

Forming Effective Teams from Agents with Diverse Skill Sets

Kirill Osipov
Department of EECS
University of Central Florida
Orlando, Florida
Email: karl@knights.ucf.edu

Gita Sukthankar
Department of EECS
University of Central Florida
Orlando, Florida
Email: gitars@eeecs.ucf.edu

Abstract—Recent work in crowdsourcing applications based on enterprise social networks (e.g. PeopleCloud) has shown that the group problem solving approach can be extended to enterprise and potentially Internet-wide scales. In crowdsourcing systems, it is desirable to specify skills at a sufficiently fine grained detail level so as to avoid ambiguity. In addition, the skill set may grow at system runtime as participants list or acquire additional skills. However, the increase of the participant skill set leads to an exponential growth in the number of the potential groups where the participant may be included. This paper studies a parallel network adaptation algorithm to address exponential growth in potential solutions arising from the increase in skill diversity. We demonstrate that systems that expect to form groups (e.g. crowdsourcing) by engaging participants equipped with diverse skill sets require more sophisticated network adaptation strategies than what can be expected based on previous research. To address this need, we evaluate a set of network adaptation algorithms for crowdsourcing and present some empirical results from a simulation based study.

I. BACKGROUND

Earlier research in the space of human/computer problem solving groups emphasized use of computational agents as human collaborators and assistants for problem solving scenarios. MAS architecture anticipates that the information about skill sets for agents and human experts is known at application design time. In these applications, identifying the right group for a given task (group formation) can be performed by using a specialized matchmaker agent [1] or through peer to peer sharing of goals and plans across agents [2].

More recently, crowdsourcing and human computation based approaches started to place more emphasis on the role of the human experts in problem solving groups. Crowdsourcing provides a convenient abstraction over the human expert to computational agent integration problem by focusing on human problem solving group formation and treating human-computational agent integration as one of the skills in a human expertise skill set. Systems such as PeopleCloud [3] [4] help build specialized "crowds" for purposes of information technology service delivery. PeopleCloud can collect information about both the task and the skill sets of its participants at system runtime. Information about the participants is collected dynamically from enterprise social network data sources and is processed against constraints (e.g. history of participant

contributions, knowledge expertise, participation levels) to identify the right crowd for a given task.

Selection of an optimal group consisting of agents and humans for a given task is an important aspect of the overall challenge of solving complex problems by relying on multiple problem solving groups working in parallel. In case of MAS, there exists a broad range of research on how to use centralized matchmaking agents for this purpose [5]. However, it is unclear whether these techniques apply at levels of scale and dynamicity in crowdsourcing systems intended to service large enterprises or the entire population of the World Wide Web.

Decentralized, peer to peer based approaches for group formation have greater potential to scale to larger pools of candidate group participants. Some of the MAS solutions have relied on peer to peer based group formation, using plan and goal sharing across computational agents [1], but have not fully addressed the challenges of integration of human experts into the sharing mechanism. In addition, the use of peer to peer systems raises a host of questions relative to the appropriate network connectivity structure between the peers.

Some studies focusing specifically on group formation [6] [7] have quantitatively evaluated the relationship between alternative group collaboration network structures and overall effectiveness of groups in solving problems or performing tasks. These studies examine group formation dynamics, i.e. the changes in collaboration network connectivity over time as collaborators seek to leave and join possible groups in order to improve both their local and system-wide problem solving performance. While the peer to peer based approaches to group formation have demonstrated greater scalability, the agents are faced with the problem of decision making on the basis of limited information about the agents within their local connection vicinity.

In crowdsourcing systems, it is desirable to specify skills at a sufficiently fine grained detail level so as to avoid ambiguity. In addition, the skill set may grow at system runtime as participants list or acquire additional skills. However, the increase of the participant skill set leads to an exponential growth in the number of the potential groups where the participant may

be included¹. This paper introduces a new network adaptation algorithm to address the exponential growth arising from the increase in skill diversity.

II. OVERVIEW

This paper contributes to study of algorithms for dynamic network adaptation for goal (problem solving) oriented networks of human experts and computational agents. We describe a novel algorithm for group formation in a decentralized network, extend an existing group formation model and examine scenarios for network adaptation under skill set assumptions described in a production crowdsourcing system.

III. MODEL FORMULATION

In our simulation model, there is a population of N agents represented by set $A = \{a_1, a_2, \dots, a_N\}$. Each agent is connected to a portion of the agent population via a social network which is modeled using a symmetric adjacency matrix E , where an element of the matrix $e_{ij} = 1$ indicates an undirected edge between agents a_i and a_j . In the paper we distinguish between *first order neighbors* of a_i defined as $N_i^1 = \{a_j : e_{ij} = 1, j \neq i\}$ and *second order neighbors*, defined as $N_i^2 = \{a_k : e_{ij} = 1, e_{jk} = 1, e_{ik} = 0, k \neq i\}$. The degree of an agent a_i is defined as d_i . Each agent a_i has a set of skills S_i which is a subset of size S_A sampled randomly from a uniform distribution over a set of the universe of skills S ; in the previous work such as [6] [7], the assumption is that each agent only possesses a single skill so $S_A = 1$. The agents interact in context of multiple time steps of a simulation. At every time step ts of the simulation, a single task T_{ts} is randomly generated by sampling without replacement from a uniform random distribution of the set of skills S to generate a set of predefined size T_A . A set of agents (group) $G = \{a_k, \dots, a_l\}$ is said to be capable of executing a task T iff $T \subset S_k \cup \dots \cup S_l$. In all of the simulations described in this paper, $S_A < T_A$ so that more than one agent is required to execute any given task. Every T_{ts} is broadcast to all N agents and more than one group of agents may have the skills (potential) needed to execute T . As described in more detail later in this paper, the quantity of these potential groups is one of the key metrics for evaluation of alternative network adaptation algorithms.

The model avoids concurrency issues by merging the steps related to formation of a group and execution of a task into a single time step of the simulation. Once all of the potential groups are identified, every agent commits to (joins) one of the groups using the algorithm described later in the paper. As long as a group has enough committed agents capable of executing T_{ts} then the group is considered to have executed T_{ts} at the conclusion of the ts step. Given the focus of this paper on Internet scale group formation, this formulation

permits the possibility that multiple groups may complete the task in parallel. While it is possible to introduce coordination or election mechanisms to ensure that a task is performed only once, this topic is outside the scope of this paper.

A. Initial Social Network Connectivity

To establish connectivity, all agents are randomly assigned a position on a square grid with side of size \sqrt{N} (based on the N value from Table I). Distance between the agents is measured under an assumption of toroid connectivity between grid's edges using Manhattan distance measure, D_{ij} . For every agent a_i , an undirected edge e_{ij} is established to every other agent a_j , as long as D_{ij} is less than or equal to a predefined constant D (see Table I). For every agent a_i , the initial N_i^1 connectivity configuration as explained here is identical for all simulation scenarios described in this paper. The algorithm implementing this connectivity configuration is identical to the random geometrical graph generation approach followed by [7] [8].

B. Candidate Group Selection and Group Formation

Since agents are operating on the basis of limited information about potential groups, every agent must make a decision about which groups can solve the tasks and which groups the agent should join to execute the task. At every time step of the simulation, an agent a_i knows only which skills are available to agents in its first order social network neighborhood N_i^1 . Before joining or forming a group, an agent a_i must compute $G_{i,ts}$ which is a set of potential groups that have the skills needed to execute T_{ts} . Agent a_i computes the intersection of the sets S_i and T and whenever the intersection of the two sets is non-empty (i.e. the agent has at least one skill needed for the task), attempts to perform the computation of a set of candidate groups that the agent expects to execute the task. The computation problem is equivalent to an optimization by minimizing the sum of indicator functions which specify whether an agent $a_j \in N_i^1$ should be in the potential group set:

$$\text{minimize } \sum_{j=1}^N I_{G_{i,ts}}(a_j) \quad (1)$$

subject to the constraint

$$T_{ts} \subset \bigcup_{k \in G_{i,ts}} S_k \quad (2)$$

The equations 1 and 2 are an example of the well known minimum set covering problem (*MinSetCover*). In this instance, the objective is to find the smallest possible set of agents such that their respective skills S_j "cover" the skills required for a given task T . While set covering is a classic example of an NP-complete problem [9], it has well known approximate solutions [10] and in context of bounded social network neighborhoods can be solved exactly and efficiently using industrial scale solvers [11].

¹Consider a social network where a vertex (node) represents an agent and the vertex degree d stands for agent's awareness of d other agents in the network, each edge representing a potential collaboration. Under the assumption that the agent represented by the node has S_A skills, the agent may offer any of its $2^{S_A} - 1$ subsets of skills to a collaborator. Considering all of its d potential collaborators, the agent may be a candidate for at most $d(2^{S_A} - 1)$ collaborative groups.

An agent can both initiate its own group and be invited to participate in a group initiated by another agent. Given any two agents a_i and a_j , a_i could be a member of $G_{i,ts}$ or $G_{j,ts}$. We will refer to groups in $G_{i,ts}$ where a_i is a member as self-initiated and denote them by $SG_{i,ts}$. Other-initiated groups $OG_{j,ts}$, are those where a_i is a member of $G_{j,ts}$. As the union of $SG_{i,ts}$ and those groups in $OG_{j,ts}$ with a_i as a member may contain members of N_i^2 ; given $g_i, g_j \in G$ agents follow a preference policy $Pref(g_i, g_j)$ to select which group to join. The policy is formulated to ensure that agents encourage smaller groups which maximizes the number of groups that can execute a task and within a single time step, prefer groups that have as many agents from the first order neighborhood as possible, which increases the importance of an effective network adaptation strategy. To decide the group to join, the agent performs additional filtering and preference sorting on all of its candidate groups based on a policy where the agent prefers

- smaller groups over larger groups
- groups most similar to its own group proposal
- groups more similar to its first order neighborhood

The policy is defined as:

$$Pref(g_i, g_j) = \begin{cases} g_i & |g_i| < |g_j| & (3) \\ g_i & |G_i \cap g_i| > |G_i \cap g_j| & (4) \\ & \wedge |g_i| = |g_j| \\ g_i & |g_i \cap N_i^1| > |g_j \cap N_i^1| & (5) \\ & \wedge |G_i \cap g_i| = |G_i \cap g_j| \\ & \wedge |g_i| = |g_j| \\ g_j & \text{otherwise} & (6) \end{cases}$$

The procedure followed by the agents in forming a collection of groups to execute a given task is summarized in Algorithm 1 listing.

C. Node Degree Based Network Adaptation

Following task execution and prior to the conclusion of every time step of the simulation, agents have an option to adapt their network connectivity. Agent a_i examines a set of agents N_i^2 such that a_k is a member of N_i^2 iff all of the following are true

- there exists an agent a_j such that it is connected to a_i via e_{ij}
- a_j is connected to a_k via e_{jk}
- a_k is not connected to a_i via e_{ij}
- a_i and a_k are not the same

Intuitively, the formal description above specifies the set of all second-order neighbors (i.e. neighbors of a_i 's neighbors) with exception of those that are also connected to a_i directly. Using the node degree of the agents in A , a_i defines two probability density functions:

$$P(a = a_i) = \frac{1}{|\{A\}|} \quad (7)$$

and

Algorithm 1 Agent Group Selection and Participation Algorithm

```

1: procedure SELECT-JOIN-GROUP( $T_{ts}, A, N$ )
2:   for all  $i \in \{1 \dots N\}$  do  $\triangleright$  iterate over agent index
3:     if  $|S_i \cap T_{ts}| > 0$  then
4:        $SG_{i,ts} \leftarrow MinSetCover(N_i^1, T_{ts})$ 
5:     end if
6:   end for
7:   for all  $i \in \{1 \dots N\}$  do
8:      $OG_{i,ts} \leftarrow \bigcup_{j \in \{1 \dots N\}} \{SG_{j,ts} : a_i \in SG_{j,ts},$ 
9:    $i \neq j\}$ 
10:     $g_{i,ts} \leftarrow Join(SG_{i,ts} \cup OG_{i,ts}, Pref)$ 
11:     $G_{ts} \leftarrow G_{ts} \cup \{g_{i,ts}\}$ 
12:  end for
13:  return  $G_{ts}$ 
14: end procedure
15: procedure JOIN( $G, ComparePolicy$ )
16:   $G_{sorted} \leftarrow ComparisonSort(G, ComparePolicy)$ 
17:   $g \leftarrow Head(G_{sorted})$   $\triangleright$  first element of the sorted list
18:  return  $g$   $\triangleright$  discard the tail of the list
19: end procedure

```

$$P(a = a_j) = \frac{d_j}{\sum_{a_k \in N_i^2} d_k} \quad (8)$$

By sampling from the probability density function 7, a_i selects an agent a_j and deletes e_{ij} thus removing a random first order neighbor. Next, a_i samples from probability density function 8 to select agent a_k and creates e_{ik} edge. This approach is designed to implement the preferential attachment policy described in [7]. Earlier research [6] [8] has addressed the question of an appropriate strategy for selection of the candidate agents for network adaptation as well as selection criteria for identifying connection destinations for the candidate agents. There is evidence that strategies that enable agents to adapt connectivity based on local structural information outperform strategies where agents attempt to model global network performance [8]. In this paper we adopt a structural strategy summarized as follows: if at time step ts an agent a_i does not select and join a group (Algorithm 1), then prior to start of $ts + 1$, a_i will adapt its network connectivity using the preferential policy approach described in this section.

IV. METHOD

In the evaluation of the model, we compare implications of alternative connectivity dynamics on group formation in presence of a variable number of skills that every agent can contribute to a group. This section provides an overview of give related scenarios that have been simulated as part of the study, each scenario focusing on a unique set of network adaptation configurations and strategies. Each of the scenarios has been simulated using a combination of common and scenario specific sets of parameters described in Table I.

TABLE I
SIMULATION SCENARIO PARAMETERS

Name	Value					Description
	Scenario					
	S1	S2	S3	S4	S5	
NumSimulations	256					Number of times each scenario has been simulated. Number of time steps per scenario, also number of tasks randomly generated per simulation. Number of agents in the simulation scenario social network Manhattan distance for initial random geometric graph agent connectivity such that a_i is connected to all neighbors a_j that have $D_{ij} \leq D$ Number of distinct skills in the simulation scenario Total number of distinct, randomly chosen skills per task
NumSteps	128					
N	64					
D	2					
$ S $	16					
T_A	8					
S_A	1	1	4	4	4	Total number of distinct, randomly chosen skills per agent

For all of the scenarios we have collected the same set of measurements to compare the impact of skill set diversity in group formation to static and preferential node attachment connectivity models.

The first scenario (S1) assumes absence of any connectivity changes relative to the initial, randomly created agent network. To reproduce results from [8] we also restrict the number of skills per agent with $S_A = 1$. S1 serves as a baseline to showcase average performance of the single skill agent network across multiple simulations with a nontrivial sample set of possible initial random geometric graph configurations.

The second scenario (S2), follows [7] to reproduce the node degree based network adaption under the $S_A = 1$ assumption. The first step of every simulation under this scenario assumes random geometric graph connectivity described in section III-A. Prior to the beginning of the second and prior to every subsequent step of the simulation in this scenario, every agent a_i updates its set of edges as described in the section III-C on adapting network connectivity using the preferential policy approach.

The third scenario (S3) extends S1 through introduction of diverse agent skill sets described earlier in this paper. No network adaptation is performed in this scenario as it is designed to serve as an illustration of the impact of a larger number of skills in a system on the group candidate identification and group formation performance.

The fourth (S4) and fifth (S5) scenarios both extend S3 using alternative network adaptation strategies. S4 uses an

exact implementation described in the section III-C while S5 performs network adaptation for every agent in the network. The latter scenario's approach is designed to illustrate performance of a purely random preference attachment policy which places no consideration on the overall system performance. Introduction of this scenario is motivated by [8] which demonstrates that structural policies may outperform adaptation strategies involving agent estimation of system performance based on locally available information.

Since the results of the variations of the network adaption policy on the network structure are similar across all of the scenarios, they are illustrated with the example shown on Figure 1. As a consequence of the configuration parameters from Table I, every agent a_i is instantiated (at $t = 0$) with $d_i = 16$. The figure shows that the network adaptation policy progressively modifies the node degree distribution towards a concentration of high degree nodes with "fat tails" of single digit degree nodes. The degree distribution does not change significantly beyond $t = 30$ to warrant additional illustrations. In the figure, every distribution is paired with a corresponding illustration of the underlying graph providing a representation of the network adaption algorithm effect on the graph structure.

Every scenario is studied through execution of multiple simulation sessions, each session consisting of a fixed and equal number of steps for all simulations. For every simulation step we measure the number of candidate groups identified for every agent as well as the number of groups formed during the step. In addition to tracking the descriptive statistics (mean, standard deviation, and minimum, maximum) for these variables on per step basis, we also compute statistics of these variables across all steps in a simulation and for all of the simulations in a scenario. The scenario scope measurements are needed to reduce potential bias due to selection of random geometric connectivity graphs as initial conditions in the first step of every simulation.

V. DISCUSSION

When comparing the results for S1 and S2, we confirm the observation that the use of agent performance based preferential node attachment policy (in S2) leads to both a higher number of groups formed and of candidate groups where the agents could participate. Compared to [7] [8] the mean values of both variables are lower due to the difference between the total number of skills in the simulations. As argued by earlier research, use of structural, preferential policy based network adaptation leads to significantly better performance in these scenarios resulting in over an order of magnitude improvement in the number of groups formed and number of candidate groups per agent.

As argued earlier in this paper (footnote 1), introduction of a larger number of skills per agent can exponentially increase the number of potential groups where the agent can participate. As shown by the results for S3, even without any network adaption, increase in skill set cardinality leads to an increase of approximately three orders of magnitude for the mean number

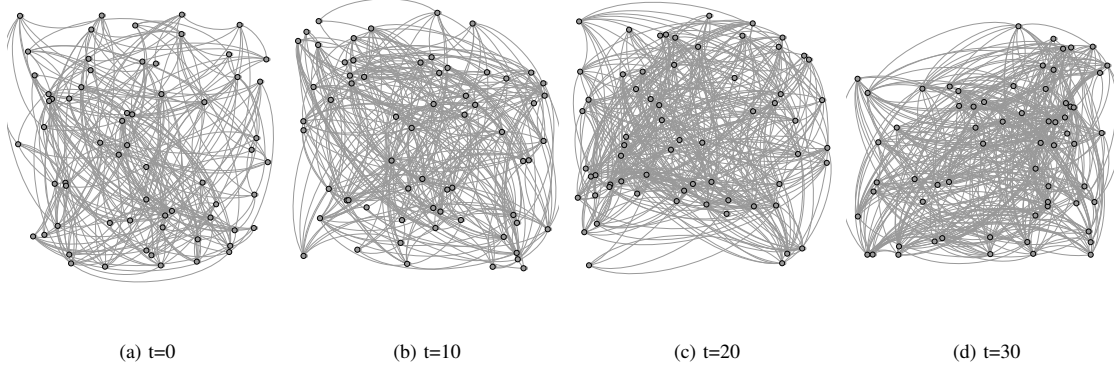


Fig. 1. Simulation results. Subfigures a-d provide a visual representation of the network connectivity structure for time steps 0 through 30 reflecting the evolution towards greater node degrees.

of the candidate groups per agent with a parallel increase in the mean number of groups formed across the simulations in the scenario.

Given the results from research on network adaption and evidence from S2, one may expect further improvement to the S3 results through the introduction of preferential attachment policy under the assumption of multiple skills per agent. However results of the simulation demonstrate that not to be the case. In S4, the mean number of groups formed across simulations is not statistically significantly different from the same number in absence of the policy, while the number of the candidate groups per agent measurably dropped. Further, S5 demonstrates that use of agent performance for network adaption is not measurably different from using a purely random preferential attachment policy.

Note from Figure 1 that over the course of the simulation, an increase of the average weighted degree of the social network was less than by a factor of two. One interpretation of this result suggests that if the number of skills in the system $|S|$ increases linearly while the number skills per agent S_A stays constant, the preferential node attachment network adaptation policy becomes less effective as the total space of possible tasks grows exponentially.

VI. CONCLUSION

An increasing number of systems seek to exploit information available in Internet scale social networks to identify teams of experts for knowledge discovery [4] or for task-oriented crowdsourcing [3]. When considering performance of such systems in terms of their ability to organize groups, it is reasonable to expect that results from studies on group formation [7] [8] should extend to more sophisticated models of individual agents and their contributions to potential groups. Our simulation based study demonstrates that models of agent capabilities that allow for runtime changes to agent skill sets (for example in crowdsourcing systems like PeopleCloud [3]) introduce scaling difficulties for traditional network adaptation policies based on preferential node attachment.

We have shown that use of more detailed skill set descriptions per agent (i.e. in terms of a number of skills per agent) is desirable as it motivated by potential crowdsourcing applications [4] and has a net positive effect on the number of the candidate groups where an agent can contribute its skills and the total number of groups that can be formed by a system. However further research is needed to more precisely analyze and quantify the impact of preferential attachment policies, and to research alternative network adaptation strategies.

REFERENCES

- [1] T. L. Lenox, T. Payne, S. Hahn, M. Lewis, and K. Sycara, "Agent-based aiding for individual and team planning tasks," in *In Proceedings of IEA 2000/HFES 2000 Congress*, 2000.
- [2] J. A. Giampapa and K. Sycara, "Team-Oriented Agent Coordination in the RETSINA Multi-Agent System," Robotics Institute, Carnegie Mellon University, Tech. Rep., 2002.
- [3] M. Lopez, M. Vukovic, and J. Laredo, "PeopleCloud Service for Enterprise Crowdsourcing," *2010 IEEE International Conference on Services Computing*, pp. 538–545, Jul. 2010.
- [4] P. Ypodimatopoulos, M. Vukovic, J. Laredo, and S. Rajagopal, "Server Hunt: Using Enterprise Social Networks for Knowledge Discovery in IT Inventory Management," *2011 Annual SRII Global Conference*, pp. 418–425, Mar. 2011.
- [5] K. Sycara, S. Widoff, M. Klusch, and J. Lu, "Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace," *Autonomous Agents and Multi-Agent Systems*, vol. 5, no. 2, pp. 173–203, Jun. 2002. [Online]. Available: <http://dx.doi.org/10.1023/A:1014897210525>
- [6] M. Gaston and J. Simmons, "Adapting network structure for efficient team formation," *Proceedings of the AAI 2004 Fall*, 2004.
- [7] M. Maghami and G. Sukthankar, "An agent-based simulation for investigating the impact of stereotypes on task-oriented group formation," in *SBP'11 Proceedings of the 4th International Conference on Social Computing, Behavioral-cultural Modeling and Prediction*, College Park, MD, USA, 2011.
- [8] M. E. Gaston and M. DesJardins, "Agent-organized networks for dynamic team formation," *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems - AAMAS '05*, p. 230, 2005.
- [9] R. M. Karp, *Reducibility Among Combinatorial Problems*. Springer Berlin Heidelberg, 2010.
- [10] V. V. Vazirani, *Approximation algorithms*. New York, NY, USA: Springer-Verlag New York, Inc., 2001.
- [11] H. D. Mittelman, "Recent benchmarks of optimization software," in *22nd European Conference on Operational Research, Prague, Czech Republic*, 2007.