SIGMOD 2017 (author)

Association for Computing Machinery
Advancing Computing as a Science & Profession

Help / Log out

Submission 574 | SIGMOD 2017 | Support | News | EasyChair

SIGMOD 2017 Submission 574

Submission information updates are disabled.

For all questions related to processing your submission you should contact the conference organizers. Click here to see information about this conference.

All **reviews sent to you** can be found at the bottom of this page.

Paper 574	
Title:	Graph Pattern Matching for Dynamic Team Formation
Paper:	
Track:	SIGMOD 2017 Second round of submissions
Author keywords:	Graph Pattern Matching Team Formation Dynamic Team Formation Incremental algorithms
EasyChair keyphrases:	data update (716), top k team formation (540), pattern update (490), team formation (466), pattern graph (340), graph pattern matching (332), data graph (310), top k team (293), team simulation (280), graph simulation (270), perfect subgraph (252), match relation (250), maximum match relation (237), auxiliary structure (210), partial match relation (206), dynamic top k team (200), incremental algorithm (196), team formation problem (174), capacity constraint (140), simultaneous pattern (130), graph pattern (120), top k perfect subgraph (120), capacity bound (120), edge insertion (120), continuous pattern (110), batch mindia minsum (110), batch cit (110), type code (100), unit pattern update (95), auxiliary data structure (95)
Topics:	Graph data management, RDF, social networks
Abstract:	Finding a list of k teams of experts, referred to as top-k team formation, with the required skills and high collaboration compatibility has been extensively studied. However, existing methods do not consider the specific collaboration relationships among team members, i.e., structural constraints, which are typically needed in practice. In this paper, we first propose a novel graph pattern matching approach for top-k team formation, which incorporates structural constraints and capacity bounds, and develop a cubic algorithm with two optimization techniques. We then formulate and study top-k team formation in a dynamic setting, and develop an incremental algorithm with an optimization to handle continuous pattern and data updates, separately and simultaneously. Using real-life and synthetic data, we finally demonstrate the effectiveness and efficiency of our graph pattern matching approach for (dynamic) top-k team formation.
Time:	Nov 04, 12:00 GMT
Author conflicts:	Nan Tang, Haixun Wang (view)

Authors

first name	last name	email	country	organization	Web page	corresponding?
Shuai	Ma	mashuai@act.buaa.edu.cn	China	Beihang University	http://mashuai.buaa.edu.cn/	✓
Jia	Li	lijia@act.buaa.edu.cn	China	Beihang University		✓
Chunming	Hu	hucm@buaa.edu.cn	China	Beihang University	http://act.buaa.edu.cn	✓
Xudong	Liu	liuxd@buaa.edu.cn	China	Beihang University		✓
Jinpeng	Huai	huaijp@buaa.edu.cn	China	Beihang University,Beijing,China	http://www.buaa.edu.cn	

Reviews

Review 3

Overall evaluation:	**-1**
Reviewer's confidence:	**3**

Other Scores

Originality:	**3**
Correctness:	**3**
Completeness:	**2**
Best paper award:	-

Review

Strong Points:	S1: Comprehensive treatment of using pattern matching as a way to solve the team formation problem, in both the static and dynamic variations.
Weak Points:	W1: My main objection is that the problem of team formation is a variant of graph pattern matching, so it is not clear that we need completely new algorithms to solve this problem. I would propose the following approach: 1- Use a graph database or graph processing system such as Neo4j, Triad, or GraphFrames to find the desired team patterns in the graph. 2- Apply a post-processing step on the retrieved patterns to eliminate the ones that do not satisfy constraints on capacity, radius, or density. 3- Rank the remaining patterns and keep the top-k. This approach is not specialized so it may not be as fast as the one proposed in the paper. But it eliminates the need for a specialized "team formation" operator. W2: The paper relies heavily on [10] and can be viewed as an application of that paper in the specific domain of team formation.
Review:	D1: What is the difference between a "ball" as defined by the paper, and a "neighborhood", which is a more commonly used term? D2: The paper defines density as |E|/|V|. Shouldn't it be |E|/(|V|^2)? D3: The novelty in the paper seems to be applying graph simulation on balls (sub-graphs) rather than the whole graph. D4: A substantial portion of the paper is dedicated to dealing with pattern updates. It seems that pattern updates will not be as frequent as data updates, and one can afford to recompute the team from scratch if a pattern is updated. D5: Section 4.4 talks about introducing an algorithm "dynamic", but this algorithm is not presented in the paper. D6: While the paper talks about team formation in social networks, the experiments do not use social network graphs, but rather citation and recommendation networks.
Required changes for the revision:	

Second round (if applicable)

Review (2nd round):	

Review 1

Overall evaluation:	**-2**
Reviewer's confidence:	**3**

Other Scores

Originality:	**2**
Correctness:	**2**
Completeness:	**3**
Best paper award:	-

Review

Strong Points:	s1. The problem of team formation is interesting and is a popular topic in recent years. s2. The presentation is logical and clear. s3. It uses real datasets in experiments.
Weak Points:	w1. The notion of "team simulation" is incremental, comparing with the existing "strong simulation", and may fail to capture some intuitively simple topologies. w2. The technical depth is of limited contributions. w3. The experiments need improvements.
Review:	This paper studies the problem of top-k team formation: Given a task and an undirected labeled graph G, in which each vertex represents an expert associated with a set of skills, and each edge represents the collaboration relationship between two experts, the problem of top-k team formation is to find a list of k highly collaborative teams of experts in G such that each team satisfies the skill requirements of the task. The problem has been studied in several recent papers. This paper proposes a notion called "team simulation", based on "strong simulation", for top-k team formation. Specifically, given a pattern graph P, it finds subgraphs in G that match P via team simulation as teams. Different from strong simulation, team simulation also considers capacity constraints on vertices. This paper proposes algorithms for both static and dynamic graphs. There are several concerns about the paper. Specifically,

1. The notion of "team simulation" is not novel. It can be simply considered as the existing "strong simulation" (the ball radius can be different from the diameter of pattern graph) with an additional capacity constraint filtering. In other words, team simulation consists of two steps: (1) use strong simulation (for undirected graph) to find a match; (2) use the capacity constraints on vertices to decide whether to filter out the match. The low coupling between two steps also makes team simulation unable to capture some simple topology. Consider the following example, in which "----" denotes an edge between vertices, and the pattern graph P consists of two vertices "v1" and "v2" with labels "A" and "B" respectively, and the data graph G has 3 vertices, namely "u1", "u2" and "u3" with labels "A", "B" and "A" respectively. In addition, both "v1" and "v2" are associated with the capacity constraint [1, 1]; that is, "v1" and "v2" each exactly matches with one vertex in G.

Pattern graph P: v1(A, [1,1]) ---- v2(B, [1, 1])

Data graph G: u1(A) ---- u2(B) ---- u3(A) and an additional edge: u1 ---- u3

Ideally, a match would be either "v1" --> "u1", "v2" --> "u2" or "v1" --> "u3", "v2" --> "u2". However, even for this simple example, team simulation fails to find any match.

2. The technical depth is of limited contributions. The algorithms proposed in this paper are simple and some of them even can be come up with from existing works without much efforts. For example, the algorithms "teamSim" and "minP" turn out to be simple adaptions from the "DualSim" and "minQ" algorithms in [27]. For the incremental algorithms, the idea is straightforward, predictable, and thus is not impressive.

3. Regarding "pattern fragmentation": (1) The proof of Proposition 5 is not correct. Note that the problem of minimum cut can be solved in polynomial time either by a flow algorithm based on the max-flow-min-cut theorem or by a non-flow-based algorithm such as Stoer–Wagner algorithm. Therefore, reduction from minimum cut problem does not imply the NP-completeness of pattern fragmentation. (2) In general, it is not possible to make all fragments roughly of the same size and minimize |C| at the same time. It is better to show your formal objective and justification. (3) It is better if an experiment showing the tradeoff for different "h" is present. Though higher "h" leads to more powerful pruning in identifying affected balls, it incurs more cost in combining matches. It is not clear for now why the choice "h = 3" is reasonable.

4. For the experiments, (1) The competing methods like "minDia", "denAlk" and "minSum" do not require the query to be a pattern graph. It is not clear how the queries are generated in the experiments. (2) In addition to comparing with the original "minDia", "denAlk" and "minSum" algorithms, it is better to add another experiment as follows: as in team simulation, search for a match inside each ball and return the top-k ones. For "denAlk", the optimization of density based filtering should also be used. Note that these variants are immediate and only the "batch" algorithm proposed can also defeat them, both exp-1 and exp-2 can be considered complete. It is also better to use measures such as diameter, sum of all-pair-distances, and MST, which were used in past works, as quality measures in exp-2. (3) It is not clear how sensitive are the proposed algorithms to the parameter "r".

Required changes for the revision:

Second round (if applicable)

Review (2nd round):

Review 2

Overall evaluation: **0**

Reviewer's confidence: **3**

Other Scores

Originality: **4**

Correctness: **3**

Completeness: **3**

Best paper award: -

Review

Strong Points:

The paper brings a principled solution to the important problem of team formation.

The technical contribution is original and deep, covering both formal and systems aspects.

The paper is very well written and pitched to the typical SIGMOD audience.

Weak Points: This problem is well-researched and various restrictions thereof have been solved in the past. (Nevertheless, the submitted solution is broader, and it is based on an interesting reduction to simulation-based matching, which seems like the right trade-off between expressivity and performance.)

Review: The problem is very well researched in the KDD community. It is therefore all the more impressive that the authors find a fresh angle of attack, which allows them to simultaneously cover its top-k variation as well as the variations of incremental updates to the data and the pattern itself. No existing solution covers all of these variations (and the pattern update flavor is completely new).

The idea is to treat team formation as an application of graph pattern matching, then draw from the extensive relevant experience of the DB community. The key insight is to select simulation-based graph matching, which

	strikes a nice balance between expressivity and performance. This has enabled the authors to make progress in a crowded field. The proposed algorithms are interesting and non-trivial. They are analyzed both theoretically and in carefully designed experiments. Due to the quality of contents and presentation, as well as due to establishing an interesting bridge to a KDD topic, this paper would generate interesting discussions between DB and KDD researchers.
Required changes for the revision:	
Second round (if applicable)	
Review (2nd round):	