Revolver: Vertex-centric Graph Partitioning Using Reinforcement Learning

Mohammad Hasanzadeh Mofrad[1], Rami Melhem[1]
and Mohammad Hammoud[2]
[1]University of Pittsburgh
[2]Carnegie Mellon University Qatar

July 2, 2018

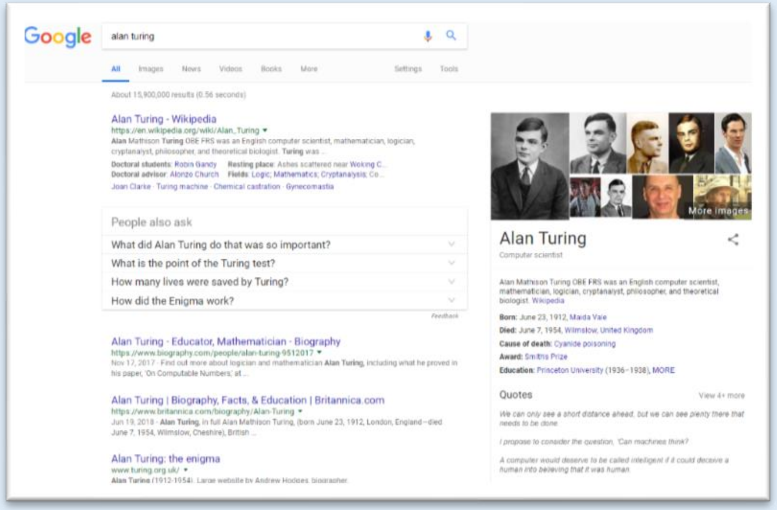
IEEE International Conference on Cloud Computing (CLOUD 2018)

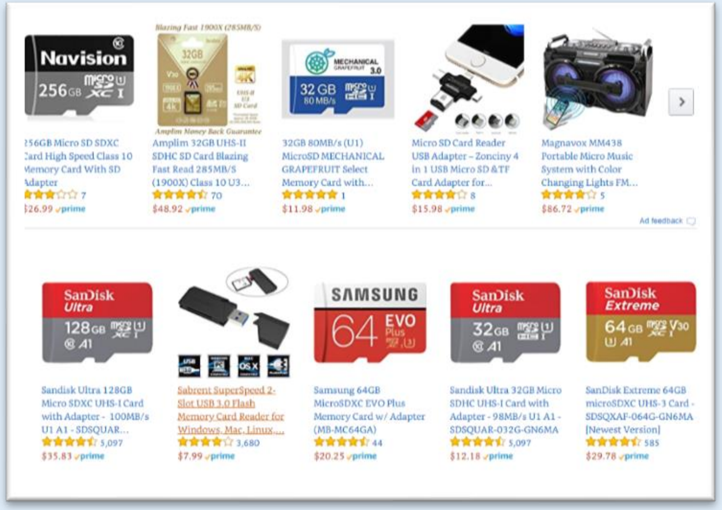
Discussion outline

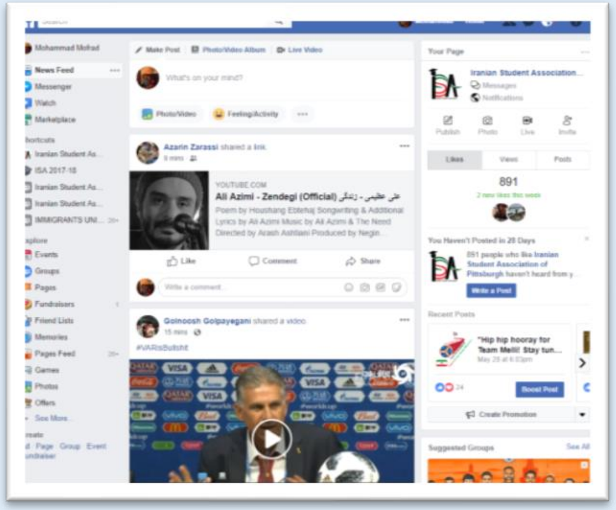
- **Motivation**
- Background
- Revolver
- Experiments

Motivation: Distributed Graph Analytics

1. **Distributed graph analytics** is the key to process big graphs.
   - Dividing a big graph into subgraphs
   - Distributing subgraphs across machines of the cluster
2. **Example applications**
   - Google PageRank
   - Facebook EdgeRank
   - Amazon Item Recommendation







Discussion outline

- Motivation
- **Background**
- Revolver
- Experiments

Background: K-way Balanced Graph Partitioning (Definition)

- Given a graph $G = (V, E)$ where V is the set of vertices and E is the set of Edges, k-way balanced graph partitioning divides the graph into k subgraphs

$$|E| / k\ (1 + \varepsilon)$$

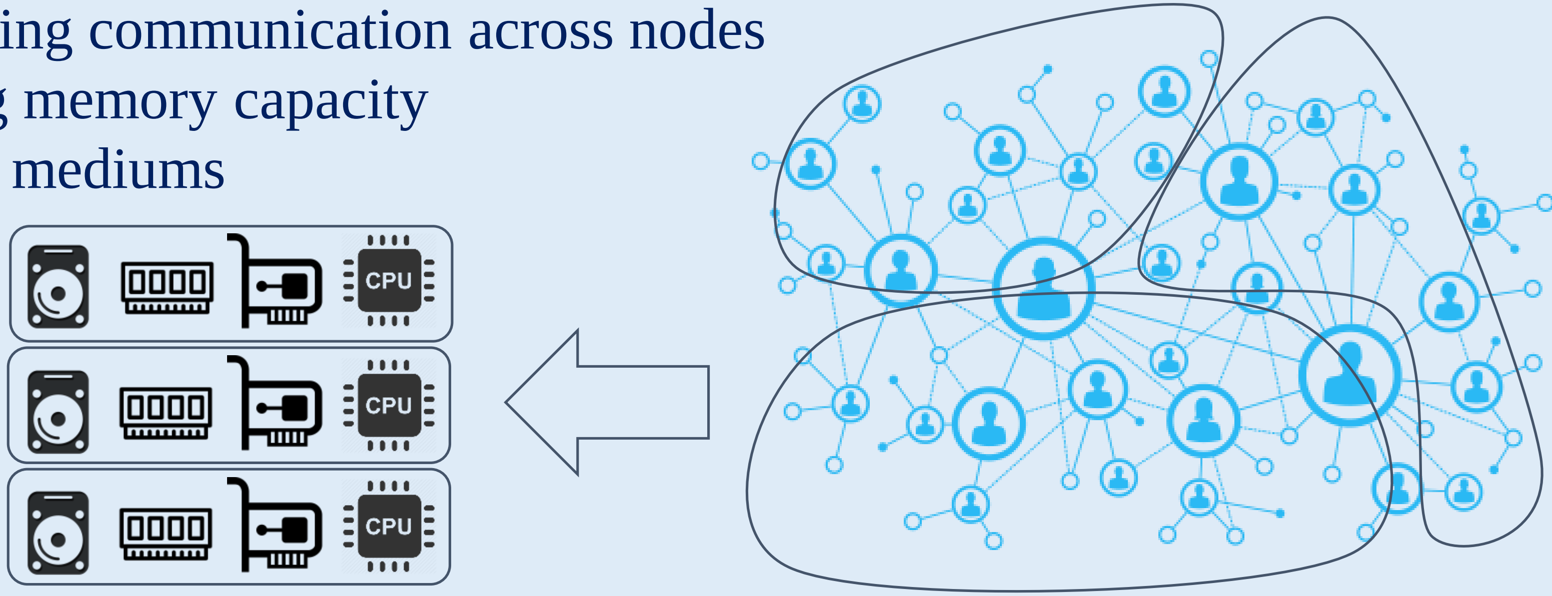
where k is the number of partitions

and $\varepsilon > 0$ is the imbalanced ratio

Mohammad Hasanzadeh Mofrad et al. “Revolver: Vertex-centric Graph Partitioning Using Reinforcement Learning.” IEEE CLOUD 2018.

Background: K-way Balanced Graph Partitioning (Goals)

- Work balance: Assigning partitions to nodes in the cluster
    1. *Computation*: Distributing computation among nodes
    2. *Communication*: Minimizing communication across nodes
    3. *Memory*: Avoid exceeding memory capacity
    4. *Storage*: Utilizing storage mediums



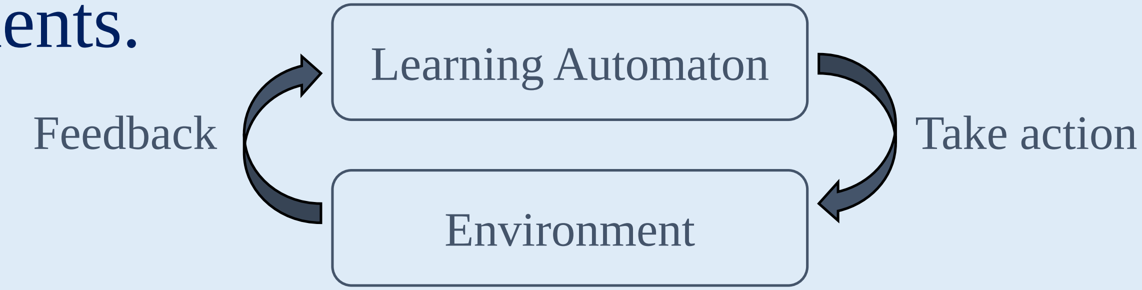
- Random and Hash partitioning algorithms have extremely *poor locality and cut-edge*

Background: Reinforcement Learning

- Reinforcement Learning is a class of machine learning algorithms inspired by stimulus-response principle.
- Examples
  - Deep Q-network: A reinforcement learning agent combined with deep neural networks capable of playing Atari 2600 games
  - AlphaGo: A game of Go player that beat a world Go champion. It has multiple neural networks trained by supervised learning from human expert moves and by reinforcement learning from self-play.

Background: Learning Automata

- Learning Automata (LA) are a subclass of reinforcement learning algorithm that select their new actions using their past experience with certain environments.



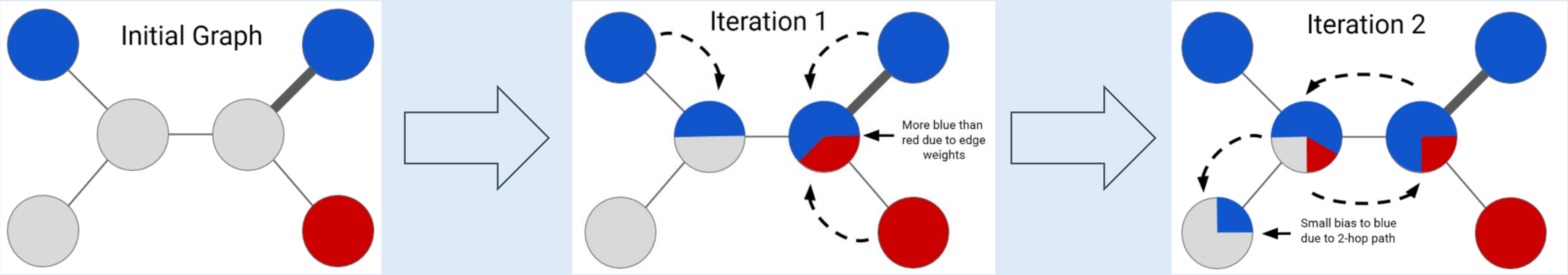
- A learning automaton is defined using the quadruple [A, P, R, T]
  - $A = \{a_1, \ldots, a_m\}$ is the action set with m being the number of actions
  - $P = \{p_1, \ldots, p_m\}$ is the probability vector
  - $R = \{0, 1\}$ is the set of reinforcement signal (Reward and penalty signals)
  - T is the linear learning algorithm where $\mathrm{P}(n + 1) = \mathrm{T}[A(n), P(n), R(n)]$

Mohammad Hasanzadeh Mofrad et al. “Revolver: Vertex-centric Graph Partitioning Using Reinforcement Learning.” IEEE CLOUD 2018.

Background: Label Propagation

- **Definition**: Label Propagation is a semi-supervised machine learning algorithm that assigns labels to the large set of unlabeled data using a small amount of labeled data.
- Let $v \in V$ then

$$Label(v) = Objective(v)$$



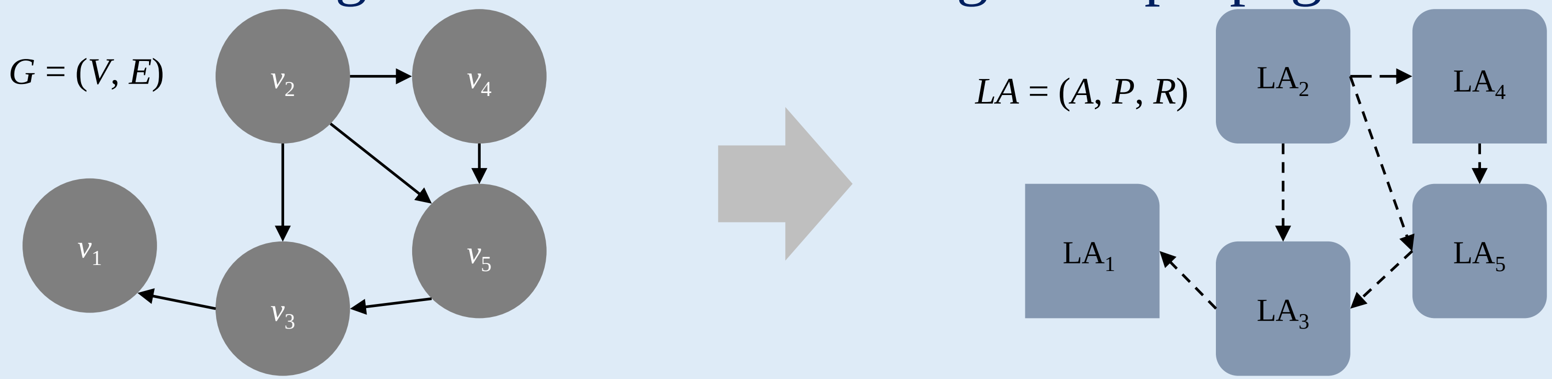
Label propagation on a graph borrowed from https://ai.googleblog.com/2016/10/graph-powered-machine-learning-at-google.html

Discussion outline

- Motivation
- Background
- **Revolver**
- Experiments

Revolver: Core Idea

- Partition $G = (V, E)$ with LA = (A, P, R) into k partitions
  - A network of LA analogous to G is created where |V| = |LA|
  - A learning automaton is assigned to each vertex v
  - Neighbor LAs can be queried using the set E
  - The action set A is the same as the set of available partitions, $|A| = k$
  - The probability vector P is initialized by $1/k$
  - The reinforcement signal R is calculated using label propagation



Revolver: Normalized Label Propagation

- Score(v, l): Computing score for l^{th} partition of vertex v

$$score(v,l) = \frac{1}{2}\left(\underbrace{\sum_{u \in N(v)} \frac{w(u,v)\delta(\psi(u),l)}{\sum_{u \in N(v)} w(u,v)}}_{\text{Weight term}} + \underbrace{\frac{1 - \frac{b(l_i)}{C}}{\sum_{i=1}^{k} 1 - \frac{b(l_i)}{C}}}_{\text{Penalty term}} \right)$$

Mohammad Hasanzadeh Mofrad et al. “Revolver: Vertex-centric Graph Partitioning Using Reinforcement Learning.” IEEE CLOUD 2018.

Revolver: Training Learning Automata

1. Action selection:
   - Actions are selected using the probability vector P
2. Scoring function:
   - For each partition of vertex v, a score is produced
3. Vertex migration:
   - Vertices migrate to their candidate (new) partition with a probability of migration
4. The reinforcement signal R is calculated as follows:
   - Reward signal: If $\psi(v)$ has the highest score or it has positive migration probability.
   - Penalty signal: Otherwise.
5. Probability update:
   - The probability vector is updated while taking account of the calculated signal.

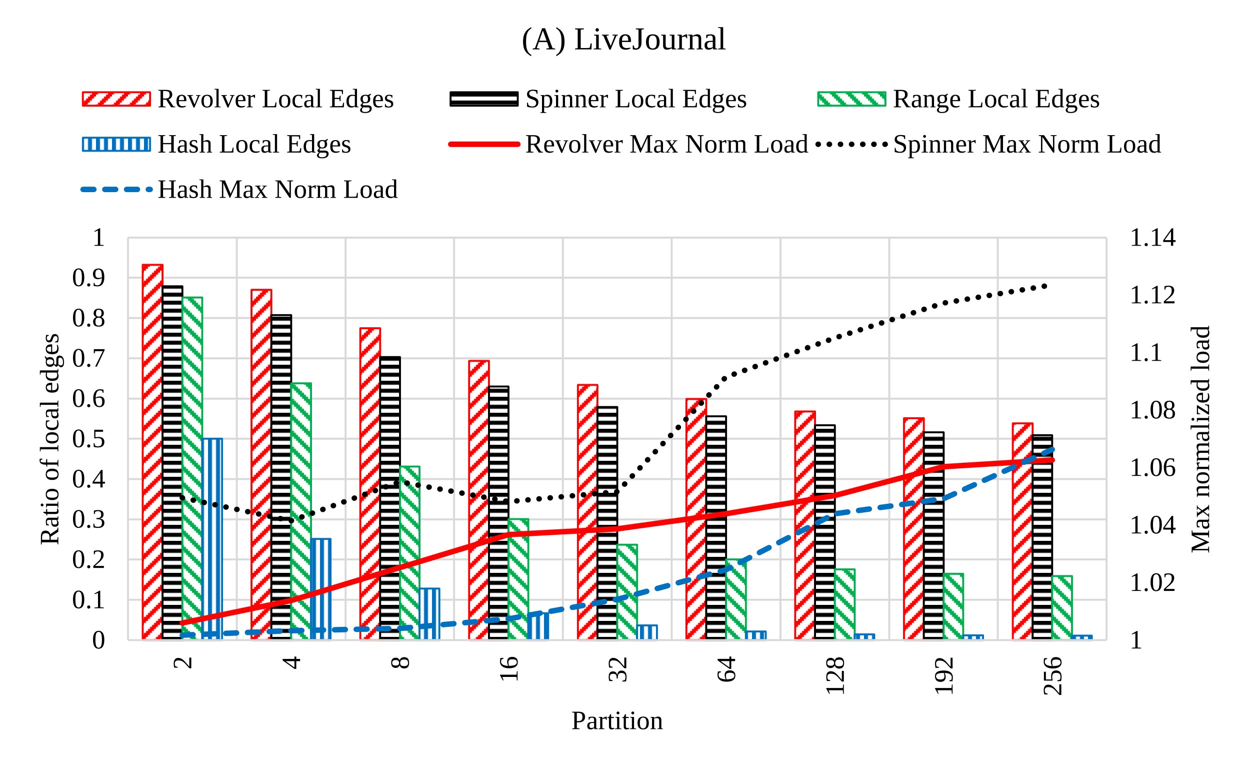
Discussion outline

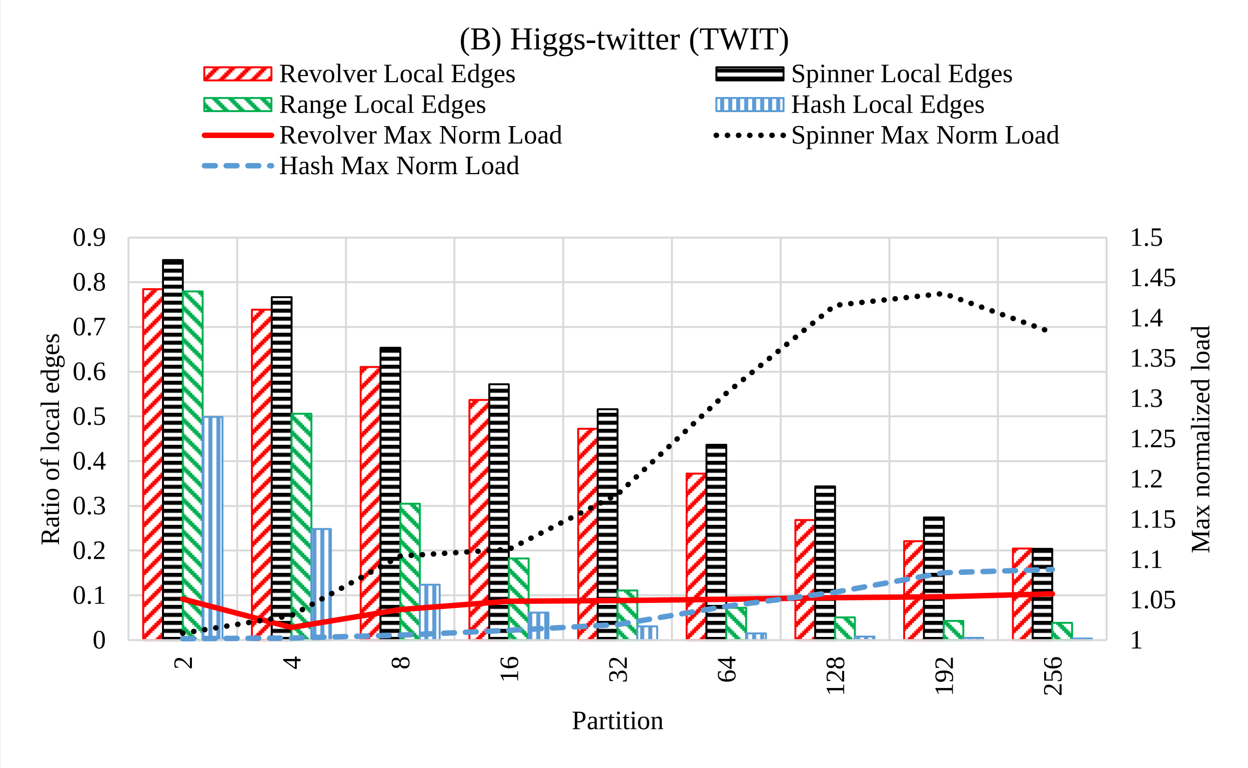
- Motivation
- Background
- Revolver
- **Experiments**

Experiments: Setup and Metrics

1. Performance metrics
   - *Ratio of local edges* = $|local\_edges| / |E|$
   - *Max normalized load* = $|max\_load| / (|E| / k)$
2. Datasets
   - LiveJournal (LJ): $|V| = 4.84$M and $|E| = 68.99$M
   - Higgs-twitter(TWIT): $|V| = 0.45$M and $|E| = 14.85$
3. Partitioning algorithms:
   - Revolver
   - Spinner
   - Hash
   - Range
4. Number of partitions:
   - 2, 4, 8, 16, 32, 64, 128 and 192
5. Imbalanced ratio $\varepsilon = 0.05$ for $|E|/k\ (1 + \varepsilon)$
6. Number of runs: 10

Experiments: Performance Results





Revolver produces the best *ratio of local edges* while not exceeding the *max normalized load*

Spinner produces the best *ratio of local edges* while exceeding the *max normalized load* (9x for 128 partitions)
Revolver does not exceed *max normalized load*

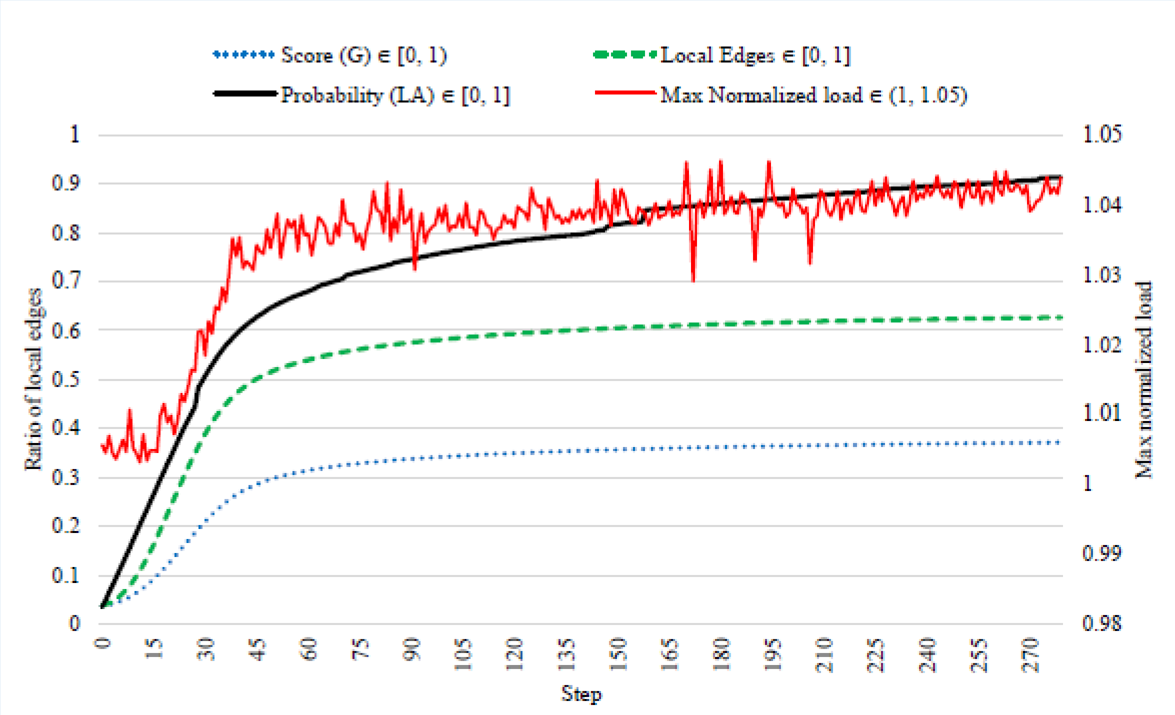
Experiments: Convergence Characteristics

- For a graph $G = (V, E)$,

$$score(G) = \frac{\sum_{v \in V} score(v)}{|V|}$$

- For a network of LA = (A, P, R),

$$probability(LA) = \sum_{la \in LA} \frac{P(\psi(la) > 0.9)}{|V|}$$



Mohammad Hasanzadeh Mofrad et al. "Revolver: Vertex-centric Graph Partitioning Using Reinforcement Learning." IEEE CLOUD 2018.

Conclusion

- Revolver, a graph partitioning algorithm
  - Vertex-centric
  - Parallel
  - Adaptive (LA)
- Learning Automata helps Revolver:
  - produce localized partitions (locality)
  - produce balanced partitions (scalability)

Thank you!

"Those who can imagine anything, can create the impossible."
— Alan Turing