Scalable Algorithms for Correlation Clustering on Large Graphs

Nathan Cordner

Boston University

26 May 2022

Outline

- Correlation Clustering and the Pivot Algorithm
- **Previous Work**
  - Scalable Cluster Improvement
  - Scalable Consensus Clustering
- **Proposed Work**
  - Scalable Algorithms for
    - Constrained Cluster Sizes
    - Constrained Number of Clusters
  - Proportional Fairness for Scalable Algorithms

Correlation Clustering [BBC04]

Given a complete graph $G = (V, E)$

- $E = E^+ \cup E^-$

Want to cluster $+$ edges and separate $-$ edges

- Maximize Agreements
- **Minimize Disagreements**

Some Applications

- Classification
- Entity Resolution
- Friend Groups in Social Networks

The Pivot Algorithm [ACN08]

Pivot$(V, E = E^+ \cup E^-)$:

- Pick random pivot node $u \in V$
- Set $C = \{u\}$
- For all $v \in V \setminus \{u\}$:
  - If $\{u, v\} \in E^+$: Add v to C
- Repeat on $V = V \setminus C$ until empty
- Return completed clustering

Runs in $O(|V| + |E|)$ time

Randomized expected 3-approximation

Efficient Implementation of Pivot [AL09]

Neighborhood Oracle $N(u) = \{v \in V \mid \{u, v\} \in E^+\}$

Pivot$(V, E = E^+ \cup E^-)$:

- Pick random pivot node $u \in V$
- Set $C = \{u\}$
- For all $v \in N(u)$:
  - If $v \in V$: Add v to C
- Repeat on $V = V \setminus C$ until empty
- Return completed clustering

Runs in $O(|V| + |E|^+)$ time

Weighted CC [BBC04, ACN08]

Every pair of nodes u, v has weights $w_{uv}^+, w_{uv}^- \geq 0$

- Clustering Cost:

$$\sum_{u,v \text{ in different clusters}} w_{uv}^+ + \sum_{u,v \text{ in same cluster}} w_{uv}^-$$

Given $G = (V, E, w)$

- Form the unweighted *majority instance* G_w
  - Place $\{u, v\}$ in E_w^+ if $w_{uv}^+ > w_{uv}^-$
  - Place $\{u, v\}$ in E_w^- if $w_{uv}^- > w_{uv}^+$
  - Break ties arbitrarily
- Run Pivot on $G_w = (V, E_w = E_w^+ \cup E_w^-)$

Weighted CC [ACN08, KPT11]

Probability Constraints: $w^+_{uv} + w^-_{uv} = 1$

- Notation: $p(u,v) = w^+_{uv}$, $1 - p(u,v) = w^-_{uv}$

Relation to original CC problem

- $\{u,v\} \in E^+ \Leftrightarrow p(u,v) = 1$
- $\{u,v\} \in E^- \Leftrightarrow p(u,v) = 0$

Pivot Approximation Results

- 5-approx with probability constraints
- 2-approx with PC and the triangle inequality

Other Algorithms for CC

LP rounding methods

- 2.5-approx for probability weights [ACN08]
- 2.06-approx for $0/1$ weights [CMSY15]
- Run time dominated by LP solver
- Some optimizations exist though (e.g. [HYY21])

Pivot still most efficient for large graphs

One Algorithm to Rule Them All?

Pivot has been successfully used to cluster

- Social network graphs [KPT11]
- Protein-protein interaction graphs [KPT11; HWH15]
- Event graphs generated from news stories [CMB17]

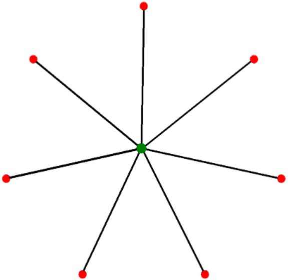
Pivot has been adapted for

- Probabilistic graphs [KPT11; MTG20]
- Chromatic correlation clustering [KSZC21]
- Fair correlation clustering [AEKM20]
- Data streaming and online settings [ACGM15; LMVW21]
- Query constraints [GKBT20]
- Cluster size constraints [PM15]

Deterministic and parallel versions [ZW09; CDK14; PORJ15]

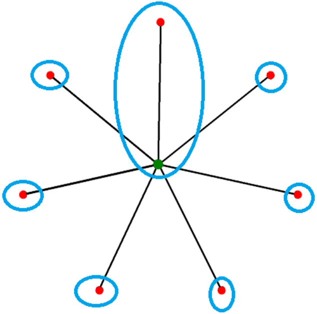
Drawbacks of Pivot

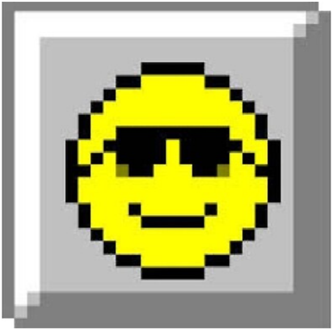
Pivot performs poorly on star graphs



Drawbacks of Pivot

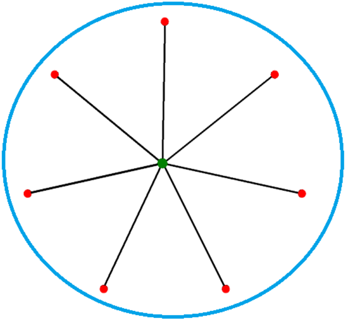
Pivot performs poorly on star graphs





Drawbacks of Pivot

Pivot performs poorly on star graphs



Can we do better and still maintain scalability?

The RandomNode Algorithm*

Pick unclustered nodes one at a time

- First node creates its own cluster
- All others: add to existing cluster, or create own
- Greedily minimize increase in clustering cost

Previous Results

- Experimentally better than Pivot (e.g. [ES09])
- Much slower though

* Inspired by Node algorithm for reducing oracle queries in entity resolution [VBD14]; also known as the Vote algorithm [ES09]

The RandomNode Algorithm

Running Time [GMT07]

- S = previously settled nodes; u = current node
- Cost of creating new cluster: $\sum_{v \in S} p(u,v)$
- Cost of adding to existing cluster C:

$$\sum_{v \in C}(1-p(u,v)) + \sum_{v \in S \setminus C} p(u,v) = |C| + \sum_{v \in S} p(u,v) - 2\sum_{v \in C} p(u,v)$$

- $\Theta(|V|^2) = \Theta(|V| + |E|)$ for weighted graphs

Neighborhood Oracle

- Only consider clusters with positive edges from u
- Reduces to $\Theta(|V| + |E|^+)$ time for 0/1 graphs

The RandomNode Algorithm

Node-at-a-Time Pivot [BGK13]

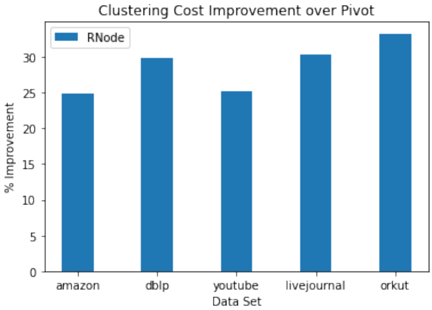
- Initialize list P and pick nodes in random order
- Join cluster of first pivot in P with positive edge
- Otherwise start new cluster and add self to end of P

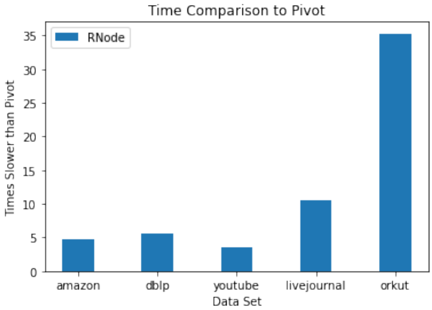
Clustering Cost: RandomNode "stays ahead" of Pivot

- On same node order, expected cost of each RandomNode decision $\leq$ expected cost of Pivot
- RandomNode inherits same guarantees as Pivot
- Justifies using $0/1$ graphs for weighted instances

The RandomNode Algorithm

Example: RandomNode gives better results, but Pivot runs significantly faster on large instances





`snap.stanford.edu/data/#communities`

Cluster Improvement

Local Search: given a clustering,

- Each node decides whether to stay or move clusters
- Iterate until improvements stop
- Slow (each iteration is like RandomNode)
- Somewhat popular though [MTG20; AEKM20]

New Idea: Use RandomNode *inside* Pivot clusters

- Method 1: use node ordering given by Pivot
- Method 2: "Deterministic" ordering [VBD14]
  - Maximizes expected cost improvement inside clusters

Cluster Improvement

Expected Cluster Size [VBD14]: $\mathsf{ECS}(u) = \sum_{v \in V \setminus \{u\}} p(u, v)$

- $O(|V|^2)$ time to compute for weighted graphs
- Equals $|N(u)|$ for 0/1 graphs; $O(|V|)$ time

DeterministicNode

- Order nodes by ECS: $O(|V| \log |V|)$ time
- Follow RandomNode with ECS node order

The Hybrid Algorithm

Hybrid Algorithm: on graph G

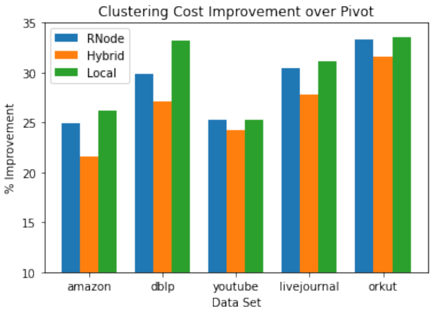
- Obtain clusters $C_1, \ldots, C_k$ from Pivot(G)
- Let G_i be the graph induced by C_i
- Return DNode$(G_1), \ldots,$ DNode(G_k)

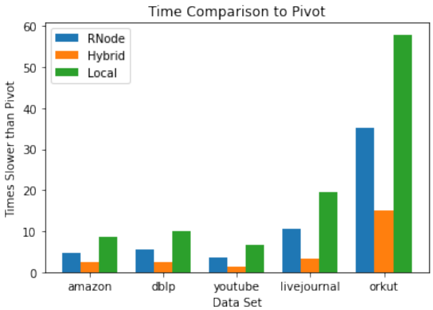
Properties

- Nearly linear running time: $O(|V| \log d + |E|^+)$
  - d is size of largest Pivot cluster
- Easily run in parallel
- Improves cluster costs from Pivot
- **Approximation Bound**: stay tuned!

The Hybrid Algorithm

Example: Hybrid gives nearly the same improvement as RandomNode, but in about half the extra time over Pivot

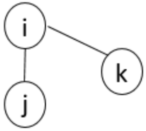




`snap.stanford.edu/data/#communities`

The Hybrid Algorithm

Theoretical Improvements



"Bad Triangles": i, j, k unclustered

- Two edges are $+$ but one is $-$

Lemma [ACN08]: Approx bound of Pivot $\leq$ worst cost ratio for bad triangles

- Triangle completely inside cluster when i is chosen as pivot ($1/3$ chance)
- **Claim**: Hybrid reduces average cost of bad triangles inside Pivot clusters
- Hybrid approximation bound:

$$3((1/3)(\textbf{hybrid triangle cost}) + 2/3)$$

The Hybrid Algorithm

Given Pivot cluster C with $m = |C|(|C| - 1)/2$ edges

- Expected number of $+$ edges in C: $m/2$
- Expected Pivot cluster cost: $m/2$

DeterministicNode: order nodes by degree (ECS)

- $m/4$ edges used to cluster first half of nodes
- $3m/4$ edges used to cluster second half of nodes
- DNode cost $\leq$ putting first half of nodes into one cluster and separating all remaining nodes
- DNode cost $\leq (2/28 + 9/28)m = 11m/28$
- Cost ratio: 11/14; Bound: $39/14 \approx 2.786$

The Hybrid Algorithm

Given Pivot cluster C with $m = |C|(|C|-1)/2$ edges

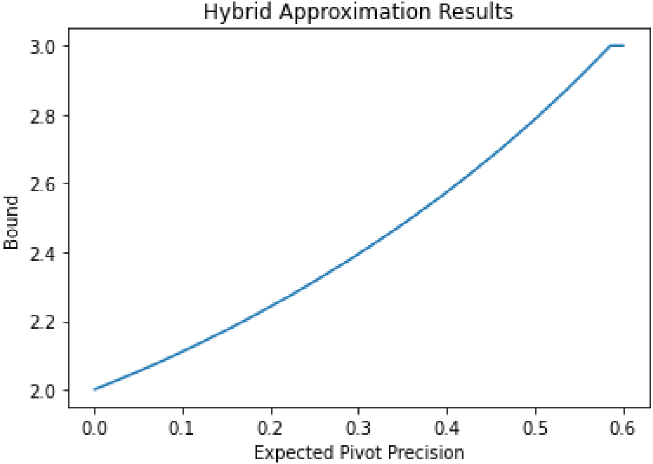
- Expected number of $+$ edges in C: pm
- Expected Pivot cluster cost: $(1-p)m$

DeterministicNode: order nodes by degree (ECS)

- p^2m edges used to cluster first pn nodes
- $(1-p^2)m$ edges used to cluster remaining nodes
- DNode cost $\leq$ putting first pn nodes into one cluster and separating all remaining nodes
- DNode cost $\leq \left[p + \frac{p^2}{2}\left(\frac{1-2p+p^2/2}{1-p^2/2} - 1\right)\right] m$
    - Ratio < 1 when $p < 2 - \sqrt{2} \approx 0.586$

The Hybrid Algorithm

Precision: ratio of positive edges inside clusters to total number of edges inside clusters



The Hybrid Algorithm

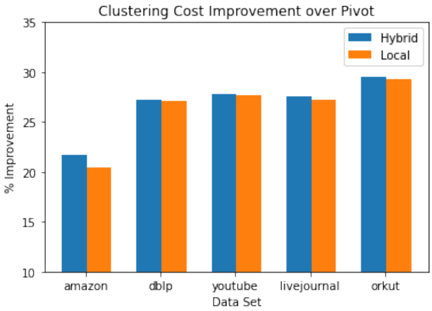
Example Revisited:

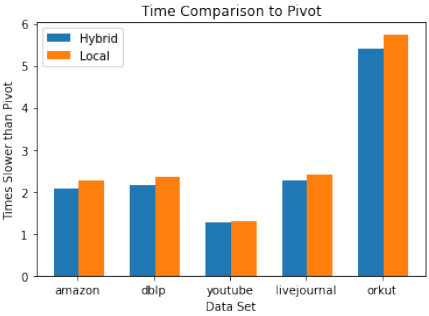
Data Set	Pivot Precision (%)	Hybrid Bound
DBLP	55.53	2.92
Amazon	50.93	2.807
LiveJournal	28.69	2.373
YouTube	24.39	2.305
Orkut	13.42	2.152

The Hybrid Algorithm

Alternative Approach: run Local Search once inside each cluster using *reverse* ECS order

- Has the same guarantees as Hybrid





Consensus Clustering

Given clusterings $\mathcal{C}_1 \ldots, \mathcal{C}_k$ of node set V

- Find clustering $\mathcal{C}$ minimizing $\sum_{i=1}^{k} \text{Disagree}(\mathcal{C}, \mathcal{C}_i)$
- $\text{Disagree}(\mathcal{C}, \mathcal{C}_i) =$ number of node pairs (u, v) clustered together in only one input clustering

Relation to Correlation Clustering

- $p(u, v) =$ number of input clusterings where u, v are clustered together divided by k
- Edge weights satisfy the triangle inequality

Consensus Clustering

Previous Pivot Problems [ACN08; GF08]

- Time inefficiency: $O(k|V|^2)$ to compute all edges
- Space inefficiency: $O(|V|^2)$ to store all edges

Improvement # 1: only compute edges as needed

- Precompute cluster labels for each node

Improvement # 2: reduce number of input clusterings

- Picking one input clustering at random: 2-approx
- Pivot on full set of inputs: 1.57-approx [ACN08]
- What about in between?

Consensus Clustering

Improvement 1 Example: Mushrooms

- 23 input clusterings, 8124 nodes
- Edges: 71.6 s; average Pivot run: 0.0091 s
- Labels: 0.03 s; average Pivot run: 0.04s

`archive.ics.uci.edu/ml/datasets/mushroom`

Consensus Clustering

Sampling input clusterings

- Assume k large and sample $R < k$ input clusterings
- $p =$ true probability u, v are clustered together (assume $p < 1/2$)
- Let $X =$ number of sampled clusters where u, v are clustered together
- Model X as a Binomial rv with R trials and success probability p
- Pivot algorithm "makes a mistake" when $X > R/2$

Consensus Clustering

What is $\mathbb{P}(X > R/2)$?

- Normalize: $Z = (X - pR)/\sqrt{Rp(1-p)}$
- Estimate $\mathbb{P}(X > R/2)$ using

$$\mathbb{P}\left(Z > \frac{R/2 - pR}{\sqrt{Rp(1-p)}}\right) = \mathbb{P}\left(Z > \frac{\sqrt{R}(1/2 - p)}{\sqrt{p(1-p)}}\right)$$

- Let $f(R,p) = \sqrt{R}(1/2 - p)/\sqrt{p(1-p)}$
- Find $\mathbb{P}(Z > f(R,p))$ by evaluating

$$\mathrm{Err}(R,p) := 1 - \Phi(f(R,p)),$$

where Φ is the standard normal CDF

Consensus Clustering

Lemma: the expected cost multiple of edge (u, v) due to error in a Pivot clustering is

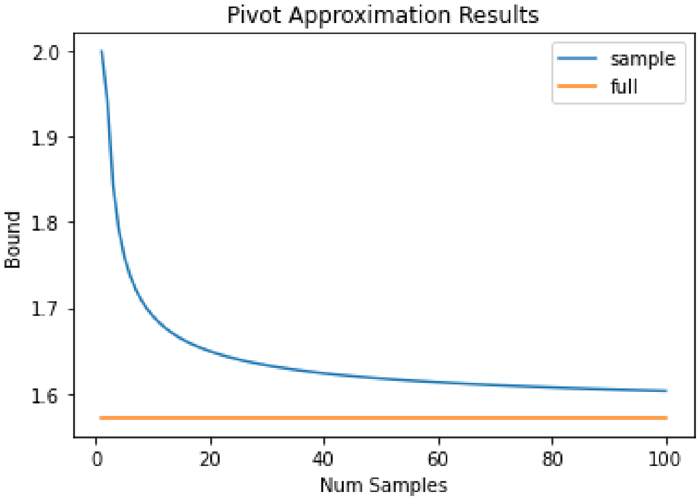
$$p \cdot (1 - \mathsf{Err}(R, p)) + (1 - p) \cdot \mathsf{Err}(R, p)$$

Cost multiple upper bound:

$$g(R) = \max_{p \in [0,1/2]} \{[p \cdot (1 - \mathsf{Err}(R, p)) + (1 - p) \cdot \mathsf{Err}(R, p)]/p\}$$

Consensus Clustering

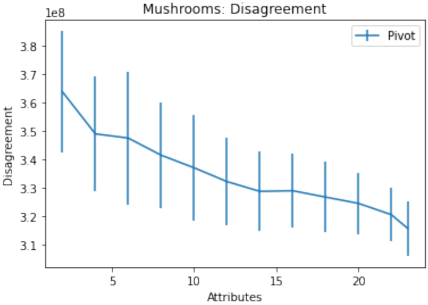
Theorem: Pivot is a $(6g(R)+5)/7)$-approx algorithm

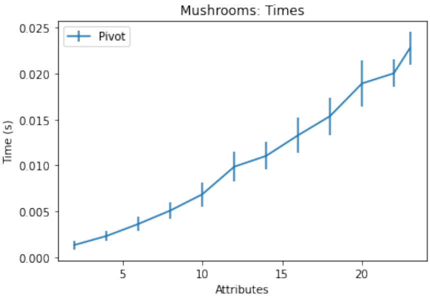


Consensus Clustering

Improvements 1 and 2 Example: Mushrooms

- 23 input clusterings, 8124 nodes
- Largest disagreement is 1.153 times smallest

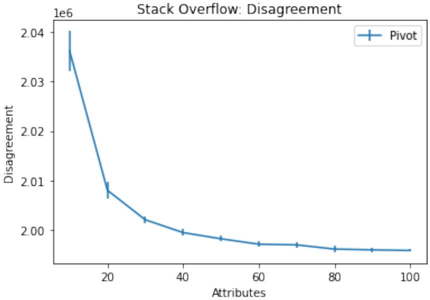


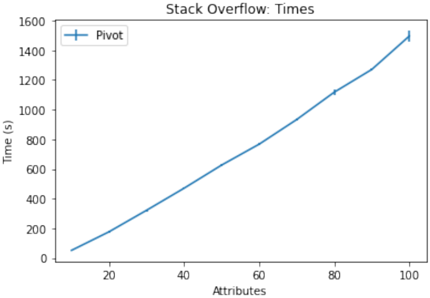


Consensus Clustering

Improvements 1 and 2 Example: Stack Overflow

- 100 input clusterings, 14284 nodes
- Largest disagreement is 1.02 times smallest





ics.uci.edu/∼duboisc/stackoverflow/

Outline

- ~~Correlation Clustering and the Pivot Algorithm~~
- ~~**Previous Work**~~
  - ~~Scalable Cluster Improvement~~
  - ~~Scalable Consensus Clustering~~
- **Proposed Work**
  - Scalable Algorithms for
    - Constrained Cluster Sizes
    - Constrained Number of Clusters
  - Proportional Fairness for Scalable Algorithms

Constrained Cluster Sizes

Uniform: given $K \geq 1$, all clusters must have size $\leq K$

- LP rounding algorithms
  - 6-approx [PM15]
  - 5.37-approx [JCTZ21]
- Pivot adaptations [PM15]
  - 7-approx by removing a smallest set of $+$ edges
  - 11-approx for random removal (**Constrained Pivot**)

Non-Uniform: size limit defined for every node [JXLW20]

- LP $2U$-approximation where $U =$ max node limit

Constrained Number of Clusters

Proposed Work

- New analysis for Constrained Pivot
  - Based on new proof technique [KSZC21]
  - **Claim**: 3-approx for both uniform and non-uniform cases
- Hybrid analysis for constrained cluster sizes
- Compare to Constrained RandomNode

Constrained Number of Clusters

Given k, find clustering with at most k clusters

- $k = 2$:
  - Pivot-like 3-approximation [BBC04]
  - Local search 2-approximation [CSW08]
  - Neither generalizes well for $k > 2$
- General case: $(1 + \epsilon)$ PTAS [GG06]
  - *Extremely* inefficient: $|V|^{O(9^k/\epsilon^2)} \log |V|$ running time
  - Still used from time to time [ACGM15; BEK21]

Constrained Number of Clusters

Proposed work

- k-RandomNode
  - **Claim**: 7-approximation algorithm
- k-Hybrid: form k Pivot clusters, finish with k-RNode
- Compare with new Pivot-like algorithms for k-CC

Fair Correlation Clustering

"Balanced" Fairness

- Colors assigned to every node; proportion of colors inside clusters must match overall proportion
- Algorithms using fairlet decomposition [AEKM20]
- LP improvements for some cases [FM21]

Other fairness definitions have yet to be considered for correlation clustering

Fair Correlation Clustering

Proposed work: Proportionally Fair CC

- k-(means, medians, centers): no set of $\geq |V|/k$ nodes prefers to be clustered together [CFLM19]
- Extend proportional fairness to correlation clustering
- Analyze fairness in Pivot and other unconstrained CC algorithms
- Analyze fairness in k-CC algorithms

References

AEKM20 Ahmadian, Epasto, Kumar, and Mahdian. *Fair correlation clustering*. 2020

ACGM15 Ahn, Cormode, Guha, McGregor, and Wirth. *Correlation clustering in data streams*. 2015

ACN08 Ailon, Charikar, and Newman. *Aggregating inconsistent information: ranking and clustering*. 2008

AL09 Ailon and Liberty. *Correlation clustering revisited: the "true" cost of error minimization problems*. 2009

BBC04 Bansal, Blum, and Chawla. *Correlation clustering*. 2004

BGK13 Bonchi, García-Soriano, and Kutzkov. *Local correlation clustering*. 2013

BEK21 Bun, Elias, and Kulkarni. *Differentially private correlation clustering*. 2021

CMSY15 Chawla, Makarychev, Schramm, and Yaroslavtsev. *Near optimal lp rounding algorithm for correlation clustering on complete and complete k-partite graphs*. 2015

CFLM19 Chen, Fain, Lyu and Munagala. *Proportionally fair clustering*. 2019

CDK14 Chierichetti, Dalvi, and Kumar. *Correlation clustering in mapreduce*. 2014

CMB17 Christiansen, Mobasher, and Burke. *Using uncertain graphs to automatically generate event flows from news stories*. 2017

References

CSW08 Coleman, Saunderson, and Wirth. *A local-search 2-approximation for 2-correlation-clustering*. 2008

ES09 Elsner and Schudy. *Bounding and comparing methods for correlation clustering beyond ILP*. 2009

FM21 Friggstad and Mousavi. *Fair correlation clustering with global and local guarantees*. 2021

GKBT20 García-Soriano, Kutzkov, Bonchi, and Tsourakakis. *Query-efficient correlation clustering*. 2020

GMT07 Gionis, Mannila, and Tsaparas. *Clustering aggregation*. 2007

GG06 Giotis and Guruswami. *Correlation clustering with a fixed number of clusters*. 2006

GF08 Goder and Filkov. *Consensus clustering algorithms: comparison and refinement*. 2008

HWH15 Halim, Waqas, and Hussain. *Clustering large probabilistic graphs using multi-population evolutionary algorithm*. 2015

HYY21 Hua, Yu, and Yang. *Star-based learning correlation clustering*. 2021

JCTZ21 Ji, Cheng, Tan, and Zhao. *An improved approximation algorithm for capacitated correlation clustering problem*. 2021

References

JXLW20 Ji, Xu, Li, and Wang. *Approximation algorithms for two variants of correlation clustering problem*. 2020

KSZC21 Klodt, Seifert, Zahn, Casel, Issac, and Friedrich. *A color-blind 3-approximation for chromatic correlation clustering and improved heuristics*. 2021

KPT11 Kollios, Potamias, and Terzi. *Clustering large probabilistic graphs*. 2011

LMVW21 Lattanzi, Moseley, Vassilvitskii, Wang, and Zhou. *Robust online correlation clustering*. 2021

MTG20 Mandaglio, Tagarelli, and Gullo. *In and out: optimizing overall interaction in probabilistic graphs under clustering constraints*. 2020

MSS10 Mathieu, Sankur, and Schudy. *Online correlation clustering*. 2010

PORJ15 Pan, Papailiopoulos, Oymak, Recht, Ramchandran, and Jordan. *Parallel correlation clustering on big graphs*. 2015

PM15 Puleo and Milenkovic. *Correlation clustering with constrained cluster sizes and extended weights bounds*. 2015

VBD14 Vesdapunt, Bellare, and Dalvi. *Crowdsourcing algorithms for entity resolution*. 2014

ZW09 Zuylen and Williamson. *Deterministic pivoting algorithms for constrained ranking and clustering problems*. 2009