Trust and Interaction-type Considerations in Multi-objective Team Compositions for Human-computation

Mirela Riveni and Schahram Dustdar
Distributed Systems Group
TU Wien
Vienna, Austria
{m.riveni,dustdar}@infosys.tuwien.ac.at

***Abstract*— The past years have shown that human computation can be crucial in a variety of applications, and not only with individual work such as in crowdsourcing but also coordinated team-based work in various settings, enterprise or ad-hoc ones. As distributed intelligence systems are increasingly designed to include people as task-executing stakeholders along with software services, the need to automate the human involvement in terms of coordination is real. Research and industry are focused on automating the process of team-formation as well as team-coordination. We investigate this topic further in this paper. While existing work have focused on interactions in general in combination with metrics such as cost, we argue that interactions in team formations should be considered in the context of their types. In this paper we propose strategies and algorithms that consider *interaction types* when forming teams, in combination with trust. We report on results of our conducted experiments with both synthetic and real data regarding the proposed team formation strategies.**

I. Introduction

Human-computation applications as well as mixed systems that incorporate human-computation are approaching their peak point, as industry and academia include people as stakeholders in various types of systems and applications for tasks that cannot be executed by software. as well as for tasks that can be executed more efficiently in a common ”collaboration” with machine-based resources. Crowdsourcing applications have gained their momentum, and along with developments in more complex modeling and programming of socially-enhanced applications [1], we are heading to complex systems where software and people work together. Focusing on the people part, there has been research on getting beyond the individual task-execution model as in crowdsourcing and focusing on coordinated team-work, where the coordination can be managed with fully automated processes or including humans-in-the-loop.

Distributed information processing by human-based collectives for complex tasks that can not yet be solved by computers, in cooperation with computers is already helping in multiple application areas, such as pattern recognition, object recognition in images etc. Collective human computation has been listed as one of the building-blocks of distributed intelligence in systems where people work in cooperation with software and hardware [2], [3], [4]. In this regard, we direct our attention to the topic of collective-based tasks that cannot be solved by a single human and require efficient teams with the right expertise. The challenge of forming productive teams, which will collaborate efficiently is still salient and requires attention. In this paper we investigate this problem.

Online collective work is not a new topic, but topics such as management mechanisms for human provided services in collectives, including people services in service-oriented architectures, automating processes with human-in-the-loop where people perform tasks, and programming human-provided services [5], are fairly new and should be addressed if we want to design collective adaptive systems with various types of resources that will lead (but also adapt to) social changes. To address the aforementioned challenges, suitable collective formation mechanisms are needed. Existing works have focused on team formation strategies, mostly based on the appropriateness of skills and team connectivity based on existing interaction analysis between possible team participants. They provide solutions to team-formation in terms of minimizing coordination cost, team size, and workload. Authors in [6] for example consider that the frequency of previous interactions are one of the indicators for efficient teams. As far as we know, existing work has focused on efficiency metrics, but focusing only on one type of connections between team members, which help in assessment of coordination cost between team members, general interaction frequency or both, while we chose to look at interaction in more details, namely focusing on *interaction types*. Thus, in this paper we consider two types of interaction networks, communication interactions which include only natural language communication between people, and coordination interaction, which include task-related interactions, such as a delegation of one task from one person to another. To generalize, our hypothesis is that if we are to assemble efficient collectives, a group of individuals who have the required skills for a specific goal can not be formed without considering multiple underlying (possible) relation types between the individuals. Our contributions in this paper are:

Proc. 2018 IEEE 17th Int'l Conf. on Cognitive Informatics & Cognitive Computing (ICCI*CC'18)
Y. Wang, S. Kwong, J. Feldman, N. Howard, P. Sheu & B. Widrow (Eds.)
978-1-5386-3360-1/18/$31.00 ©2018 IEEE

- a novel approach to team formation considering *interaction types*,
- a team-formation algorithm by ranking experts based on trust and weighted interaction values,
- a team formation Genetic-algorithm based on data of previously formed teams that have already completed projects.

II. Team Formation based on trust and multiple interaction types

A. Problem Statement

The specific team-formation problems that we investigate in this paper can be defined as in the following definitions.

Problem 1: Given a network of experts with information about their previous performance and interactions, and a given project-requirement, find a collective/team of experts from the network who can execute the project effectively.

Problem 2: Given a set of existing teams with information about their previous performance and interactions, and a given project-requirement, find and select a collective/team of experts who can execute the project effectively.

B. Model

To approach the stated problems we consider two types of information crucial for our team-formation model and strategies:

1) **interaction links**, and
2) **subjective and objective data regarding expert performance**, ie., metrics that include votings and recommendations, acceptance or rejection of tasks assigned from the system, expert productivity etc.

Regarding the interaction links, we are distinguishing two types of them in the context of a specific collaboration:

1) **communication links**, which include messages in natural language; and
2) **coordination links**, which include task-related interactions such as delegation of tasks (control flow with human in the loop).

Regarding performance data of experts, we consider an overall trust metric for experts and teams over a specific period of time. In [7], we have introduced a trust model from two perspectives: a trust metric to assess the trustworthiness of an individual/member of a team and a trust metric assessing the trustworthiness of a team. Both trust metrics are calculated in the context of the goal for which a team has been assembled, ie., a team member has a local trust score in the context of a specific team within which he/she works providing a specific skill-type, whereas a global trust metric is relevant across all teams he/she has worked within a time interval. We regard the actuality of having an available global trust score for people in the context of a particular skill as an effective approach for initial expert recommendation for inclusion in teams, and we consider trust in addition to communication and coordination links as one of the indicators of team efficiency. We discuss the trust metric later in this paper again, because we used it in our experiments but we do not describe the actual formulas as we have defined them in the aforementioned work.

Specifying our expert environment model now, we denote a pool of experts as $P = \{p_1, p_2, p_3 ... p_n\}$, a team of people (a collective) as $C \subseteq P$. and a set of initial tasks for a specific team C as $T = \{t_1, t_2, t_3 ... p_n\}$. Every expert in the pool P has her/his own profile properties. The important ones for us in this work are $s(p)$ denoting the skill type of p, $stt(p)$, denoting the trust score of p, and $c(p)$ denoting the labor cost of expert p for skill $s(p)$. As aforementioned, we consider two interaction network types: a) a communication network, in which the interaction intensity between two people is denoted by weight $w_m(p_i, p_j)$, and b) a coordination network, in which the intensity of interactions in terms of coordination is denoted by weight $w_c(p_i, p_j)$. The shortest path between two people in the network is denoted by $d(p_i, p_j)$. The edges within a team are denoted with the set E, and the total number of edges is denoted with $|\mathfrak{E}|$. A communication edge between two team-members is denoted with e_m, while a coordination one with e_c. The total number of edges between p_i and p_j is denoted with $e_{m,c}$. Thus, $w_m(p_i, p_j) = \frac{\sum e_m}{\sum e_{m,c}}$, and $w_c(p_i, p_j) = \frac{\sum e_c}{\sum e_{m,c}}$. A pair of two team members within a team between who exists at least one interaction link is denoted with $(p_i, p_j)_e$. We denote the collection of teams, in which a specific person has been a member (over a specific time period), with $\mathfrak{U}^\tau = \{C_1^\tau, C_2^\tau ... C_n^\tau\}$.

The requirements for team-members (for each particular task) are denoted with $I(s, r, c)$ individually, where s is the skill-type, r is the reputation of a member of a team in the context of a particular skill, and c is labor cost. The initial network consists of multiple experts who have previously worked together in teams and consequently created different types of interaction links. From this network we distinguish two types of interactions, one based on *communication* and another based on *coordination* interactions.

Because we consider trust scores of experts, in this paper we hypothesize that given a high trust score of experts, a high number of interactions between experts in the context of message exchanges and/or task-related coordination is an indicator that those experts can work well with each other. Thus, in our model we aim to form teams with a high value for the weighted values of these type of interactions along with a high value of trust. We denote the total weight of interactions within a specific team, with $W_m(C)$, whereas the total weight of interactions in terms of task-coordination as $W_c(C)$.

Each node in our model is defined with a trust score called Socio-technical trust (STT) score, which is a complex metric that includes multiple measurable metrics indicating an expert's performance, as well as social-trust scores from votes of collaborators and customers indicating collaboration satisfaction. The team trust score is a normalized aggregated trust score from the member expert trust values. The threshold value for the team STT is denoted with δ. Now we have all the indicators over which we want to execute a team-formation algorithm (notations are given in Table I). Thus, optimally, we need to form a team that:

- includes experts with matching skill requirements
- satisfies $STT \geq \delta$
- minimizes the team diameter, $D(C)$
- maximizes the weight of communication interactions between team-members, $W_m(C)$
- maximizes the weight of coordination interactions between team-members. $W_c(C)$

The team-weight of communication interactions is defined as:

$$W_m(C) = \frac{1}{\sum(p_i,p_j)_e} \sum_{i \neq j} w_m(p_i, p_j) \quad \forall p_i, p_j \in C \tag{1}$$

The team-weight of coordination interactions is defined as:

$$W_c(C) = \frac{1}{\sum(p_i,p_j)_e} \sum_{i \neq j} w_c(p_i, p_j) \quad \forall p_i, p_j \in C \tag{2}$$

However, to have the optimal team is difficult due to the dynamic nature of human computation networks, and thus, we approach the problem of team formation in two ways: a) by using an Analytic Hierarchy Process (AHP) method in ranking the importance of the above constraints, and using a non-evolutionary Pareto-based approach to rank formed teams, and b) by using an ML Genetic Algorithm approach, with the presumption that we have team-based historical information, for assembling new efficient teams out of existing teams and their data. We compare both approaches in the experiment section. The team requirements and consequently, indicators of efficiency, in our model are: $C_(s), STT, D(C), W_m(C)$, and $W_c(C)$.

Let us look at an example to further motivate our problem and the reason behind our argument that interaction types provide valuable information regarding team collaborations and their efficiency. Figure 1 shows two same sections of an expert network with two teams on each of them. The lighter links represent communication interactions, and the ones in dark blue represent two types of interactions existing between team members, communication and coordination, and edge values are given respectively as well for illustration purposes. In sub-figure 1(a) the team formed by A,B,C,D,K is more favorable than the one formed by K,E,F,G,H. Both teams have the same number of nodes and close values for the total team socio-technical trust scores (if we have a summed value over all node STT-values), which are presented by the values in red. In the same sub-figure the favorable team is better connected, has more connections between team members. Thus, this case is fairly intuitive. However, looking at sub-figure 1(b), we see that the more favorable team is the one that includes K,E,F,G,H. The STT score does not have a high discrepancy, but the values for the edge weights for both interaction types are much higher then the ones for the team with nodes A,B,C,D,K. This means that a ranking algorithm or a team-formation algorithm should take into consideration the case when a team that is "less connected" has more communication and coordination interactions between connected members, because this might be an indicator of effective collaborations and higher trust between team members. Hence, it is these type of specific cases that bring us to consider the type of interaction links, and the distance between team members/diameter in a multi-objective team-formation strategy that we present in Algorithm 1.

TABLE I
NOTATIONS

Notation	Description
P	Pool of people/experts
C_i	Team $i \subset P$
e_m	a communication interaction-type link between two exerts
e_c	a coordination interaction-type link between two experts
$(p_i, p_j)_e$	a pair of team members with at least one interaction link
$w_m(p_i, p_j)$	communication link strength between expert p_i and expert p_j
$w_c(p_i, p_j)$	coordination link strength between expert p_i and expert p_j
STT	Socio-technical trust score of a team
$W_m(C)$	Normalized weight of communication interactions (message-exchange) between team-members
$W_c(C)$	Normalized weight of coordination interactions between team-members
$D(C)$	Team diameter

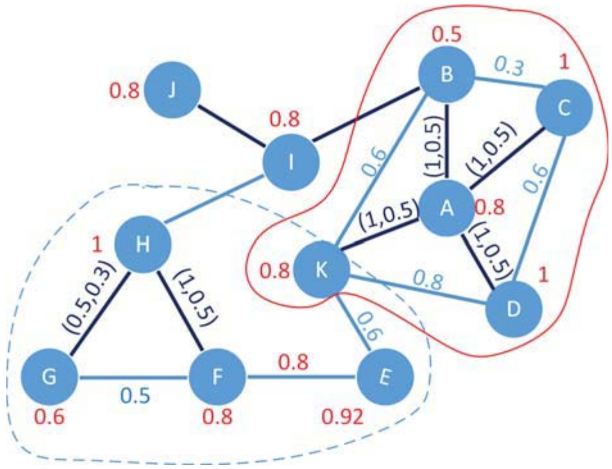
C. Expert role connected to different types of interaction links-Discussion

One reason of separating communication interactions and coordination interactions is because these two types of interactions separately can inform about the member roles. More specifically, one application of the analysis of these two interaction types is the differentiation between a leader of the team or unavailability of a team member. For example, if a member of a team has a high trust score and it has more communication interactions in terms of message-exchanges in natural language and more coordination interaction, then these three metrics can indicate that the member might be a leader of the team, as it communicates more but also manages the control flow, e.g., delegates tasks to other members of the team. If on the other hand, a member of the team does not have a high number of communication messages and has high coordination-based interactions, it can mean that the member has been unavailable because it has not communicated and it had rejected assigned tasks or it has delegated tasks to co-members. However, these are only intuitive assumptions, and we plan to investigate the problem of role-identification based on interaction types in our future work.

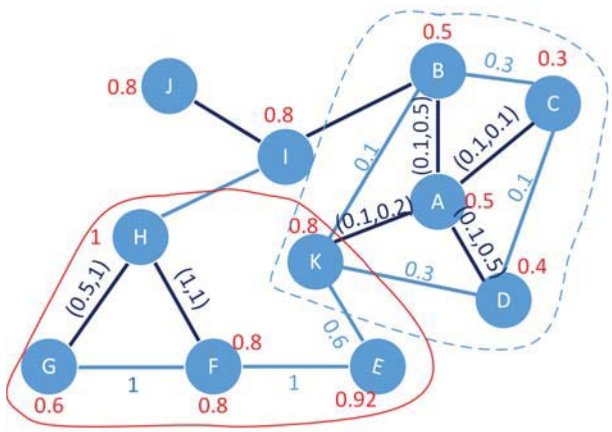
III. Programming Team Formation

With Algorithm 1 we present a strategy to form teams out of a ranked list of available experts, while Algorithm 2 presents a genetic algorithm method to form teams out of an existing initial set of teams.

Algorithm 1 ranks all experts by trust separating them by skill. We utilized a trust score such as the one presented in [7]. To specify, the trust metric is a metric composed of two complex metrics, each composed by atomic ones, specifically STT is composed of *social trust*, which in our implementation is represented by votes from other experts with whom an expert has collaborated, and *performance-based trust*, which is calculated based on atomic tasks such as



(a) Favorable and non-favorable teams in the case when a better connected team has favorable interaction weights. The bordered area with a full line represents a favorable team, whereas the bordered area with a dashed line represents a non-favorable one.



(b) Favorable and non-favorable teams in the case when a better connected team has less favorable interaction weights.The bordered area with a full line represents a favorable team, whereas the bordered area with a dashed line represents a non-favorable one.

Fig. 1. The lighter lines represent interactions only in terms of communication with message-passing in natural language, the darker lines represent the presence of two types of interactions: communication and coordination. Edge values represent interaction weights, a single value represents communication weight, while sets of two values represent weights of communication and coordination, respectively. Values in red represent STT scores of experts.

success rate and effort. Because the trust metric is a composite one, we use an AHP based algorithm to rank experts as AHP provides a hierarchical weighting method for parameters that are comprised of several others. Next, a team is formed in such a way that based on skill types an expert with the highest trust score for each skill type is included within the team. After forming multiple such teams the algorithm compares teams based on communication and coordination interaction weights, and team diameter, and ranks the teams maximizing communication and coordination interactions, and minimizing the team diameter.

Algorithm 2 is a genetic algorithm, which takes as input a number of appropriate teams for the tasks. A population is represented by a set of teams, each team having the same number of team-members. The set of genes considered for each team are the values for $W_m(C)$, $W_c(C)$ and $D(C)$. For the fitness function, we set the following requirements: $STT >= 0.5$, $((W_m(C) + W_c(C))/2) >= 0.5$, and $D(C) <= 0.5$. The algorithm gives a sorted list of teams based on a comparison function of our communication, coordination, and team diameter objectives.

Algorithm 1 Team-formation algorithm utilizing AHP for ranking experts and non-evolutionary Pareto based team selection

Require: Graph G(P,I), T, $I(s, r, c)$, C_r
$U = \emptyset$, $PC = \emptyset$;
for all task in T with $I(s_k, r, c)$ **do**
    $rankWithAHP(P)$
    /* store ranked experts in separate lists by skill type */
    $listP(s_k) \leftarrow$ ranked experts with skill s_k
    for all $listP(s_k)$ **do**
        /* create x number of teams by choosing experts from the same rank, from each list with ranked experts based on skill, in descending order */
        $C_i \leftarrow listP(s_k).getFirstListElement()$
        $U \leftarrow C_i$
    for all C in U **do**
        /* check for all objectives: $minD(C) \in CO, maxW_m(C) \in CO, maxW_c(C) \in CO$ */
        for all o in CO **do**
            ParetoCompare(C_i, C_j, o)
        $PC \leftarrow C_i$
    Rank(PC)
return PC

Algorithm 2 Team-formation Genetic Algorithm

Require: PC
$population = \leftarrow PC.size()$, $newPopulation = \emptyset$;
for all teams in PC **do**
    team $\leftarrow$ setGenes($W_m(C), W_c(C), D(C)$)
    $population.add(team)$
while generationCount $\leqslant$ maxSteps **do**
    for all team in PC **do**
        calculateFitness()
    selectParentsByRouletteWheel()
    newChildrenCrossover()
    if mutateVar $\leqslant$ mutatePercent **then**
        newChildrenMutation()
    /* calculate fitness for children/*
    calculateFitness()
    $newPopulation \leftarrow newPopulation.add(child)$
    $population \leftarrow newPopulation$
    $sort(population)$

IV. Experiments

A. Evaluation with synthetic data

We implemented Algorithm 1 for team formation and selection, combining AHP for team formation and Pareto-based efficiency for selecting the most appropriate team, based on pre-set requirements. Thus, we used an a priory decision making approach with a human decision-maker (DM), considering a scenario where teams are not formed ad-hoc but with a customer request. We used the AHP method for ranking people based on skill-types and trust [1]. We modeled and generated a pool of human profiles with skills and different metric values, such as values for cost per task, and a socio-technical trust score. Each person has a single skill. We modeled and generated tasks, where each task was associated with a single skill. Every person was modeled to have a connection with a (random) number of other experts from the pool of resources indicating a previous collaboration. Every connection/edge had two weighted values, one indicating a communication interaction and the other indicating a coordination interaction.

Algorithm 1 has two blocks, the first one forms and ranks teams based on AHP analysis of two requirements for teams: cost per task and socio-technical trust score (reputation) of team members (lines 1-9); the second block (lines 10-15) ranks teams based on Pareto analysis of three pre-set objectives: higher $W_m(C)$ and $W_c(C)$, and lower $D(C)$. We set the requirement for the STT of the team as $STT \geq 0.5$ in the team formation with AHP. We generated 10 initial tasks, with 6 tasks having the same skill and 4 tasks having different skills, so as to simulate for example realistic teams such as development, where development skills are represented more often, while testing, and design skills for example are represented with fewer people. Thus, the size of each team is the same and does not influence the formation, ranking and selection process. The teams were formed such that we ranked and matched people to the requirements for each incoming task, by skill type, STT and cost per task. For one run of an algorithm 30 teams were created with 30 rounds of 10 task assignments. We ran a Pareto comparison and ranking method on the 30 teams and ranked them based on the three interaction-based objectives, namely the normalized, communication weights, coordination weights, as well as the diameter weight of the team. Table II gives the results of the 10 most appropriate teams returned by a run of Algorithm 1.

Analyzing the results we could observe that for example the most appropriate team with our algorithm does not necessarily have the highest trust score but it has a fairly high trust score and enough high scores for the communication and coordination values and low enough value for the function value. Looking at Table II, we notice that team with Id 21 has better scores when considering the three objectives, although it might have worse scores when considering single objectives one by one, compared to individual objectives of various other teams. Let us check two other interesting examples, the teams with Id 6 and 22 for example. Team 22 is better with regard to $W_m(C)$ and $W_c(C)$ when considered together, but on the other hand is worse regarding $D(C)$ than the team with Id 6, that is why it is ranked much lower. Thus, selecting the highest ranked team returned from Algorithm 1 seems a valid option, considering every pre-set requirement. Perhaps it is not the best solution considering single requirements, as for example team with Id 22 might be better in terms of trust and cost (not shown in the results for the sake of clarity) but not better than team with Id 21 in terms of the overall requirements considered in combination.

[1]We based our model on the framework presented by authors in [8].

TABLE II

Ranked teams and values of three objectives

Team ID	Trust	$W_m(C)$	$W_c(C)$	$D(C)$
21	0.8	1	0.6	0.4
6	0.64	0.8	0.6	0.3
23	0.75	0.6	0.6	0.1
19	0.52	0.5	0.2	0.2
8	0.65	0.5	0.4	0.3
22	0.82	0.5	0.8	0.8
27	0.6	0.3	1	0.5
3	0.65	0.4	0.2	0.6
17	0.62	0.2	0.6	0.8
7	0.56	0.1	0.5	0.5

Algorithm 2 takes as an input 10 team configurations ranked according to higher communication and coordination weights and lower team diameter. Table III shows the teams at the beginning of the algorithm, and the values for our three objectives, while Table IV shows the last returned teams as they were changed/generated in each run for new generation of teams with the algorithm. The results show that up to the point of the pre-set number for new team generations, the communication weight got to the value of 1 for all generated teams, while the communication weight value did so for only a few teams, this is due to our configurations and the pre-set values for team members regarding these values.

From the perspective of the comparison of both algorithms, we can conclude that in the case we want to form new teams from existing ones, which have been invoked with similar project requirements and with the same number of team members, a genetic algorithm approach that considers both high trusted teams, high communication and coordination interaction weights, and low team distance measure returns better teams than a non-evolutionary algorithm (Algorithm 1) forming teams from a pool of experts not considering previous individual membership in teams. However, Algorithm 1 returns efficient enough teams when considering individuals from a large pool instead of individuals from already existing teams. Needless to say, Algorithm 2 can be run on existing team-network structures in the case that such logs exist, and only on cases where team-structures include all the skill types required.

B. Evaluation with real data

In addition to our synthetically generated data-set we investigated how Algorithm 1 behaved with a real data-set

TABLE III

RANKED TEAMS AND VALUES OF THREE OBJECTIVES AS INPUT FOR ALGORITHM 2

Team ID	$W_m(C)$	$W_c(C)$	$D(C)$
12	1	0.4	0.1
13	0.8	0.8	0.5
2	1	0.5	0.5
28	0.8	0.8	0.6
20	0.2	0.5	0.2
27	0.6	0.4	0.8
14	0.1	0.4	0.2
17	0.1	0.2	0.8
25	0.3	0.2	0.1
6	0.3	0.2	0.2

TABLE IV

RANKED TEAMS AND VALUES OF THREE OBJECTIVES AT A FINAL RUN OF TEAM GENERATIONS IN ALGORITHM 2

Team ID	$W_m(C)$	$W_c(C)$	$D(C)$
4	1	1	0.2
2	1	1	0.2
8	1	1	0.4
14	1	0.8	0.1
5	1	0.6	0.1
24	1	0.5	0.4
27	1	0.5	0.1
11	1	0.5	0.3
3	1	0.2	0.4
1	1	0.1	0.1

for comparison. We utilized a data-set created by authors in [9][2], which provides real data for the activities of software engineering student teams for a final project in a software engineering course. The data-set provides a variety of data regarding 74 teams working on projects with the same requirements, collected during several semesters at the San Francisko State University. Each team within the data-set is assigned two different grades, one for the development process, and another for the final software built. The grades assigned are A, and F, A representing good results or above expectations and F representing below expectation teams.

The difference between our generated data and the real data-set is that with the synthetic data we generated expert profiles and formed teams according to their rank in the context of their trust score, and then ranked the teams optimizing three objectives, while the real data-set does not provide information regarding team member skills and competencies, rather it provides data on a team-level. Thus, we evaluate the ranking part of Algorithm 1 based on the three objectives only: communication score for the team, coordination score and distance between team members. Moreover, we did not form teams by ranking experts but ranked teams from the data-set by mapping appropriate team related data that fitted our model.

Mapping the data from the data-set, we selected the following team indicators that fitted most to our objectives: *meetingHours*, from the data-set gives the number of team meeting hours which we associated with *communication weight* value in our model to indicate the communication intensity of teams, and *teamMemberResponseCount* and *leadAdminHoursResponseCount*, which are self-reporting team-member and team-lead reports collected multiple times during the development process, which we associated with our *coordination weight* value in our model. The sum of *teamMemberResponseCount* and *leadAdminHoursResponseCount* for each team represents a teams coordination weight value.

The teams in the data-set are of two types, local (from the same university) and global (composed of members from multiple universities). Thus, mapping this data to our model, we assign $D(C) = 1$ to the local teams, and $D(C) = 0.5$ to global teams, with the assumption that members in the local teams know each other better than those in the global teams, and thus we assign a hard-coded distance weight value to each team based on this assumption. Table V shows the returned list of the first ranked 13 teams. After examining the ranked teams we noticed that most of the teams had A score for both the development process and the product delivered, but some had A either for the development process or the product delivered. However, the global teams which were formed from various universities had less meetings than those that could meet online, but some of these global teams got grade A even with lower communication interactions. If we take the grade A as a trust indicator, the results show that communication and coordination interactions should be considered together with a trust score for team formation algorithms to be effective. Consequently, trust plays an important role in two contexts when considering communication, coordination and network distance. On one side it can be used as an additional indicator to clarify cases where the communication link number is low, because if the trust between two experts is high, low communication does not mean bad communication. On the other hand, trust can be an indicator of communication link type such that if we want to denote communication links with positive and negative signs denoting positive and negative communication between two experts then understandably, a trust score can be used in decision-making scenarios for the sign of communication links. We leave this problem for our future research.

V. A FUZZY APPROACH TO THE TRUST AND INTERACTION-TYPES FORMATION MODEL: DISCUSSION

The algorithms proposed work with people who are available to work on a required project/goal. In the case when there is uncertainty regarding the availability of workers, and we want to consider it, a fuzzy-set ([10]) based approach can also be utilized to form teams considering their availability in addition to the multi-objective functions. Hence, to make the pool of resources even more concrete for Algorithm 1, a fuzzy based approach would allow us to consider a pool of resources with high trust and high availability for the required project. With the fuzzy approach trust indicator constraints can be more relaxed as well. If we define f as a variable

[2] We found the data-set from: http://archive.ics.uci.edu/ml/datasets/Data+for+Software+Engineering+Teamwork+Assessment+in+Education+Setting.

TABLE V
RANKED TEAMS AND VALUES OF THREE OBJECTIVES FROM A REAL-WORLD DATA-SET

Team ID	$W_m(C)$	$W_c(C)$	$D(C)$	Type
2	176.57	49.0	0.5	local
4	145.57	49.0	0.5	local
1	102.92	48.0	0.5	local
5	219.0	46.0	0.5	local
0	94.95	41.0	0.5	local
3	78.36	43.0	0.5	local
6	56.32	41.0	0.5	local
8	138.86	38.0	0.5	local
12	139.86	34.0	1.0	global
9	54.29	31.0	0.5	local
7	52.86	23.00.1	1.0	global
10	47.71	35.0	1.0	global
11	52.43	27.0	1.0	global

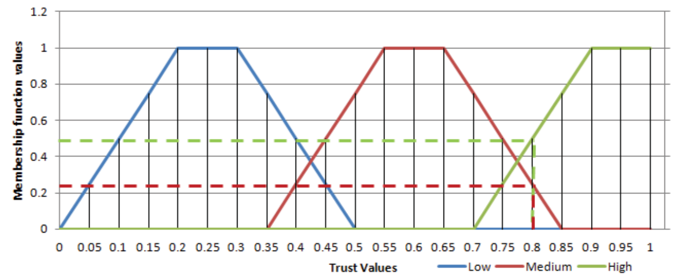
with the label "fitness", then we would have a set with terms $F = \{high, medium, low\}$ indicating the level of fitness of an expert into the collective based on the membership functions for its trust and availability. The membership functions can be defined as trapezoid ones such as:

$$\mu_X(x) = \begin{cases} 0 & if \quad (x \le a) \bigvee (x > d); \\ (x-a)/(b-a) & if \quad a < x \le b; \\ (d-x)/(d-c) & if \quad c \le x < d; \\ 1 & if \quad b \le x \le c. \end{cases} \tag{3}$$

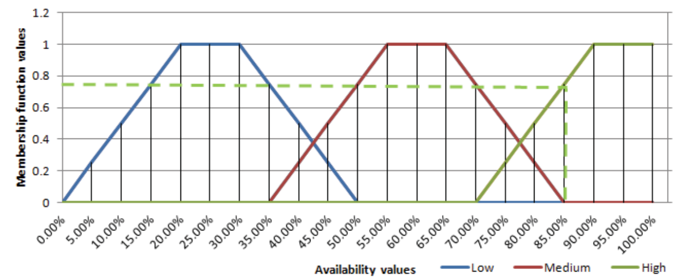
For low, medium, and high fitness a, b, c and d are pre-set values. Figure 2 shows an example of the membership functions, and specific values of an expert: a trust score of an expert 0.8, and availability 85%. If we denote t as a variable indicating trust and v indicating availability, some rule examples that can be utilized in this example are:

- if t is high & v is high then f is high
- if t is high & v is medium then f is medium
- if t is medium & v is medium then f is medium
- if t is low & v is low then f is low

In Algorithm 1 we can apply the fuzzy approach step for the ranked list of experts after Line 5. Then we can use the results of the first rule which would give experts with high fitness, as input to the algorithm in Line 6, instead of the larger list of resources returned by AHP used in Line 6. Of course, availability can be utilized with AHP as well together with other metrics, but the purpose of this section is to discuss possible approaches. The detailed description of fuzzification and defuzzification methods is out of the scope of this paper, as these depend on the client and the developer of the algorithm as well (a Mamdani approach can be used for the following steps for example). The purpose of this section is to illustrate a case where a fuzzy approach could be used to specify a group of people from a large pool of resources, and use the list as input data for forming multi-objective teams as in Algorithm 1.



(a) Particular values for *trust* and membership functions



(b) Particular values for *availability* and membership functions

Fig. 2. Selected *trust* and *availability* values and their corresponding values within the defined membership functions

VI. RELATED WORK

The authors in [6] have presented two heuristics based on genetic algorithms and simulated annealing for team-formations that consider skills and connectivity of teams members. They also present and discuss a recommendation model for adding new members to the team to fulfill skill requirements. Interactions are also considered in [11] where authors present multiple strategies for team-assembly focusing on multiple aspects of cost, they present team-assembly strategies considering the cost of communication, a strategy to find a team based on the cost of team-members, as well as finding Pareto-optimal teams considering both the communication cost and the team-member cost. Lappas et al. in [12] also present Pareto-optimal team formation algorithms with minimized communication cost. Anagnostopoulos et al. in [13] discuss forming teams considering the trade-offs between team-size and load with the help of a greedy task-assignment algorithm, while in [14] they provide team-formation algorithms that consider coordination cost and workload balancing. Examples of work that in general consider the underlying social network from which teams are formed are [15], [16].

Of course if we allow for elastic teams once the team is formed we can adapt the teams at run-time, similar to our approach [7], and avoid considering the trade-off between team-size and cost at the time of the formation of the team as is the concern of authors in [17].

Authors in [18] discuss team formation as well as network adaptations based on two different approaches, namely, structure-based and performance-based strategies. Authors in [19] present algorithms for team formation based on skill and coordination cost in two different situations, when a team does not have a leader and when a team has a leader responsible for the team coordination, and run experiments on the DBLP dataset, concluding that the algorithms can form small teams with a high number of common publications and high expertise. An AHP-based approach of a multi-objective optimization is presented in [20].

[21] have discussed fuzzy based approaches for assessing trust in service-oriented computing where people are treated as computational resources. They argue that people should be able to define trust as they see fit to the domain of application, and argument the fuzzy approach to trust. We make the same argument for availability, as people who work online can set their own schedules, but also requesters of work/clients, should be able to define what is meant under *availability* in their own application domain. Availability can be also specified for the type of skills that are required for various steps of a project with regard to time. For example, an expert might be available as a developer 60% of his/her total working time, while for the designing skills he/she might be available only 40%.

[3] presents an overview of distributed intelligence where human teams are mentioned as entities in distributed intelligence systems. Another mentioned fact is that organizational and social paradigms are a starting point for designing agent systems that can cooperate and collaborate for a common complex objective. Hence, we postulate that learning the formation of efficient human-based collectives could be valuable for agent and mixed-system resource-assembly problems. Distributed intelligence including human computation in various application areas are discussed in [4]. The authors present the term 'global brain' and elaborate on the benefits of distributed intelligence, such as for example disaster prevention and relief, research, innovation, traffic management and others. Considering the team formation problem in this work, the benefits of it can be seen in many areas where distributed expert teams need to be formed for a specific objective. The importance of social interactions in human computation is emphasized in [22], where the authors make parallels between neural networks and human computation ones.

VII. Conclusion and Future Work

In this work we argued the importance of differentiating interaction types between experts and the way these can be included in team-formation mechanisms. We assessed team efficiency by a trust score and conducted experiments with the assumption that more communication and coordination links means more efficient teams. However, in cases where we do not have a trust metric, a high number of the communication and coordination links can also result in non-efficient teams, because communication links can be negative. We plan to approach this problem in future work. In addition, in future work we will work on extending this model with the assumption that experts provide multiple skills, when a single worker can be included in a team with various roles to work for various task types, and investigate how our model and algorithms perform in those cases. In addition, we plan to investigate forming a set of multi-domain teams where multiple teams are formed to work together in a complex system that requires multiple areas of general domain-expertise for separate teams, in this way engaging in an investigation of a larger-scale distributed computing challenge. Another open research challenge we are interested to address in the future concerns mechanisms of management for multiple-teams working together with software services in mixed-resource systems of distributed intelligence.

References

[1] A. Bozzon, P. Fraternali, L. Galli, and R. Karam, "Modeling crowdsourcing scenarios in socially-enabled human computation applications," *Journal on Data Semantics*, vol. 3, no. 3, pp. 169–188, Sep 2014. [Online]. Available: https://doi.org/10.1007/s13740-013-0032-2

[2] G. Fischer, "Distributed intelligence: Extending the power of the unaided, individual human mind," in *Proceedings of the Working Conference on Advanced Visual Interfaces*, ser. AVI '06. New York, NY, USA: ACM, 2006, pp. 7–14. [Online]. Available: http://doi.acm.org/10.1145/1133265.1133268

[3] L. E. Parker, "Distributed intelligence: Overview of the field and its application in multi-robot systems," *Journal of Physical Agents*, pp. 5–14, 2008.

[4] F. Heylighen, *From Human Computation to the Global Brain: The Self-Organization of Distributed Intelligence*. New York, NY: Springer New York, 2013, pp. 897–909. [Online]. Available: https://doi.org/10.1007/978-1-4614-8806-4_73

[5] S. Dustdar and H. L. Truong, "Virtualizing software and humans for elastic processes in multiple clouds- a service management perspective," *IJNGC*, vol. 3, no. 2, 2012.

[6] C. Dorn, F. Skopik, D. Schall, and S. Dustdar, "Interaction mining and skill-dependent recommendations for multi-objective team composition," *Data Knowl. Eng.*, vol. 70, no. 10, pp. 866–891, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.datak.2011.06.004

[7] M. Riveni, H. L. Truong, and S. Dustdar, "Trust-aware elastic social compute units," in *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*, 2015, pp. 135–142. [Online]. Available: http://dx.doi.org/10.1109/Trustcom.2015.367

[8] V. Casola, A. R. Fasolino, N. Mazzocca, and P. Tramontana, "An ahp-based framework for quality and security evaluation," in *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 03*, ser. CSE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 405–411. [Online]. Available: http://dx.doi.org/10.1109/CSE.2009.391

[9] D. Petkovic, M. Sosnick-Pérez, K. Okada, R. Todtenhoefer, S. Huang, N. Miglani, and A. Vigil, "Using the random forest classifier to assess and predict student learning of software engineering teamwork," in *2016 IEEE Frontiers in Education Conference, FIE 2015, Eire, PA, USA, October 12-15, 2016*, 2016, pp. 1–7. [Online]. Available: https://doi.org/10.1109/FIE.2016.7757406

[10] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965. [Online]. Available: https://doi.org/10.1016/S0019-9958(65)90241-X

[11] 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.*, 2013, pp. 587–595. [Online]. Available: https://doi.org/10.1137/1.9781611972832.65

[12] 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*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 467–476. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557074

[13] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Power in unity: forming teams in large-scale community systems," in *CIKM*, 2010, pp. 599–608.

[14] ——, "Online team formation in social networks," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12. New York, NY, USA: ACM, 2012, pp. 839–848. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187950

[15] C. Li, M. Shan, and S. Lin, "On team formation with expertise query in collaborative social networks," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 441–463, 2015. [Online]. Available: https://doi.org/10.1007/s10115-013-0695-x

[16] M. Chhabra, S. Das, and B. Szymanski, "Team formation in social networks," in *Computer and Information Sciences III*, E. Gelenbe and R. Lent, Eds. London: Springer London, 2013, pp. 291–299.

[17] I. Kamel, Z. A. Aghbari, and K. Kamel, "Smartrecruiter: a similarity-based team formation algorithm," *IJBDI*, vol. 3, no. 4, pp. 228–238, 2016. [Online]. Available: https://doi.org/10.1504/IJBDI.2016.10000808

[18] M. E. 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*, ser. AAMAS '05. New York, NY, USA: ACM, 2005, pp. 230–237. [Online]. Available: http://doi.acm.org/10.1145/1082473.1082508

[19] M. Kargar and A. An, "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*, ser. CIKM '11. New York, NY, USA: ACM, 2011, pp. 985–994. [Online]. Available: http://doi.acm.org/10.1145/2063576.2063718

[20] A. Mazza, G. Chicco, and M. Rubino, "Multi-objective distribution system optimization assisted by analytic hierarchy process," in *2012 IEEE International Energy Conference and Exhibition (ENERGYCON)*, Sept 2012, pp. 393–400.

[21] F. Skopik, D. Schall, and S. Dustdar, "Modeling and mining of dynamic trust in complex service-oriented systems," *Inf. Syst.*, vol. 35, no. 7, pp. 735–757, Nov. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.is.2010.03.001

[22] L. J. Larson-Prior, *Parallels in Neural and Human Communication Networks*. New York, NY: Springer New York, 2013, pp. 39–49. [Online]. Available: https://doi.org/10.1007/978-1-4614-8806-4_5