Forming Diverse Teams Based on Members' Social Networks: A Genetic Algorithm Approach

Archan Das[1], Diego Gómez-Zará[2](✉), and Noshir Contractor[2]

[1] Illinois Mathematics and Science Academy, Aurora, IL 60506, USA
adas@imsa.edu
[2] Northwestern University, Evanston, IL 60208, USA
dgomezara@u.northwestern.edu, nosh@northwestern.edu

Abstract. Previous research shows that diverse teams in background and skills can outperform homogeneous teams. However, people often prefer to work with others who are similar and familiar and fail to assemble teams with high diversity levels. We propose a team formation algorithm that suggests diverse teams based on individuals' social networks, allowing them to keep high familiarity levels. Our novel algorithm is based on the NSGA-II genetic optimization that splits students into well-connected and diverse teams within an organizational network. It optimizes measures of team communication cost and diversity in $O(n^2)$ time. The optimization finds Pareto optimal solutions that optimize both metrics, returning teams that have both diversity in member attributes and previous connections between members. We tested the algorithm on real team formation data collected from the *MyDreamTeam* platform. The solutions provided by the algorithm are superior to the teams assembled by the students, in both diversity and communication cost measures.

Keywords: Team formation · Social networks · Diversity · Genetic algorithms

1 Introduction

In traditional classes, teachers can either assign students manually to different teams or let them self-assemble their teams. Choosing the method is not an easy decision since both have limitations. On the one hand, assigning students manually to teams increases the likelihood of heterogeneous teams, having students from different gender, status, demographics, and competence. Forming diverse teams can help students learn from other classmates with different strengths and backgrounds, which can lead to collaborative solutions [9]. However, students can be against working with unfamiliar classmates. When an instructor assembles a team, students are likely to feel forced to work with other classmates, a situation that decreases the likelihood of working effectively together

R. M. Benito et al. (Eds.): COMPLEX NETWORKS 2020, SCI 944, pp. 346–357, 2021.
https://doi.org/10.1007/978-3-030-65351-4_28

[5]. Moreover, constraints such as testing several combinations from a large class or checking students' availability will make the team formation task even harder. On the other hand, when students assemble their own teams, they can show more commitment and positive group attitudes [2] since they are more likely to choose partners based on friendship and similarity [7]. Although, students are less likely to give relevance to choosing teammates who possess different backgrounds or sets of skills. As a result, students will tend to form homogeneous teams because they prefer to choose teammates who are similar and familiar [10], and the likelihood of forming creative and high-performing teams decreases [18].

In this paper, we introduce a novel team formation algorithm that combines both strategies to assemble diverse teams based on students' current social networks. Based on a NSGA-II genetic optimization, this algorithm considers metrics for both diversity and previous relationships between team members and determines optimized team assignments for all students in the networks. The contributions of this paper are (i) the definition of the diverse team formation problem based on members' social networks, (ii) a genetic approach to solving that problem, and (iii) an experimental evaluation of our optimization compared to teams randomly assembled and assembled by students on the same network.

The paper is structured as follows. Section 2 describes previous studies in team formation algorithms. We define the team formation problem in Sect. 3 and our algorithm in Sect. 4. We present an empirical evaluation of this algorithm in Sect. 5. We end this article by discussing the implications of this algorithm and future work.

2 Related Work

Computer science research has devoted considerable attention to developing team-formation algorithms and frameworks. Forming efficient teams is a complex task since finding the most diverse team combination requires assessing all the possible solutions. In a group with n members that are splitting up into teams of m members, the total number of possible ways to form teams is $n!/((n-m)!m!)$, where $n > m$. For example, forming teams of size 5 in a 50-student class, the instructor must evaluate 2,118,760 combinations. Considering all team combinations that systems can provide—of different sizes and memberships—the primary goal of team formation algorithms is to find an efficient method to assemble teams. This problem has been classified as an NP-hard problem since finding the best team combination requires computing all the possible team combinations (i.e., brute-force search), which cannot be done in polynomial time [6]. Contributions in this field are based on what variables and mechanisms are considered to find optimal solutions that approximate the best solutions, using less computer memory and less time.

Through several methods and algorithms, these systems' objective function is to maximize a specific team's characteristics (e.g., social connections, skills covered by the team) subject to communication or personnel costs. Some algorithms consider the sum of individuals' skills as part of the optimization problem (e.g., forming a team of experts from a research community), and others assign members according to their specific roles in the team [19]. Other systems consider members' social networks to assemble their groups. One example is Lappas et al. [14], which considers the team-formation problem using members' skills and social networks. The systems' goals are not only to assemble groups that meet the tasks' skill requirements but also to assemble teams that can work effectively together. Some crowdsourcing systems fit into this category since one of their challenges is to divide projects' tasks and assign them to crowd-workers according to their skills and availability [20]. Optimizations including other factors in addition to communication cost have found success in the use of discrete particle swarm optimizations [22] and genetic optimizations.

Multiple studies consider the role of social networks, previous collaborations, and the intensity of interactions among a pool of individuals to assemble teams [1,12,17]. Other developed systems consider how members can complement their personalities and skills to create balanced teams [16]. Latorre and Suárez [15] develop a framework that facilitates team assembly in a systematic and reproducible way. This framework uses workers' social networks, prior experience, and previous collaborations to build compatibility networks among participants, where each connection represents whether the workers have compatible (or incompatible) social skills. In contrast to these previous studies, we propose an algorithm that optimizes for diversity in team formation within social networks.

3 The Diverse Team Formation Problem

Given a social network N of students U with individual characteristics, the team formation problem consists of assigning students to teams that maximize within-team connection and diversity.

3.1 Inputs

The algorithm accepts a list of n students $U = \{u_1, u_2..., u_n\}$. Every student has values for l different categorical attributes $G = \{g_1, g_2..., g_l\}$ and m different continuous attributes $C = \{c_1, c_2..., c_m\}$. These attributes may have different constraints. Each attribute represents some information about the student, such as age, gender, or race. The second parameter is an undirected and unweighted network N that represents social connections among the n students. In other words, if students i and j are friends ($i, j \in U$), then $N_{i,j} = 1$. Otherwise, $N_{i,j} = 0$. The network N must be fully-connected (i.e., for any two students in the network there must be some path of connections between them). Finally, the algorithm requires a team size k, which specifies the number of members per team. The students must be able to divide evenly into teams with no extras (i.e., $n \bmod k = 0$).

The genetic algorithm accepts two parameters, the population size P and the maximum number of generations. The population size P is the total number of chromosomes in the population of the algorithm, while the maximum number of generations is the generation at which the optimization stops.

3.2 Output

The algorithm returns a list of team solutions T of size P. A solution T_p, $p \in P$, represents a specific combination of q teams based on the n students, $T_p = \{t_1, ..., t_q\}, p \in P$. In each solution, (1) each team $t \in T_p$ is formed by k students, where $k < n$; (2) each student $u \in U$ is in exactly one team t from T_p; and (3) all teams of T_p have size k. The solutions are optimized for both diversity and communication cost as outlined below. The solution set is comprised of successive fronts of Pareto-optimal solutions. A solution is Pareto optimal if there is no other solution that is superior in both metrics.

3.3 Metrics

Our algorithm is a bi-objective optimization on two metrics: communication cost and diversity. The goal of the algorithm is to find a set of Pareto optimal solutions to the problem, where no solution in the set is inferior to another solution in both metrics.

Communication Cost. Kargar et al. [13] found the total sum of distances between team members to be a reasonable measure of communication cost, as it is more stable to changes in the network than other potential measures. We define the distance between students as the shortest path length while traversing the edges of the network from one student to the other.

The goal of optimizing communication cost is to form teams with certain levels of familiarity. Teams with lower communication costs exchange information and ideas more efficiently than teams with higher communication cost, enabling more effective teamwork [11]. We define the sum of distances (denoted Cc) of a team t, $t \in T$, with k students as:

$$Cc_t = \sum_{i,j \in t, i \neq j}^{k} D(u_i, u_j)$$

where $D(u_i, u_j)$ returns the length of the shortest path between nodes u_i and u_j, and $i, j \in t, i \neq j$. Our algorithm minimizes the average sum of distances across all formed teams in the student network. Computing the sum of distances of a set of teams runs in $O(n^2)$ time. Minimizing the sum of distances is an NP-hard problem [14]. We then use a genetic algorithm to approximate the optimal solution.

Diversity Metric. In addition to Cc, we include as a goal generating diverse teams with a broad array of backgrounds and skill sets. We define a diversity metric based on two measures for team diversity [8]. The first one is coefficient of variation (CV), which is defined as the ratio of the standard deviation to the mean of a variable x, and its formula is $\sqrt{\sum(x_i - x_{mean})^2/n}/x_{mean}$. A low score means all members have similar levels of the attribute, and a high score means that members have different levels of the attribute. The coefficient of variation measures continuous team member attributes. The second team diversity measure is the Blau index B, which is defined as $1 - \sum p_i^2$ and p is the proportion of team members who fall into a particular category i. A low score means that all members fall into the same category, and a high score means that members fall into different categories. This metric works with categorical team member attributes. These two team diversity measures are useful because they do not change when the input data is scaled up or down linearly, and they both tend to stay around the same values. This allows us to aggregate diversity for different attributes into a single value.

The overall diversity metric V of a team t is defined as the sum of B for all categorical variables and CV for all continuous variables. In other words,

$$V_t = \sum_i^l \sum_j^k (\sqrt{\sum(x_{i,j} - x_{i,mean})^2/k}/x_{i,mean}) + \sum_i^m \sum_j^o (1 - \sum p_{i,j,t}^2)$$

where l is the number of continuous team member attributes, m is the number of categorical team member attributes, k is the number of members of the team t, $x_{i,j}$ is the value of the continuous attribute i for the student j, $x_{i,mean}$ is the average of the continuous attribute i for the team t, o is the number of values for the categorical attribute i, and $p_{i,j,t}$ is the proportion of students with the categorical attribute i with the value j in the team t. The algorithm maximizes the average V across all formed teams in the network.

4 Algorithm

In order to solve the diverse team formation problem, we propose a multi-objective genetic optimization. We chose to implement a genetic optimization because of their efficiency: it can run in $O(n^2)$ and generate a set of optimized solutions to multi-objective problems such as diverse team formation.

4.1 Genetic Optimization

We use the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) algorithm formulated by [4]. Genetic algorithms optimize a set of possible solutions called chromosomes into an optimized solution through natural selection, imitating evolution in nature. NSGA-II is a multi-objective genetic optimization that employs elitism (i.e., selecting the best chromosomes of the current generation to the next generation) and runs in $O(n^2)$ time. Elitism ensures that Pareto-optimal solutions are kept in the population, even if they are found early on in the optimization. This improves the ability of the optimization to converge on the correct

Pareto front in a reasonable number of generations. The non-dominated sorting ensures that one objective will not dominate the other. These properties improve the ability of the algorithm to provide a high performing Pareto front in a small number of generations. We adapted the NSGA-II algorithm to our specific team formation problem. The steps of the optimization are outlined below.

Chromosomes. In our implementation, each chromosome represents a potential set of teams T in the network N. The population of solutions is initialized with each chromosome having students assigned to random teams. The total number of chromosomes in the population is an input parameter. Chromosomes are stored as 2-dimensional arrays of shape (q, k), where q is the number of teams and k is the number of students per team. Each chromosome is a potential solution to the problem, and the goal is to output a high-performing set of chromosomes.

```
Input  : Two parent chromosomes $p_1$ and $p_2$, number of teams $q$, a list
         $U$ including all students
Output: A child chromosome $c$, with $q$ teams
initialization;
$teampool \leftarrow$ Concatenate($p_1$, $p_2$);
$c \leftarrow$ SelectNRandomElements($teampool$, $q$);
$repeat\_locations \leftarrow []$;
$missing\_members \leftarrow U$;
for $t$ in $c$ do
    for $m$ in $t$ do
        if $m$ not in $missing\_members$ then
            Append($repeat\_locations$, [GetIndex($c$, $m$),
             GetIndex(GetIndex($c$, $m$), $m$)]);
        end
        else
            Remove($missing\_members$, $m$);
        end
    end
end
for $l$ in $repeat\_locations$ do
    $m \leftarrow$ SelectNRandomElements($missing\_members$, 1);
    $c[l[0]][l[1]] \leftarrow$ m;
    Remove($missing\_members$, $m$);
end
return $c$;
```

Algorithm 1: Crossover of two chromosomes. Each chromosome is a potential solution to the team formation problem.

Crossover. The first step of each generation is to take the existing chromosomes and generate offspring from random pairings of parent chromosomes until the population size is doubled. NSGA-II is designed to work on any data structures that can be reasonably combined to produce offspring in the crossover step. Our method of crossover is outlined in Algorithm 1. This method involves random sampling of members in forming teams. The random sampling provides sufficient mutation for the algorithm to introduce chromosome diversity into the population without adding another mutation step. One pitfall with this approach is that it is difficult to control the level of mutation in the crossover.

Non-dominated Sort. After the crossover step, the overall diversity V and the communication cost Cc of each chromosome is calculated and stored. This includes both parent chromosomes and child chromosomes. A front is created consisting of every chromosome that is Pareto optimal. For any chromosome in the first front, there exists no other chromosome in the population with more diverse and more connected teams.

After the first front is found, those solutions are kept, and a second front is found, which consists of Pareto optimal solutions disregarding solutions in the first front. The sort keeps finding successive fronts until the total number of chromosomes surpasses the predetermined population size.

Crowding Distance. After multiple fronts are found, there may be more solutions in all of them combined than the population size of the algorithm. In order to select the best chromosomes for the algorithm, a crowding distance metric is computed that determines how close chromosomes are to each other in performance. The population is then trimmed down to the correct size by eliminating chromosomes that perform similarly to other chromosomes. This helps keep a broader front of solutions instead of keeping redundant chromosomes. These chromosomes form the parents of the next generation of optimization.

4.2 Optimized Solutions

After the optimization has run through the previously specified number of generations, the algorithm returns an output of p solutions where p is the population size. These solutions form the final Pareto front. A set of Pareto optimal solutions is superior to a single solution for our bi-objective optimization because it allows the teacher to evaluate multiple solutions exploring different priorities for diversity and communication cost.

5 Empirical Evaluation

We evaluated our algorithm on data from real team formation cases and compared our optimized teams to the teams assembled by students. The algorithm was implemented in Python 3.8.5., and the experiments were conducted on an Intel Core i5-8265U computer with 8 GB of RAM.

5.1 Dataset

Our dataset comes from the *MyDreamTeam* recommender [3], which helps students assemble their own teams. On this platform, students create profiles, search for others, and send invitations that can be accepted or rejected until teams are formed. This dataset includes students from university courses in the US. The team task given to the students was a project related to their coursework for the undergraduate courses and a case study analysis and discussion tasks for graduate courses. Each student only assembled into one team and participated in only one course. Relationships between students were determined by their responses to a survey asking them which other classmates of the course they already knew.

5.2 Experimental Setup

We ran the optimization on two courses of this dataset: The first course has 60 students with 771 connections, which were split into 10 teams. The second course has 48 students with 211 connections, which were split into 12 teams. The optimization was run for 50 generations with a population size of 20 chromosomes. The control measures are the average communication cost, and diversity of 20 randomly generated possible solutions, with no optimization and students shuffled into teams.

5.3 Composition Measures

To evaluate the quality of the teams produced by this algorithm, we compared the experimental results to the real teams assembled by the students on *MyDreamTeam*. To evaluate the differences, we used (a) communication cost (Cc) measured by the net sum of distances and (b) overall diversity (V) measured by an aggregate of appropriate diversity indices, as outlined above in Sect. 3.3.

5.4 Results

Figure 1 compares team solutions generated by our optimization (blue circles) to the teams self-assembled by the students in each course (orange triangles). The x-axis represents the total teams' communication cost (Cc) of the solutions. Lower scores on this axis represent solutions with lower communication costs (i.e., teams internally more connected). The y-axis represents the total teams' diversity score (V) of the solutions. Lower scores in this axis represent solutions with less diverse teams. The optimized solution outperforms the self-assembled teams in both diversity and communication cost metrics for both sets of students. Both solutions significantly outperform the randomly generated teams in communication cost. However, the self-assembled teams' diversity scores are inferior compared to the random teams, while the optimized teams are more diverse than both the randomly generated teams and the self-assembled teams. The difference in measures between the three solutions was greater in the smaller, less connected group than in the larger, more connected group.

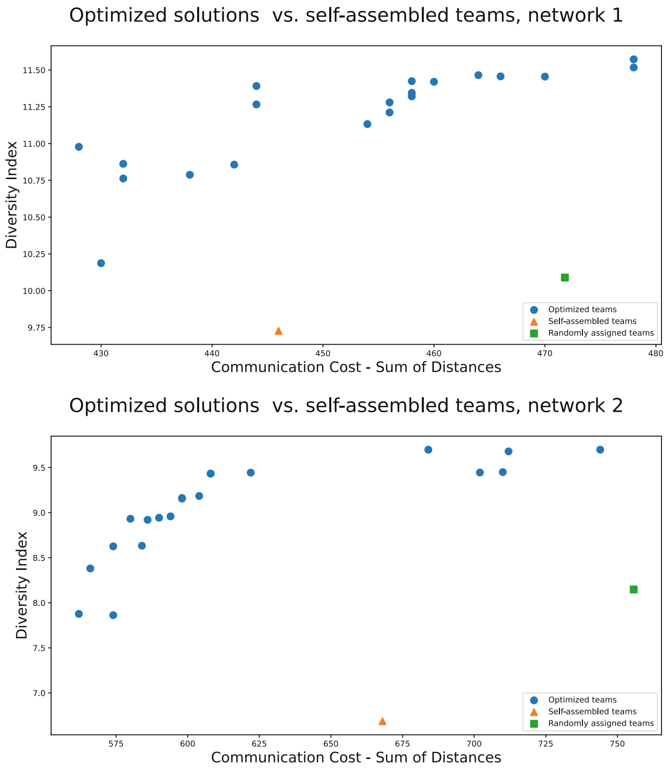

Fig. 1. Comparison of optimized solutions and self-assembled teams on diversity and communication cost metrics. The optimized solutions are blue circles, the self-assembled solution is the orange triangle, and the randomized solution is the green square.

5.5 Performance

As stated before, every step of this algorithm runs in $O(n^2)$ time. We found 20 chromosomes running for 50 generations to be sufficiently powerful for a high-performing Pareto front. On average, the optimizations for the first course ran for 11.676 s and used 60.74 MB of memory. The optimization for the second course ran in 8.628 s and used 59.81 MB of memory.

6 Discussion

Communication Cost. The self-assembled teams had a lower communication cost than the set of randomly generated teams, suggesting that people naturally tend to form teams with people they know. Still, our genetic optimization outperformed self-assembled teams in communication cost. People generally tend to form teams with those they are familiar with, which increments satisfaction and commitment with the team [21]. Our algorithm optimized team communication cost to lower values than the self-assembled teams observed, suggesting that while team members have some intuition on forming well-connected teams, they lack reliable knowledge on higher-order connections between students. [11] found that indirect connections still greatly influence performance within community models. Even if two team members do not directly know each other, their indirect relationship still matters, even though it is difficult for individual students to know about their relationships with students other than those that they are directly connected to. Our algorithm excels in considering the broader social network structure because it is given a complete view of relationships between students that is not available to the students themselves.

Team Diversity. The genetic optimization was significantly superior to self-assembled teams in maximizing diversity. The self-assembled teams were actually less diverse than the set of randomly generated teams, indicating that students preferred to team up in less diverse teams in general. Our team formation algorithm presents a significant opportunity to boost team diversity over self-assembled teams and guarantees high familiarity levels among team members. As prior research suggests, self-assembled teams have a tendency to form into homogeneous groups of members with similar attributes. These homogeneous teams may perform worse than more diverse teams for complex and creative tasks [21]. Team formation algorithms are less subject to these human biases than in team self-assembly, and can thus optimize teams for diversity.

Tradeoffs. The algorithm generates diverse teams with a level of familiarity due to prior connections. While it was definitely possible to form very connected and very diverse teams, we did notice a tradeoff in the absolute extreme of both metrics: the solutions that were the most diverse, or the most connected teams, often sacrificed performance in the other metric. However, solutions are generally found that score nearly optimally in both metrics. This is apparent from the points closest to the upper left corner in Fig. 1.

Limitations. This work presents the following limitations. The measures for diversity and communication cost are scaled specifically to each unique network, and thus cannot be compared across different sets of users. Moreover, the diversity measure is an aggregate of multiple diversity metrics for each attribute sampled; thus, it is difficult to assign any real meaning to the diversity metric apart from relative differences with the same network. For a 60 member network split into teams of 6, the theoretical maximum value for diversity is 34.5: the upper limit changes depending on the size of each team. Future algorithms should consider how different measures of diversity can be analyzed separately and according to the specific pool of students.

7 Conclusion and Future Work

In this paper, we studied the problem of assembling teams from a social network that minimize communication cost and homogeneity. We defined objective functions for the two and proposed a genetic algorithm for finding the most well-connected diverse teams. Our experiments on a real dataset showed that the proposed algorithm produces diverse teams with lower communication cost and higher diversity than those assembled by the students.

As future work, we will compare this algorithm with other algorithms implementations (e.g., Brute-Force, Rarest-First, Steiner) and datasets to evaluate our algorithm's performance. Moreover, we will explore how to incorporate constraints in this team formation problem, such as the task requiring a specific set of skills, and consider different levels of relationships between students and evaluate the algorithm's performance on weighted and directed networks. Finally, we will compare whether teams assembled with this algorithm can outperform teams assembled randomly, by instructors, or by the same students. This testing should be done in different educational contexts. By combining both diversity metrics and students' social networks, we expect that students can work together with familiar individuals who can work better in tasks that require a combination of diverse backgrounds and skills.

References

1. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., Leonardi, S.: Online team formation in social networks. In: Proceedings of the 21st International Conference on World Wide Web, WWW 2012, New York, NY, USA, pp. 839–848. Association for Computing Machinery (2012). https://doi.org/10.1145/2187836.2187950
2. Chapman, K.J., Meuter, M., Toy, D., Wright, L.: Can't we pick our own groups? The influence of group selection method on group dynamics and outcomes. J. Manag. Educ. **30**(4), 557–569 (2006)
3. Contractor, N., DeChurch, L.A., Sawant, A., Li, X.: My dream team assembler (2013)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
5. Dobbin, F., Kalev, A.: Why diversity programs fail. Harvard Bus. Rev. **94**, 14 (2016)
6. Eftekhar, M., Ronaghi, F., Saberi, A.: Team formation dynamics: a study using online learning data. In: Proceedings of the 2015 ACM on Conference on Online Social Networks, COSN 2015, New York, NY, USA, pp. 257–267. Association for Computing Machinery (2015)
7. Gómez-Zará, D., Paras, M., Twyman, M., Lane, J.N., DeChurch, L.A., Contractor, N.S.: Who would you like to work with? In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, New York, NY, USA. Association for Computing Machinery (2019). https://doi.org/10.1145/3290605.3300889

8. Harrison, D.A., Klein, K.J.: What's the difference? diversity constructs as separation, variety, or disparity in organizations. Acad. Manag. Rev. **32**(4), 1199–1228 (2007)
9. Herriot, P., Pemberton, C.: Competitive Advantage Through Diversity: Organizational Learning from Difference. Sage Publications Inc., Thousand Oaks (1995)
10. Hinds, P.J., Carley, K.M., Krackhardt, D., Wholey, D.: Choosing work group members: balancing similarity, competence, and familiarity. Organ. Behav. Hum. Decis. Process. **81**(2), 226–251 (2000)
11. Huberman, B.A., Hogg, T.: Communities of practice: performance and evolution. Comput. Math. Organ. Theory **1**(1), 73–92 (1995)
12. Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, New York, NY, USA, pp. 985–994. Association for Computing Machinery (2011). https://doi.org/10.1145/2063576.2063718
13. Kargar, M., An, A.: Discovering top-k teams of experts with/without a leader in social networks, CIKM 2011, New York, NY, USA, pp. 985–994. Association for Computing Machinery (2011). https://doi.org/10.1145/2063576.2063718
14. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, New York, NY, USA, pp. 467–476. Association for Computing Machinery (2009). https://doi.org/10.1145/1557019.1557074
15. Latorre, R., Suárez, J.: Measuring social networks when forming information system project teams. J. Syst. Softw. **134**, 304–323 (2017)
16. Lykourentzou, I., Antoniou, A., Naudet, Y., Dow, S.P.: Personality matters: Balancing for personality types leads to better outcomes for crowd teams, pp. 260–273 (2016)
17. Majumder, A., Datta, S., Naidu, K.: Capacitated team formation problem on social networks. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2012, New York, NY, US, pp. 1005–1013. Association for Computing Machinery (2012). https://doi.org/10.1145/2339530.2339690
18. Mannix, E., Neale, M.A.: What differences make a difference?: The promise and reality of diverse teams in organizations. Psychol. Sci. Publ. Interest **6**(2), 31–55 (2005). https://doi.org/10.1111/j.1529-1006.2005.00022.x. pMID: 26158478
19. Retelny, D., Robaszkiewicz, S., To, A., Lasecki, W.S., Patel, J., Rahmati, N., Doshi, T., Valentine, M., Bernstein, M.S.: Expert crowdsourcing with flash teams. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST 2014, New York, NY, USA, pp. 75–85. Association for Computing Machinery (2014). https://doi.org/10.1145/2642918.2647409
20. Salehi, N., McCabe, A., Valentine, M., Bernstein, M.: Huddler: Convening stable and familiar crowd teams despite unpredictable availability, pp. 1700–1713 (2017)
21. Wax, A.: Self-assembled teams: attraction, composition, and performance. Ph.D. thesis, Georgia Institute of Technology (2015)
22. Zheng, Z., Pinkwart, N.: A discrete particle swarm optimization approach to compose heterogeneous learning groups. In: Proceedings - IEEE 14th International Conference on Advanced Learning Technologies, ICALT 2014, pp. 49–51, September 2014