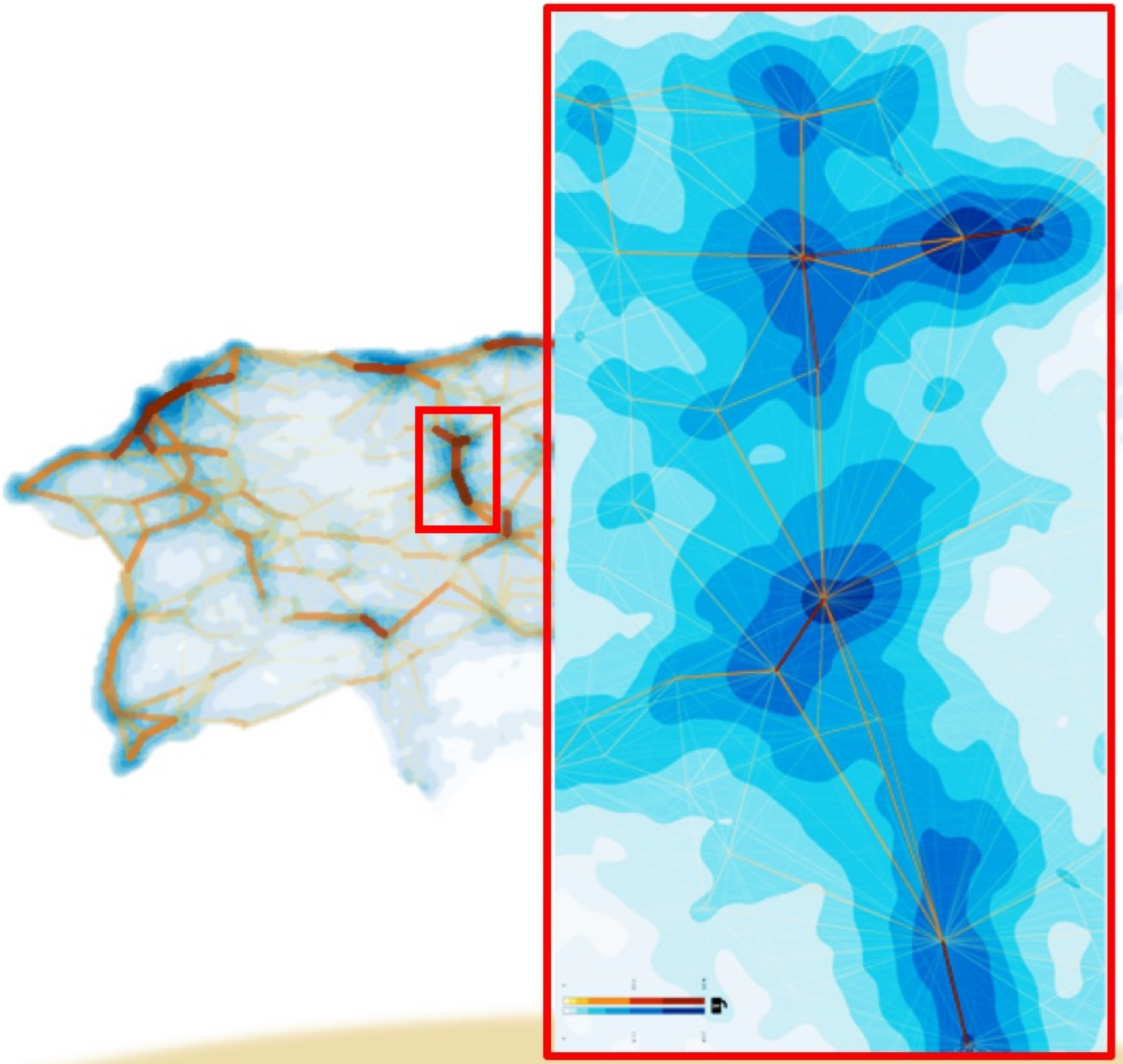
Hugetric-00020, 7M vertices, 11M edges

Applications: *Multi-Gila* and *Lago*



Asic-320, 121k vertices, 515k edges

Applications: *Multi-Gila* and *Lago*



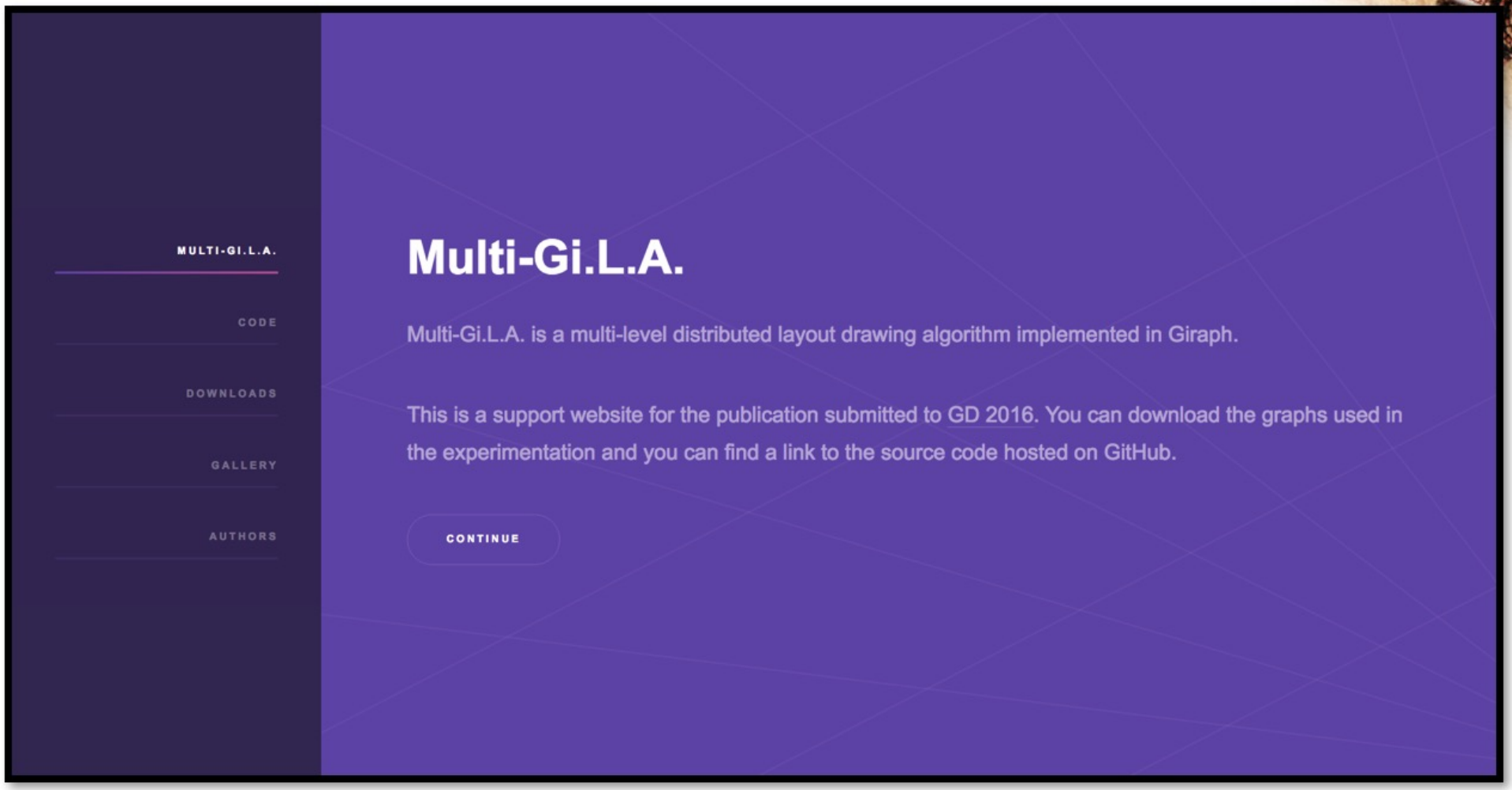
Asic-320, 121k vertices, 515k edges

Conclusions & Future Work

- Multi-Gila: the first distributed multilevel FD
  - Quality of drawings comparable to FM^3
  - Exhibits high scalability with large graphs
- Future work
  - New repulsive forces approximation techniques
  - New coarsening techniques

THANK YOU

Multi-Gila Website



http://multigila.graphdrawing.cloud