TeamGen: An Interactive Team Formation System Based on Professional Social Network*

Cheng Ding
School of Data Science and Engineering, East China Normal University
cding@student.ecnu.edu.cn

Fan Xia
School of Data Science and Engineering, East China Normal University
fxia@sei.ecnu.edu.cn

Gopakumar Gopalakrishnan
Infosys Limited.
Gopakumar@infosys.com

Weining Qian
School of Data Science and Engineering, East China Normal University
wnqian@sei.ecnu.edu.cn

Aoying Zhou
School of Data Science and Engineering, East China Normal University
ayzhou@sei.ecnu.edu.cn

ABSTRACT

We introduce TeamGen, an interactive team formation system, to form project teams interactively by leveraging professional social network information of potential members. Unlike earlier approaches that focused on creating flat teams, i.e., teams without communities and central authorities, we model teams as hierarchical structures to reflect the ubiquitous nature of teams in real commercial and open source projects. Correspondingly, our team formation algorithms emphasize local density of sub teams to assess communication costs of newly formed teams. During the demonstration, audience can (a) explore professional social network of potential members, (b) learn the effectiveness and efficiency of our proposed team formation algorithms by comparing them with existing ones, (c) inspect and understand the process of team formation, and (d) interactively refine a project team by revoking computed position assignments.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Social Network; Team Formation

1. INTRODUCTION

The problem of staffing teams with right individuals has received substantial scholarly attention. With the availability of information regarding collaboration history or professional social network of individuals, an effective team can be composed by optimizing the communication cost among team members. An on-line professional social network consists of people with different work related attributes. The on-line interactions and email communications as well as off-line activities such as project co-assignment in enterprise settings reflect connections between people. Previous studies on team formation from social network perspectives have mainly focused on the formation of flat teams. For instance, Lappas, Liu, and Terzi [7] describe an approach that uses social relationships between individuals and defines the total communication cost as the optimization term of the objective function. In another relevant research, Cao et al. [3] address the Jury Selection Problem using micro-blog services (e.g., Twitter) to solve decision-making tasks. The authors describe two models for selecting jury members from a Twitter graph of 690K nodes that minimize overall decision making error-rate. Other studies [2, 5, 1, 9, 8, 6] focus on different variations of this problem or different algorithms for better performance or efficiency. Rangapuram et al. [9] proposed more realistic team formation approaches on dense sub-graphs.

However, most of such studies are focused on the problem of solving at flat teams with simple constraints on positions, i.e., teams without sub-groups and central authorities. At the same time, teams with tree structure, such as a project team in a large company shown in Fig. 1, is becoming quite ubiquitous in real world. To develop a realistic interactive team formation system, we need to solve the following two main challenges. First, the professional social network need to be created and updated by continuously integrating various operational data. The efficiency of accessing the network is also critical to the performance of whole system. Second, we model both project teams and team specifications by the tree structure. However, forming such hierarchical teams is a NP-Hard problem.

In this demo, we present TeamGen, a system to form hierarchically structured project teams interactively by leveraging professional social network of potential members in a large multinational enterprise with over 145,000 employees working in more than 50 countries across the world. In the next section we first introduce the hierarchical team formation problem. Section 3 describes implementation of the system, including the system architecture, two heuristic

*Gopakumar Gopalakrishnan and Weining Qian are co-correspondance authors.

WWW 2017 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054725

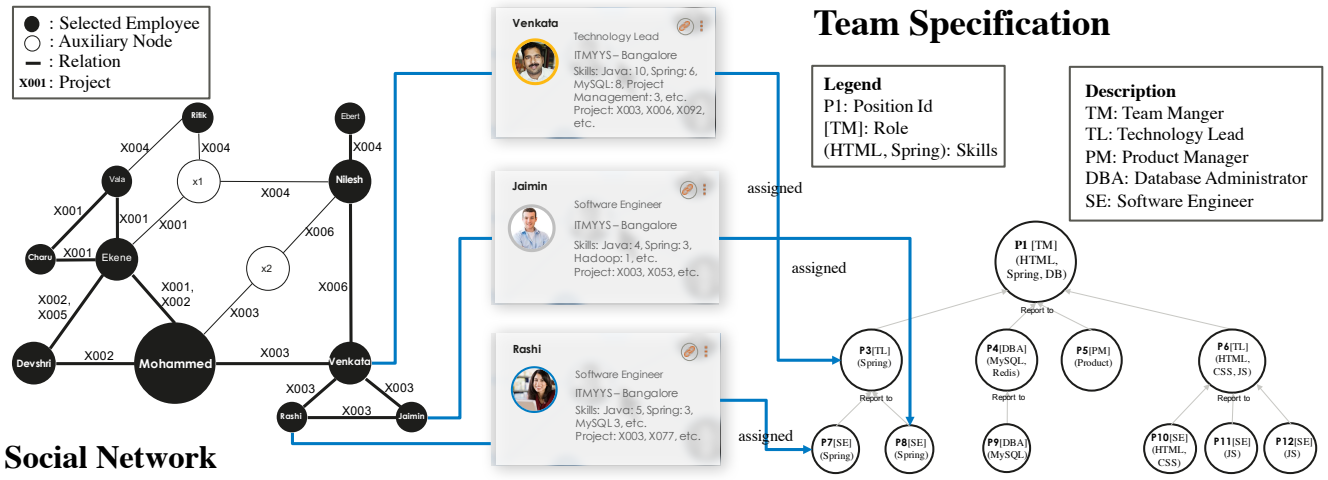


Figure 1: Team formation of a real project team

team formation algorithms and the sub-team index[4]. The demonstration outline is presented in the last section.

2. HIERARCHICAL TEAM FORMATION

Professional Social Network. *Professional Social Network* is modeled as an undirected weighted graph, where nodes correspond to employees and edges reflect social connections between employees. Social connections are derived from previous interactions including common project assignments, common meetings attended etc. The *social distance* equals $1 - \frac{|H_i \bigcap H_j|}{|H_i|+|H_j|}$, where H_i represents the number of activities e_i attended and $|H_i \bigcap H_j|$ is the total number of activities e_i and e_j attended together. The smaller the *social distance* is, the closer the two employees are. Thus, the *social distance* between two nodes is defined as distance of the shortest path connecting two nodes. Besides, we only consider paths within 3 hops in the social network, otherwise a given maximum distance is assigned.

Team Specification. In previous studies, *Team Specification* is defined as a set of skill-constrained positions in a typically flat structure. However, teams that operate in real organizations are more complex. First, positions in teams have more constraints, including skills and proficiency levels, role, deadline, location, work load and etc. Second, teams in real organizations are hierarchical with "report to" relationships between positions.

Therefore, we define *Team Specification* of projects, denoted as $TeamSpec_i$, as a tree with node set V and edge set E, wherein i represents project number, V is a subset of whole constrained positions and each edge $e_{n,m}$ in E indicates position p_m reports to p_n. Further, the sub team specification led by p_n in project i, denoted by $TeamSpec_i(p_n)$, is defined in Equation 1. It consists of p_n and people reporting to p_n directly.

$$TeamSpec_i(p_n) = \{p_m | p_m = p_n \vee e_{n,m} \in E\} \quad (1)$$

Team Formation. Members working together in a team have a communication cost, induced by different factors such as, social distance, variations in skills, expertise, education background etc. Thus, we define the communication cost between two employees as a weighted sum of different features in Equation 2, denoted by $CommCost(e_n, e_m)$.

$$CommCost(e_n, e_m) = \sum_{i \in |factors|} w_i * f_i(e_n, e_m) \quad (2)$$

The objective of *Team Formation* is to identify potential team members in such a way that chosen members match given specifications for different positions and have minimal communication cost among them so that they can work together well as a team. Traditional team formation approaches optimize the *global density*, which is the summation of communication cost of all employee pairs. As shown in Fig. 1, we propose that the person assigned to $P7$ should be familiar with others in the sub-team led by person of $P3$. However, she may not know people in sub-team led by person assigned to $P6$. Accordingly, we define *local density* as the summation of communication cost of pairs in all sub-teams. Existing algorithms such as Rare First [7], Best Sum Distance and Best Leader Distance [5] emphasize the *global density* while our proposed algorithms *TopDown*, *BottomUp*, and *IndexBased* focus on *local density*. The effectiveness and efficiency of all these algorithms will be compared in this demo.

3. SYSTEM IMPLEMENTATION

3.1 System Overview

The architecture of our system is presented as Figure 2. The main layers of *TeamGen* include *Data Preparation*, *Data Service*, and *Team Formation*. Professional data is collected and pre-processed in the *Data Preparation* layer and migrated to the *Data Service* layer. Intensive analyses are applied on data dumped from *Data Service*. *Team Formation* invokes query interfaces of *Data Service* to access data required by the formation process. In the following sections, we discuss implementation details of each layer.

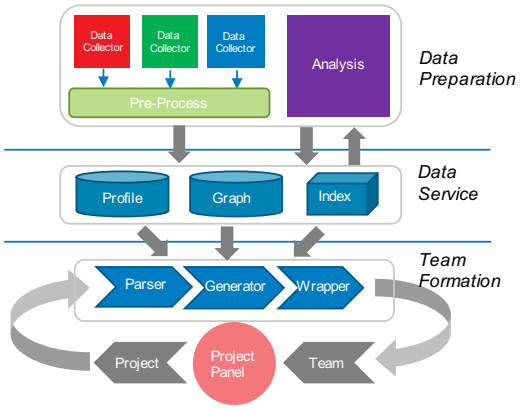


Figure 2: Architecture of TeamGen

3.2 Data Preparation Layer

Data from different sources is collected and integrated by customized data collectors. We identify two types of data, profile and relation. Profile describes basic attributes about an employee, such as name, role, skills, location, etc. Relation describes social connections between employees, such as common project assignments etc.

3.3 Data Service Layer

The *Data Service* layer loads cleansed data from data preparation layer into databases and creates indexes for fast accessing. We use relational database to store employee profiles and use graph database to store the professional social network. Inverted index are built so that we can search for members with certain skills using skills as keyword queries.

The *Professional Social Network* in this work is derived from project co-assignment data. When co-assignment indicates an employee worked in a project for t days, we update the project time property of the employee in the graph by t. Then, we create an edge between she and others in sub-teams containing her if those edges don't exist. Each edge is also annotated with project time property and is incremented correspondingly. To provide fast query capability for querying communication cost within 3 hops, we derive the original graph by calculating the distance of each edge using the defined equation $1 - \frac{|H_i \bigcap H_j|}{|H_i|+|H_j|}$. A 3-hop breadth first search is used to connect each node to more reachable nodes. Distance to a reachable node is sum of edges' distance on the path.

3.4 Team Formation Layer

The *Project Panel* is a web-based interface that can build team specifications. *Generator* accepts team specifications and other configurations. Then, it uses configured algorithms and optimizers to generate teams. Finally *Generator* pass the formed team back to project builder for visualization. In this section, we discuss the implementation of *Generator*, including optimizer, algorithm, index and evaluations.

Optimizer. Algorithms are configured with different optimizers as search strategies to form a sub-team. For example, *FFOptimizer* is used in Top Down (TDTF) algorithm. Given a sub-team leader, the optimizer iterates through each follower position in her sub-team and select a competent candidate who is closest to the leader. *FFOptimizer* optimizes the average communication cost between the leader and its followers, without considering communication cost between followers. Another optimizer, *FSOptimizer* aims at optimizing a sub-team where followers of non-leaf positions are already selected. The procedure behaves similar to algorithms for finding flat teams with an exception that people assigned to follower of each position affect the choice of current position.

Algorithms. In addition to several existing algorithms, i.e., Rare First [7], Best Sum Distance and Best Leader Distance [5], we also implement additional algorithms such as Top Down (*TDTF*), Bottom Up (*BPTF*) and index-based algorithms to strengthen the team formation results for hierarchical team specifications. Details of the proposed algorithms are introduced in [4].

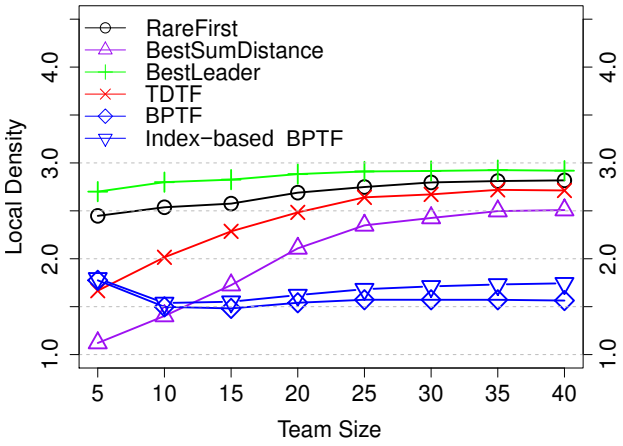
To construct hierarchical teams, team formation algorithms take team specifications and professional social network as inputs, search employees in a top-down or bottom up manner, and generate a set of <position, person> pairs as team assignments. Algorithms *TDTF* and *BPTF* are based on the two aforementioned heuristics. The *TDTF* starts by constructing the root node plus its sub-team using *BestSum* and then invokes *FFOptimzer* to form a sub-team for each newly qualified non-leaf position. In a different way, *BPTF* constructs the team from the lowest level of the team specification tree to the root position using *FSOptimizer*.

Index. A large fraction of time is required for constructing sub-teams according to the performance analyses, which is omitted for limitation of space. We also observe many sub-team specifications appear frequently based on the statistics of real-life projects data provided by a large enterprise. To avoid expensive sub-team construction and accelerate the performance, we use inverted index to implement the sub-team index. For each sub-team specification appearing in past projects, *BPTF* is used to construct the best team based on currently available employees. We use skill pairs as terms of the inverted index and insert the generated team into corresponding posting lists. Each posting list is sorted in ascending order of sub-teams' local density.

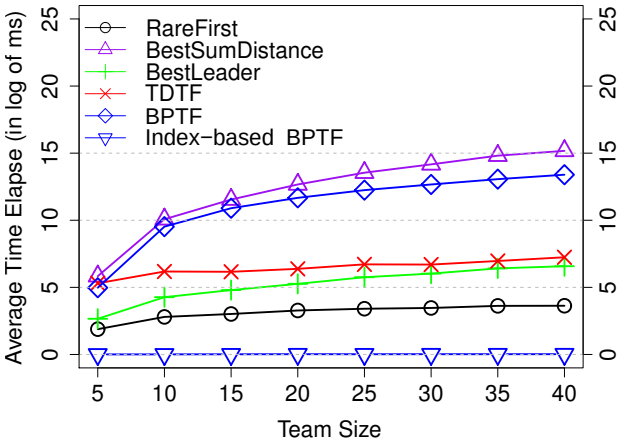
The generated sub-team index modifies the way in which optimizers search sub-teams. For a sub-team specification, optimizers enumerate skill pairs and use them to query the sub-team index to find out sub-teams which contain all skill pairs. Pre-computed sub-teams are returned in ascending order of local density. A validation is applied to each returned sub-team to check whether it fulfills the specification, i.e., it has competent team members assigned to all positions. Currently, we stop the verification process until k qualified sub-teams are obtained from the index. Finally, the one which leads to the minimal local density after merged to team assignment is chosen from the k results.

Other Strategies. 1) Since team formation is not a one-time activity, our system support continuous updating of teams by locking the determined positions. 2) The procedure of team formation can be divided into a sequence of steps, which enables users to inspect and interact with the team formation process.

Experimental Results. All the team formation algorithms are implemented using Java. To compare the performance and efficiency of different algorithms, we have implemented several evaluators, including global density, local density, runtime and so on. We use the project assignment



(a) team size vs. local density



(b) team size vs. runtime (log scaled)

Figure 3: The performance and effectiveness of approaches under different team sizes

data of a large enterprise that contains over 20,000 real-life projects for simulation experiments. Due to page limitation, only results of local density and runtime are shown in Fig. 3(a) and 3(b) [4]. The results suggest that BPTF and indexed based algorithm work best when team size is large. Besides, local density is not sensitive to the team size while the performance of other approaches degrades for larger team. Although index-based algorithm causes minor decrease in local density, it executes much more efficiently as shown in Fig. 3(b).

4. DEMONSTRATION OUTLINE

We deploy the demonstration system in a single server. A large project staffing data in a multinational organization that includes over 146,000 employees and over 1,200,000 relations are loaded into the system. Then, the following four main aspects of *TeamGen* are demonstrated.

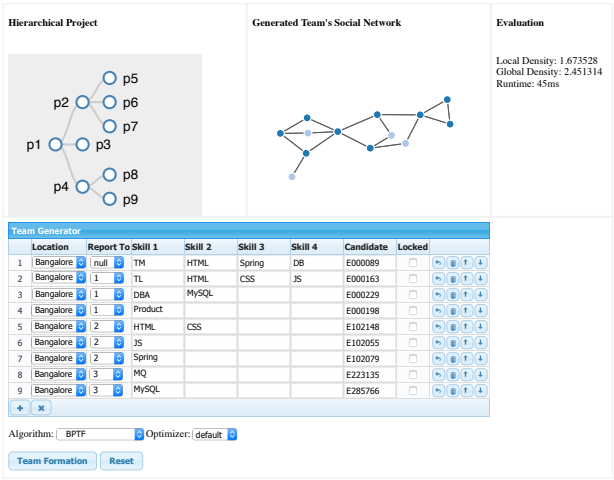


Figure 4: The user interface of *TeamGen*

First, the professional network among members in an enterprise, shown in Fig. 5, can be explored to understand how edges are created between people. Descriptive statistics of the graph, such as information about node degree, shortest path, and node attributes can be viewed. Besides, skills of people are provided in their profile and the distribution of skills can also be found.

Second, several **built-in team specification** examples derived from past large complex projects are provided by the system. Audience can select those specifications and view their hierarchical structures and position requirements. Upon submitting team formation request, the system will form the team using above mentioned algorithms. The interface can show a live comparison of the exisitng approaches, including:

- Two effectiveness measurements, i.e., local density and global density, are computed to reflect the communication cost of the generated team.
- The sub-graph of the professional social network consisting of people in the generated team is also presented to provide a vivid picture of the interactions across sub teams.
- Finally, the time elapse of the algorithm execution is given to measure the efficiency of different algorithms.

Third, the **usage of sub-team index** in the formation of project teams is explained earlier. The system will show the steps to form the project team and queries sent to the sub-team index for each step. The returned sub-teams results together with the query latency are displayed and how a sub-team is selected from results is explained.

Finally, a web-based **interactive project builder** is provided so that audience can create team specifications by themselves. Taking the team specification as input, *TeamGen* can generate the whole team at once or step by step with users' interactions. For example, users can interactively refine the newly formed team by revoking assignments of certain positions. The system will incrementally compute new members for those positions using implemented algorithms.

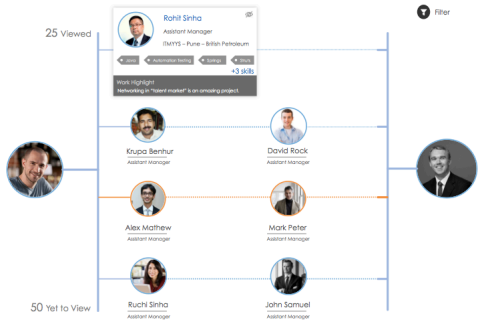


Figure 5: Explore the professional social network

Relationships between selected employees can be browsed according to their professional social network. The aforementioned aspects of different algorithms are also displayed. Through this part of the demonstration, audience can learn how our system would be used in real world settings.

5. ACKNOWLEDGMENTS

This work is partially supported by National Hightech R&D Program (863 Program) under grant number 2015AA015307, and National Science Foundation of China under grant numbers 61432006 and 61672232.

6. REFERENCES

[1] A. An, M. Kargar, and M. ZiHayat. Finding affordable and collaborative teams from a network of experts. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA.*, pages 587–595, 2013.

[2] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Power in unity: forming teams in large-scale community systems. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 599–608, 2010.

[3] C. C. Cao, J. She, Y. Tong, and L. Chen. Whom to ask? jury selection for decision making tasks on micro-blog services. *CoRR*, abs/1208.0273, 2012.

[4] C. Ding, F. Xia, G. Gopalakrishnan, W. Qian, and A. Zhou. Online formation of large tree-structured team. In *Database Systems for Advanced Applications - DASFAA 2017 International Workshops: BDMS, BDQM, MoI, and SeCoP, Suzhou, Jiangsu, China, March 27-30, 2017, Proceedings*, 2017.

[5] M. Kargar and A. An. Discovering top-k teams of experts with/without a leader in social networks. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 985–994, 2011.

[6] M. Kargar, A. An, and M. ZiHayat. Efficient bi-objective team formation in social networks. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II*, pages 483–498, 2012.

[7] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 467–476, 2009.

[8] A. Majumder, S. Datta, and K. V. M. Naidu. Capacitated team formation problem on social networks. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1005–1013, 2012.

[9] S. S. Rangapuram, T. Bühler, and M. Hein. Towards realistic team formation in social networks based on densest subgraphs. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1077–1088, 2013.