

Chapter 1

Team Formation in Social Networks*

Meenal Chhabra, Sanmay Das, and Boleslaw Szymanski

Abstract It is now recognized that the performance of an individual in a group depends not only on her own skills but also on her relationship with other members of the group. It may be possible to exploit such synergies by explicitly taking into account social network topology. We analyze team-formation in the context of a large organization that wants to form multiple teams composed of its members. Such organizations could range from intelligence services with many analysts to consulting companies with many consultants, all having different expertise. The organization must divide its members into teams, with each team having a specified list of interrelated tasks to complete, each of which is associated with a different reward. We characterize the skill level of a member for a particular task type by her probability of successfully completing that task. Members who are connected to each other in the social network provide a positive externality: they can help each other out on related tasks, boosting success probabilities. We propose a greedy approximation for the problem of allocating interrelated tasks to teams of members while taking social network structure into account. We demonstrate that the approximation is close to optimal on problems where the optimal allocation can be explicitly computed, and that it provides significant benefits over the optimal allocation that does not take the network structure into account in large networks. We also discuss the types of networks for which the social structure provides the greatest boost to overall performance.

Department of Computer Science,
Rensselaer Polytechnic Institute, Troy, NY
{chhabm, sanmay, szymansk}@cs.rpi.edu

* This work was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 and in part by an NSF CAREER Award (0952918). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies either expressed or implied of the Army Research Laboratory. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

1.1 Introduction

Good team-formation is one of the keys to success of any organization. Many researchers emphasize that the performance of an individual in a group depends not only on her skills but also on her relationships with other members of the group [3, 6]. Relationships between team members do not necessarily have to be friendship relations. The important question is whether members work together synergistically – how compatible they are in collaborative environments (in fact, there is empirical evidence, at least among MBA students, that friendship within teammates is *negatively* correlated with performance [1]). Increasingly, organizations are attempting to optimize team composition, for example by taking personality types into account when forming teams to work on tasks [3].

We are interested in the problem of optimal allocation of members to tasks within an organization, taking into account their network of relationships with others. We consider a model where there are multiple task types, and each member has a certain cognitive ability for each task type. Each member is therefore characterized completely by her ability to perform each of the different types of tasks. The organization has a set of projects to be performed, each project consisting of a set of tasks of different types. Accordingly, each task is characterized by its value, its type, and the project to which it belongs. The motivating idea is that the organization is a group of experts (consultants in a service company, or analysts in an intelligence office). Each expert has expertise in a particular task type regardless of the project to which this task belongs. For example an accounting consultant can analyze finances of different companies or transportation system analysts can analyze military movements in many regions or countries. Experts on different task types may have a synergistic effect on each others' performance when they are allocated to the same project by fruitfully sharing information that could be valuable for different task types.

We consider the problem of optimal allocation of members to tasks in this framework. We introduce a model that captures the elements described above, and then demonstrate that taking social network structure into account can have significant benefits in terms of the overall optimality of task allocation. We introduce a greedy algorithm for the problem of task allocation taking social network structure into account, and demonstrate experimentally that it is a good approximation to the optimal allocation on small graphs. Then, we show that it achieves significant benefits on large graphs compared with the optimal solution that does not take social network structure into account. Finally, we use this greedy algorithm to explore the properties of different kinds of social networks, and find that the most affected graphs are small world networks, followed by random graph networks, and the least affected graphs are preferential attachment networks. Our work is related to previous papers that demonstrate the effect of underlying social graph structure on team performance [5, 4]. It is perhaps closest to the work of Lappas et al., who also consider the social graph structure of individuals while forming teams [6]. However, their focus is different; in their model, agents have binary skills, and each task is focused on the composition of appropriate skill sets. In contrast, our work focuses on optimal resource allocation in a utility-theoretic framework.

1.2 The Model

There are n experts E_1, \dots, E_n who work for an organization. The social network representing synergistic relationships is given by S . Two experts are connected in S if they boost each others' performance when they work on the same team assigned to a particular project.

There are z different task types. Each expert has a different skill level associated with each of these z task types. The skills of an expert E_i are thus represented by a vector $S_i = (s_{i1}, s_{i2} \dots s_{iz})$ where s_{ij} is the probability that E_i can complete a task of type j successfully. This is a general measure; it may include not only technical skills but also interpersonal or organizational skills. We assume that the manager responsible for task allocation (M) knows the skills vector for each expert.

Time proceeds in discrete steps and at every T units of time, Manager M allocates work to the experts for the next T units; we call this period of T units a 'round'. Manager M allocates m different projects R_1, \dots, R_m . Each project R_i has q tasks T_1^i, \dots, T_q^i . The distribution of task types is the same for each project. Each task T_i^j of project R_j (represented by T_i^j) is associated with a value V_i^j which represents the gains received by the organization for successfully completing the task (this is a direct measure of utility). Therefore, each available task in the organization is characterized by three attributes: task type, project to which it belongs, and value. For convenience, we construct a vector of probabilities of successful completion of tasks in a given project R_k for each expert E_i , $P_i^k : (p_{i1}^k, \dots, p_{iq}^k)$ from S_i , the vector of skills, as defined earlier.

We assume the manager re-allocates each of the q different tasks in all the m projects at the beginning of each round; therefore, overall there are mq tasks to be allocated to the n experts. The values of tasks do not remain the same at each round, so the allocation algorithm has to be run at the beginning of each round. The tasks are designed such that they can be finished in a round. An expert can only be assigned one task.

Network Effects: The social network S of experts is known to the manager. This is not an unrealistic assumption, because while working with these experts, the manager may have acquired this information. There are also certain personality tests, like Myers-Briggs, Kolbe Conative Index etc., which the manager could conduct to discover expert types which could be used to build a network based on compatibilities between different expert types. Now we specify an explicit model of how experts can help each other out in performing their tasks.

Let W represent the friendship adjacency matrix created from the social graph S :

$$W_{ij} = \begin{cases} 1 & \text{if } E_i \text{ \& } E_j \text{ are friends,} \\ 0 & \text{otherwise} \end{cases}$$

Let f_i^k represent the number of friends of E_i among experts assigned a task of project R_k . We model the network effect as a boost in the probability that an expert successfully completes a task. Let $B = 1 - e^{-f_i^k}$ denote a coefficient that represents the

improvement in the performance of an expert resulting from collaborating on a task with friends; B defines the fraction of the performance gap $(1 - p_{ij}^k)$ that is covered by collaboration. By definition, $0 < B < 1$. Then, the boosted probability that expert i successfully accomplishes task T_j in the project R_k , denoted as p_{ij}^{k+} is defined as:

$$p_{ij}^{k+} = \frac{p_{ij}^k}{1 - B(1 - \max(p_{ij}^k, 1/2))} \quad (1.1)$$

so the higher the coefficient B , the greater the boost. The boost is naturally limited by the factor $1 - p_{ij}^k$, and we further cap it at covering the gap of $\frac{1}{2}$ for lower p_{ij}^k 's. The denominator of the boosted probability expression is at most $\min(p_{ij}^k, 1/2)$ so the boosted probability is less than $\min(2p_{ij}^k, 1)$.

1.3 Algorithms For Task Allocation

The manager's goal is to maximize expected utility. Let us define indicator variables a_{ij}^k as follows:

$$a_{ij}^k = \begin{cases} 1 & \text{if } E_i \text{ is allocated task } T_j \text{ in project } R_k \\ 0 & \text{otherwise} \end{cases}$$

Allocation Ignoring Network Effects: If there is no network effect then the performance of any task depends on the selected experts' skill levels. Then the total expected utility is given by:

$$U = \sum_{i=1}^n \sum_{j=1}^q \sum_{k=1}^m a_{ij}^k p_{ij}^k V_j^k \quad (1.2)$$

The objective is to find allocation variables a_{ij}^k such that Equation (1.2) is maximized.

Since the manager knows the skill level of all the experts, this problem reduces to maximum weight matching in a bipartite graph with experts on one side and tasks on the other, as shown in Figure 1.1. The weight of the edge between expert E_i and task T_j^k is equal to the expected value if E_i is assigned to task T_j^k , i.e., the product of the probability of successful completion p_{ij}^k and the value associated with that task V_j^k . In our experiments we use the Hungarian algorithm to solve the maximum weight matching problem and find the optimal allocation.

Note that, even though the manager ignores network effects when deciding on the task allocation, the effects still come into play in terms of overall performance. Therefore, when we compare solutions that take network effects into account, we include the hidden network effect in the utility term after the allocation has been done without including that term when deciding allocation.

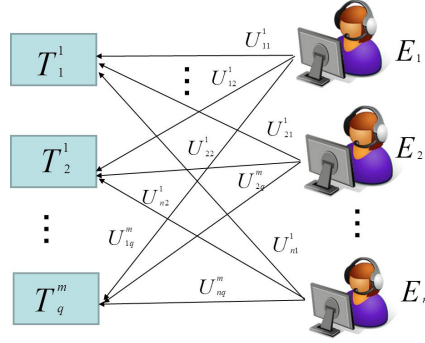


Fig. 1.1 The left side represents tasks and the right side represents experts. The weight U_{ij}^k on the edge between expert E_i and task T_j^k is the expected return value if E_i is assigned task T_j^k . The objective is to find the matching such that the total expected return value is maximized. It is easy to observe that the optimal allocation is the same as the maximum weight matching.

Taking Network Effects Into Account: The total expected utility can be calculated by updating the success probability in Equation 1.2 to take into account network effects:

$$U_f = \sum_{i=1}^n \sum_{j=1}^q \sum_{k=1}^m a_{ij}^k p_{ij}^{k+} v_j^k \quad (1.3)$$

We need to assign a_{ij}^k 's such that the total expected utility U_f is maximized. Let $\text{Opt} = \max_{a_{ij}^k} (U_f)$ represent the maximum expected utility that can possibly be achieved. This problem is computationally hard to solve optimally, so we focus on a greedy approximation algorithm.

A Greedy Approximation: We propose a greedy approximation algorithm that can be used by the organization for task allocation and works as follows. First, construct a weighted bipartite graph with experts on one side and tasks on the other (this is distinct from the graph representing social synergies among experts). As above, the weight on an edge between expert E_i and task T_j^k is equal to the expected return value if the expert E_i is assigned to task T_j^k , i.e., the product of the probability that expert E_i can finish the task T_j^k and the return value associated with task T_j^k . Second, select a link with maximum weight and assign the task on the link to the corresponding expert. Update the success probabilities of all experts connected to this one in the graph of social synergies for all tasks that are on project k . Repeat this process until there are no more tasks or no more experts. The overall complexity of this algorithm is $O(\min(n, mq)nmq)$.

1.4 Experimental Results

Network Models: We consider three standard network models:

1. Random Graph Network (RGN): We use the $G(n, p)$ Erdos-Renyi model for random graph generation.
2. Small World Network (SWN): We use the β model of Watts and Strogatz [7]. The network is represented by a tuple (n, k, β) , where n is the number of nodes in the network, k is the mean degree of each node, and β is a parameter such that $0 \leq \beta \leq 1$ which represents randomness.
3. Preferential Attachment Network (PAN): This network model captures the “rich get richer” phenomenon. We use the mechanism of Barabasi et al. [2] to generate these networks.

Testing the Greedy Algorithm: We compare the performance of the greedy algorithm described earlier with respect to the optimal allocation. We know of no efficient means of computing the optimal allocation when taking network structure into account. As validation, we consider graphs with a small number of experts, so that brute-force search is feasible for finding the optimal solution. We consider a network with 10 experts and two projects, each with five tasks. The average degree is 2. There is only one task type (equivalently, all experts are equally proficient in carrying out any task). The probability of success is 0.2 for any expert to successfully complete any task. The return value of each task is an i.i.d. sample from a Gaussian distribution with $\mu = 1$ and $\sigma = 0.05$.

The results are shown in Table 1.1. We observe that the greedy approximation yields close to optimal results. It is interesting to note the variation in performance with respect to network topology. There is a significant change in the optimal utility when the underlying network creation model is modified. For example, the order of optimal utility is $SWN > RGN > PAN$. This holds even though the average degree is kept constant across the different types of networks. We also observe that for small world networks (SWN), the changes in rewiring factor do not affect the optimal performance achievable; however, there is a slight increase in the performance of the greedy algorithm as the rewiring probability β decreases. One caveat is that these are small networks, so the approximation may be worse in larger networks. Understanding the approximation properties of this algorithm is an interesting open research question.

This experiment also demonstrates that there is a significant advantage to considering network effects during team formation. Although, we cannot calculate the optimal utility for larger networks due to computational costs, if we can show that the utility attainable using the greedy algorithm is significantly higher than that attained when network effects are not taken into account, this is a lower bound on the gains that could be achievable. We thus turn our attention to exploring utility differences between the greedy allocation and the optimal allocation that ignores network effects in larger networks.

Network Type	(k)	β	Pr(Success)	Utility w/o network	Opt Utility	Greedy
SWN	2	0.25	0.2	2.75	3.30	3.20
SWN	2	0.10	0.2	2.76	3.30	3.24
SWN	2	0.00	0.2	2.77	3.29	3.29
PAN	2	-	0.2	2.64	3.06	2.72
RGN	2	-	0.2	2.66	3.11	2.98
RGN	2	-	$\mathcal{N}(0.2, 0.05^2)$	2.70	3.16	3.04

Table 1.1 Comparison of greedy and optimal allocations. k denotes mean degree, ‘Utility w/o network’ is the utility when allocations are made without considering the social network, i.e., using the maximum weight matching approach described in Section 1.3, but the utility is calculated using Equation 1.3; ‘Opt Utility’ is calculated by finding the allocation which maximizes Equation 1.3; Greedy allocates tasks using the algorithm described earlier (utility is again calculated using Equation 1.3). Two key observations are that: (1) Not considering the network structure in allocation is significantly suboptimal; (2) The greedy allocation yields almost optimal performance.

Real World Networks: For the rest of this paper we use the term *Utility Ratio* (UR) as the ratio of the utility achieved using the greedy algorithm and the utility achieved by optimizing the allocation without considering the network effect (although of course the network effect is taken into account in computing the actual utility). Table 1.2 shows that the possible gains from smarter task allocation strategies really become evident when we look at larger organizations. Again note that while the percentage gain from considering network structure in allocation is roughly equivalent for the three types of networks, the overall utility tends to be higher for small world networks especially.

Network Type	(k)	β (Randomness)	Pr(Success)	Utility w/o network	Greedy	UR
SWN	96	0.25	0.2	154.6	192.0	1.24
SWN	96	0.00	0.2	155.0	191.9	1.23
PAN	96	-	0.2	152.0	185.8	1.22
RGN	96	-	0.2	155.1	189.8	1.22

Table 1.2 For medium-size organizations: Experimental results when 480 experts are assigned to 48 projects, each with 10 tasks. ‘UR’ represents utility ratio.

Effects of Team Size and Connectivity: Figure 1.2 shows benefits achieved by considering network effects as a function of team size and of connectivity for random graph networks. The benefit diminishes with increasing team size if the connectivity of the social network remains constant. This is because the chances of having a socially synergistic expert in a large team even with a random allocation (or one that doesn’t take network structure into account) is higher than it would be if teams were small. For connectivity, the utility ratio initially increases because of the increase in socially synergistic experts assigned to the same project, but later decreases as connectivity increases. Again, this decrease is because as connectivity increases, socially synergistic experts are more likely to work on the same project even if network structure is not explicitly a factor in the allocation. We experimented with other types of networks as well but the observations were similar, so we do not report them here for the sake of brevity.

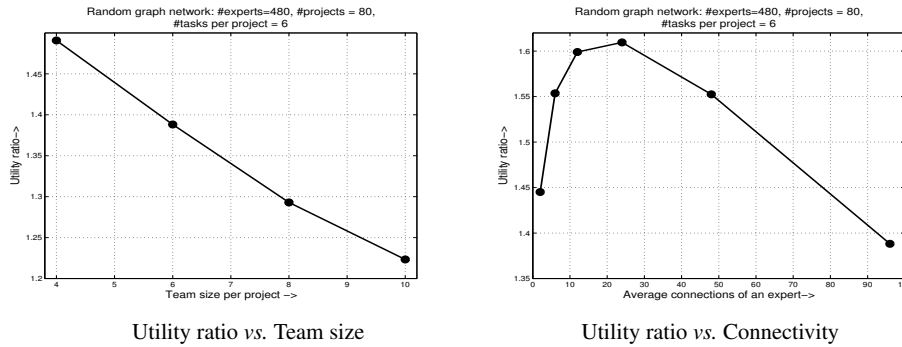


Fig. 1.2 As the team size increases, the likelihood of an expert having a neighbor in his/her team also increases in any allocation; therefore, the utility ratio decreases with the increase in team size. As the average connectivity increases, initially the utility ratio increases because there are more experts available who can boost a given expert's productivity; however, it later decreases. Again, this is because the likelihood of having an expert's neighbors working on the same project by chance also increases in any allocation strategy.

1.5 Conclusions

Our results demonstrate the value of considering social network structure in allocation of tasks in networks of experts. We have also characterized situations in terms of graph structure, connectivity, and team size, in which organizations may find it particularly valuable to explicitly take social network structure into account in determining the allocation of experts to tasks.

References

1. T. Baldwin, M. Bedell, and J. Johnson. The social fabric of a team-based mba program: Network effects on student satisfaction and performance. *Academy of Management Journal*, pages 1369–1397, 1997.
2. A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
3. S. Chen and L. Lin. Modeling team member characteristics for the formation of a multi-functional team in concurrent engineering. *Engineering Management, IEEE Transactions on*, 51(2):111–124, 2004.
4. M. Gaston and M. desJardins. Agent-organized networks for dynamic team formation. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 230–237. ACM, 2005.
5. M. Gaston, J. Simmons, and M. DesJardins. Adapting network structure for efficient team formation. In *Proceedings of the AAAI 2004 Fall Symposium on Artificial Multi-agent Learning*, 2004.
6. 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*, pages 467–476. ACM, 2009.
7. D. Watts and S. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.