A Model for Estimating the Performance of a Team of Agents

Farzad Kamrani, Rassul Ayani
KTH Royal Institute of Technology
Stockholm, Sweden
Emails: last-name@kth.se

Farshad Moradi
Swedish Defense Research Agency (FOI)
Stockholm, Sweden
E-mail: farshad.moradi@foi.se

***Abstract*—In this paper, we present a model for estimating the performance of a team of agents, based on the capabilities of the agents and importance of these capabilities for the task. Performance of a team is assumed to be the sum of contributions of individual agents and contributions of subgroups built in the team. We introduce a set of notations, which is required for discussing the suggested models. We also propose a model to estimate the benefit of an agent from interaction with other agents in a subgroup. Based on this benefit model and different (common) strategies, the agents devise plans in which they formulate to what extent they are willing to cooperate with other agents. A negotiation algorithm that resolves the conflicts between the desires of the agents is presented. The effect of this algorithm and different strategies are tested on a set of generated data. The test results show that the performance of a team when the agents choose a cooperation strategy that follows the principle of least effort (Zipf's law) is higher than teams with other cooperation strategies.**

I. Introduction

In an earlier work [1], we presented a model for estimating the performance of agents in a business process model. This model was used in [2] for optimizing the performance of a business process model by assigning tasks to agents. However, in both these works it was assumed that each task is assigned to only one agent and we did not consider the cases where multiple agents perform a task and the impact of agents' interactions. The first step toward studying the performance of teams in a business process model is to provide a model for value added by agents that work cooperatively and interact with each other. This paper presents a method to model the performance of agents that cooperate as a team.

Teams are ubiquitous in organizations today and it is commonly accepted that the performance of a team is higher than the sum of the performance of individual team members. The positive effect of teamwork on the performance of organizations is discussed e.g. in [3], [4], [5]. A substantial amount of research has been devoted to the analysis of the performance of teams in different fields such as education, health care, science, military, performing arts and manufacturing [6], [7], [8], [9], [10], [11], [12].

In this work the aim has been to provide a general framework to model and measure the performance of agents in a teamwork context. Since it is generally difficult to validate agent models, the framework is decomposed to submodels for individual agents, group of agents and a model for interaction of agents. Although this framework is accompanied by agent models and group models, these models can easily be modified or substituted by more accurate models with higher fidelity.

The outline of the rest of this paper is as follows. In Section II, we define the problem and provide a notation that is used throughout this paper. In Section III, a model for the performance of individual agents is presented. In Section IV, a model for how interaction between agents affects their performance is suggested. In Section V, different cooperation strategies of agents are discussed. Section VI introduces a negotiation algorithm, which is used to resolve the conflicts between agents. In Section VII, results of a series of tests for different agent strategies and the negotiation algorithm are presented. Section VIII concludes the paper.

II. Problem formulation

We define a *team* as a set of 2 or more agents $\{a_1, \dots, a_m\}$, that work individually or in cooperation with other agents to perform a task and denote the team by $G_{\mathcal{A}}$, where the index $\mathcal{A}$ is the set of agent indexes. That is $\mathcal{A} = \{1, \dots, m\}$, $|\mathcal{A}| \geq 2$ and $i \in A \Leftrightarrow a_i \in G_{\mathcal{A}}$.

During a teamwork process different interactions between agents occur and different constellations of agents are built. We call a constellation of 2 or more agents a *subgroup* and denote it by $G_{\mathcal{S}}$, where the index $\mathcal{S}$ is the set of the indexes of agents that participate in the subgroup, that is $\mathcal{S} \subseteq \mathcal{A}$, $|\mathcal{S}| \geq 2$, $i \in \mathcal{S} \Leftrightarrow a_i \in G_{\mathcal{S}}$. For instance in a team of 5 agents $\{a_1, \dots, a_5\}$, subgroup $G_{\{1,2,4\}}$ consists of the three agents a_1, a_2 and a_4, or if $G_{\mathcal{S}}$ is a subgroup of three agents, then $G_{\mathcal{S} \cup \{a_i\}}$ is a subgroup that contains agents in $G_{\mathcal{S}}$ and a_i.

The set of all possible subgroups in a team is denoted by $\mathbb{G} = \{G_{\mathcal{S}}, \mathcal{S} \subseteq \mathcal{A}, |\mathcal{S}| \geq 2\}$. In set theory, the *power set* of a set $\mathcal{A}$, (denoted by $2^{\mathcal{A}}$) is defined as the set of all subsets of $\mathcal{A}$. For a set with m members, the cardinality of its power set is 2^m. Therefore, in a team of m agents the number of all possible interacting subgroups is $|\mathbb{G}| = 2^m - m - 1$, (sets with only one member and the empty set are excluded).

If t_c is the total time required to complete a task and $t_{\mathcal{S}}$ is the time that subgroup $G_{\mathcal{S}}$ is active, we denote the quotient $t_{\mathcal{S}}/t_c$ as $\alpha_{\mathcal{S}}$. For example, if the subgroup $G_{\{1,2,4\}}$ is active 2 days in a task that lasts 10 days, then $\alpha_{\{1,2,4\}} = 0.2$, or if $G_{\{1,2,3,4,5\}}$ is a team meeting consisting of all members and $\alpha_{\{1,2,3,4,5\}} = 0.1$, it indicates that 10% of the time all 5 agents are working together.

In addition, we use $\alpha_{i,i}$ to show the proportion of time that agent a_i is working alone and $\alpha_{i,0}$ for the case when agent a_i is not involved in current activity. It is obvious that for each agents a_i the sum of all its time proportion is 1.0. That is, $\forall a_i \in \{a_1, \ldots, a_m\}$, we have

$$\alpha_{i,0} + \alpha_{i,i} + \sum_{\substack{\mathcal{S} \subseteq \mathcal{A} \setminus \{i\} \\ |S| \geq 1}} \alpha_{\mathcal{S} \cup \{i\}} = 1.0 \tag{1}$$

For a team of m agents $G_{\mathcal{A}}$, we define the *time schedule* of the team as a set of real numbers $T = \{\alpha_{i,0}, \alpha_{i,i}, \alpha_{\mathcal{S} \cup \{i\}} : i \in \mathcal{A}, \mathcal{S} \subseteq \mathcal{A} \setminus \{i\}, |\mathcal{S}| \geq 1\}$, where all $\alpha_{i,0}, \alpha_{i,i}$ and $\alpha_{\mathcal{S} \cup \{i\}}$ fulfill the constraint expressed by (1). Given a team $G_{\mathcal{A}}$ and its time schedule T, the overall structure of the team is settled, i.e. it will be clear that during the teamwork process which subgroups are established and how much time each agent is active in each subgroup.

Inspired by [13], [14] we assume that the value added by agents consists of two contributions,

- Individual contributions,
- Subgroup contributions,

that is

$$u(total) = u(individual) + u(subgroups), \tag{2}$$

where all contributions are expressed as value added to the process by agents in a time unit. For a given team and a predetermined time schedule, the value produced by the team in a time unit is calculated according to

$$u(total) = \sum_{i=1}^{m} \alpha_{i,i} v(a_i) + \sum_{\substack{\mathcal{S} \subseteq \mathcal{A} \\ |S| \geq 2}} \alpha_{\mathcal{S}} v(G_{\mathcal{S}}), \tag{3}$$

where $v(a_i)$ is the value produced by the individual agent a_i in a time unit, and $v(G_{\mathcal{S}})$ is the value produced in a time unit by subgroup $G_{\mathcal{S}}$, and values for α are defined as above. We make the assumption that, the value produced by a subgroup may be expressed as the sum of values produced by agents while they are influenced by the rest of subgroup members,

$$v(G_{\mathcal{S}}) = \sum_{i \in \mathcal{S}} v'(a_i, G_{\mathcal{S} \setminus \{i\}}). \tag{4}$$

Therefore, we call the function $v'(a, G_{\mathcal{S}})$, *influence function*.

Substituting (4) in (3) yields

$$u = \sum_{i=1}^{m} \alpha_{i,i} v(a_i) + \sum_{\substack{\mathcal{S} \subseteq \mathcal{A} \\ |S| \geq 2}} \sum_{i \in \mathcal{S}} \alpha_{\mathcal{S}} v'(a_i, G_{\mathcal{S} \setminus \{i\}}), \tag{5}$$

which is equivalent to

$$u = \sum_{i=1}^{m} (\alpha_{i,i} v(a_i) + \sum_{\substack{\mathcal{S} \subseteq \mathcal{A} \setminus \{i\} \\ |\mathcal{S}| \geq 1}} \alpha_{\{i\} \cup \mathcal{S}} v'(a_i, G_{\mathcal{S}})). \tag{6}$$

Equation (6) sums all agents' individual contribution and their contribution while they are influenced by other agents in subgroups. All contributions are weighted by their corresponding time proportion factor α. It is worth to note that the influence function depends not only on the agent a_i and other agents $G_{\mathcal{S}}$ but also on the *task* they are performing. If the agents, the task, the time schedule and the influence function are known, then calculating the value produced by the team would be straightforward, by using (6). However, there are two major difficulties to be overcome in order to accomplish this, which this paper aims to discuss and present a solution to.

1) The nature of the influence function and how agents influence each other in teamwork is one major difficulty in estimating the value produced by a team. Depending on the type of the task and the required capabilities a wide range of interaction forms may merge resulting in different functions.
2) In most situations teams exhibit some degree of self-management and the team schedule is not completely predetermined. Agents are more or less free to choose other agents that they want to cooperate with. Different factors may affect the choice of agents and the degree of cooperation with other agents. However, we assume that agents prefer cooperation that is most beneficial to them, i.e. increases their performance. The benefits of agents from the cooperation are directly affected by the influence function. This assumption leads to the following three related issues, which are the focus of this paper:
   a) different strategies that agents may choose for cooperation,
   b) how the conflicting choices of agents are resolved,
   c) the effect of the chosen strategy on the final outcome of the value produced by the team.

III. A MODEL FOR MEASURING INDIVIDUAL CONTRIBUTIONS

In [1], we presented a model for estimating the performance of human agents in a business process. In this model the value added by an agent to a task (per time unit) is described as a function of the *capabilities* of the agent and the importance or *weights* of these capabilities for the task. Capabilities of agents and their importance for the task are assumed to be either subjective assessments that are assigned by domain experts and decision makers to the agents and tasks, or measurable quantities. For instance, in a software development process various programming skills may be examples of capabilities of agents, or in a military planning team, more subjective characteristics such as cognitive ability, creativity and communication may constitute the set of the required capabilities. Capabilities are graded by a number ranged between 0 and 5, where 0 is lack of the capability and 5 is the highest possible value.

The effect of these characteristics on the value added by the agents depends on the nature of the work, e.g. creativity may have a huge impact on a problem-solving task, while it is less important for work on a conveyor belt. Hence, the sum of agents' capabilities weighted by their importance for the task constitutes an indicator of the value added by agents to the task [1].

Considering m agents each having n attributes, a *capability* matrix $\mathcal{C} = [c_{ij}]_{m\times n}$ is defined, where c_{ij} is the attribute j of agent i. In the same way, a *weight* vector $\mathcal{W} = [w_j]_{n\times 1}$ is defined for the task, where w_j is the weight of attribute j. The weighted sum of attributes, $v(a_i)$ is the value added by agent a_i to the task (per time unit), which is estimated by

$$v(a_i) = \sum_{j=1}^{n} c_{ij} w_j. \tag{7}$$

Thus, the sum of values added by individual agents (per time unit) is calculated by

$$u(individual) = \sum_{i=1}^{m} \alpha_{i,i} v(a_i) = \sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_{i,i} c_{ij} w_j. \tag{8}$$

IV. A MODEL FOR MEASURING SUBGROUP CONTRIBUTIONS

In Section II we proposed,

$$u(subgroups) = \sum_{i=1}^{m} \sum_{\substack{\mathcal{S}\subseteq\mathcal{A}\setminus\{i\}\\ |\mathcal{S}|\geq 1}} \alpha_{\{i\}\cup\mathcal{S}} v'(a_i, G_{\mathcal{S}}). \tag{9}$$

That is, the total contribution of subgroups per time unit is the sum of contributions of agents (per time unit), while they are influenced by other agents in subgroups multiplied by the proportion of time that the agent is involved in the subgroup activity.

The assumption is that agents perform better since their capabilities are increased as a result of cooperation with other agents with a higher capability. For each agent $a_i \in G_{\mathcal{S}}$ with n capabilities $\{c_{i1}, \ldots, c_{in}\}$, each capability c_{ij} is increased by a value, which we call *benefit function* $b(c_{ij}, \{c_{kj}\}_{k\in\mathcal{S},k\neq i})$. Thus, the new capability c'_{ij} is calculated by

$$c'_{ij} = c_{ij} + b(c_{ij}, \{c_{kj}\}_{k\in\mathcal{S},k\neq i}). \tag{10}$$

In words, the benefit of an agent from cooperation (with respect to each capability) is a function of its own and other agents' capability.

We consider some simple rules to deduce an equation for the benefit function.

1) Agents help each other. In a subgroup of two or more agents, agents that have a lower capability than the *maximum capability* in the subgroup will benefit from the cooperation. The maximum capability is defined as $c_j^{max} = \max\limits_{i\in\mathcal{S}}\{c_{ij}\}$. Hence, it is justified to rewrite $b(c_{ij}, \{c_{kj}\}_{k\in\mathcal{S},k\neq i})$, as $b(c_{ij}, c_j^{max})$.
2) Agents do not disturb each other. The capability of an agent who has the maximum capability is not affected by the cooperation.
3) Capabilities are enhanced, they are not created. The benefit of an agent with capability 0 is always 0, with respect to that capability.
4) Equally good agents cannot help each other. The benefit of agents with the same capability is always 0, with respect to that capability.

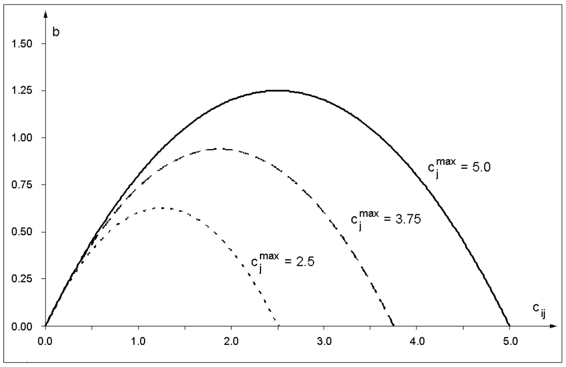

Fig. 1. The benefit function, i.e. the increase of capability j of agent a_i, while cooperating with an agent with a higher capability c_j^{max}, for 3 different values of c_j^{max}.

5) For each capability, the benefit function has its only maximum for a value $c^*_{ij} \in (0, c_j^{max})$. For simplicity we assume $c^*_{ij} = c_j^{max}/2$, as illustrated in Fig. 1.
6) It is always more beneficial for an agent to be the member of a subgroup that has higher capabilities. For two groups $G_{\mathcal{S}}$ and $G_{\mathcal{S}'}$ with c_j^{max} and c'^{max}_j respectively, $c_j^{max} > c'^{max}_j \Rightarrow b(c_{ij}, c_j^{max}) \geq b(c_{ij}, c'^{max}_j)$ for all values of $c_{ij} \in [0, c'^{max}_j]$.

These criteria imply that for a constant value of c_j^{max}, the benefit function $b(c_{ij}, c_j^{max})$ is a concave function over $c_{ij} \in [0, c_j^{max}]$ with $b(0, c_j^{max}) = 0$ and $b(c_j^{max}, c_j^{max}) = 0$. Furthermore, for a constant value of c_{ij}, $b(c_{ij}, c_j^{max})$ is an increasing function on $c_j^{max} \in [0, 5]$. We suggest a simple function that fulfills the above requirements,

$$b(c_{ij}, c_j^{max}) = c_{ij}(c_j^{max} - c_{ij})/c_j^{max}. \tag{11}$$

This function is shown in Fig. 1 for three different values of c_j^{max}, i.e. $= 2.5, 3.75$ and 5.0.

A consequence of the definition of the benefit function is that in a subgroup of two agents a_1 and a_2, we have

$$b(c_{1j}, c_{2j}) = \begin{cases} 0, \quad c_{1j} > c_{2j} \\ c_{1j}(c_{2j} - c_{1j})/c_{2j} \quad otherwise, \end{cases} \tag{12}$$

which means that for each capability the benefit of one of the agents from the cooperation is always 0. However, each agent may be better in different capabilities, which will make the cooperation mutually beneficial.

We use (10) and (11) to obtain the modified capabilities of agent a_i and calculate its contribution, while it is influenced by other agents in subgroup $G_{\mathcal{S}}$,

$$v'(a_i, G_{\mathcal{S}}) = \sum_{j=1}^{n} (c_{ij} + c_{ij}(c_j^{max} - c_{ij})/c_j^{max}) w_j. \tag{13}$$

Substituting the value obtained by (13) in (9) yields

$$u(subgroups) = \sum_{i=1}^{m} \sum_{\substack{\mathcal{S}\subseteq\mathcal{A}\setminus\{i\}\\ |\mathcal{S}|\geq 1}} \alpha_{\{i\}\cup\mathcal{S}} \sum_{j=1}^{n} (c_{ij} + c_{ij}(c_j^{max} - c_{ij})/c_j^{max}) w_j. \tag{14}$$

In the following, we consider a special type of team in which the schedule of team meetings and groups including more than 2 agents are predetermined. In other words, we assume that the value of $\alpha_{\mathcal{S}}$ is predetermined, $\forall \mathcal{S} \subseteq \mathcal{A}, |\mathcal{S}| \geq 3$. However, agents are free to divide the rest of their time between working alone or cooperating with one other agent. We assume that this kind of cooperation is not regulated by the organization and is something that emerges naturally from the interaction among agents. That is, each agent a_i is free to decide all its own $\alpha_{i,i}$ and $\alpha_{\{i,j\}}$, $\forall j, j \neq i$ provided that this does not violate its time constraints,

$$\alpha_{i,i} + \sum_{\substack{j=1 \\ j \neq i}}^{m} \alpha_{\{i,j\}} = 1.0 - \alpha_{i,0} - \sum_{\substack{\mathcal{S} \subseteq \mathcal{A} \setminus \{i\} \\ |\mathcal{S}| \geq 2}} \alpha_{\{i\} \cup \mathcal{S}}. \qquad (15)$$

According to our assumptions, the right-hand side of (15) is the unscheduled time of agent a_i, which we denote by α_i.

For a team consisting of m agents, we let $\alpha_{i,i}$ and $\alpha_{\{i,j\}}$, define the *interaction matrix* $\alpha = [\alpha_{ij}]_{m \times m}$. If $i \neq j$, element α_{ij} of this matrix is the proportion of time agent a_i is working with agent a_j (i.e. $\alpha_{\{i,j\}}$), otherwise it is the time agent a_i is working without interference of other agents ($\alpha_{i,i}$). The sum of each row is the unscheduled time of each agent, and is equal to α_i.

The commutativity requirement implies that the matrix α is a symmetric matrix. If there is no cooperation between agents, matrix α would be equal to I_m.

We assume that each agent desires to maximize its own contribution in the team, by cooperating with agents that have a higher capability. However, agents have to respect the time constraints and are obliged to compromise with other agents, when they have different desires. That is, for the interaction matrix it is required that $\forall i, \ \sum_{j=1}^{m} \alpha_{ij} = \alpha_i$ and $\alpha_{ij} = \alpha_{ji}$.

V. Agent Interaction Model

Here, we presuppose that the agents' understandings of the environment are accurate and consistent with each other. That is they know their own and other agents capabilities and the weight of these capabilities for the task. Moreover, they know the benefit function and know how cooperation with other agents will affect their performance. We assume that each agent tries to maximize its own contribution to the team by distributing its unscheduled time between other agents, while it respects its own time constraint and is willing to compromise with other agents.

Considering the discussions in IV, the best choice for each agent would be to only cooperate with agents having the highest capabilities that are most important for the task. However, this strategy is implausible, since it requires that some agents are forced to cooperate with others against their intentions. A more realistic strategy is that agents choose a compromise-oriented approach, which through negotiation can lead to a more fair cooperation. In the following, we discuss four possible cooperation strategies and suggest a negotiation algorithm that resolves the conflicts among agents.

Each agent a_i uses the benefit function and the capabilities of itself and other agents to produce a row vector $[b_{ii'}]_{1 \times m}$, where $b_{ii'} = \sum_{j=1}^{n} b(c_{ij}, c_{i'j}) w_j$. These m row vectors together build a matrix $\mathcal{B} = [b_{ii'}]_{m \times m}$, where the element $b_{ii'}$ gives the benefit of cooperation with agent i' for agent i. This matrix is shared information that can be accessed by all agents. Agents use this information and by following a common strategy give an initial suggestion for their cooperation with other agents. The initial suggestions of agents go through a negotiation algorithm so that desired consistency is reached.

Here, we compare 4 different strategies.

1) Equal, where each agent distributes its available time equally between all other agents, regardless of the benefit of the cooperation.
2) Random, where each agent distributes its available time randomly (drawn from a uniform distribution) between different agents.
3) Proportional, where each agent distributes its available time in proportion to the benefits of cooperation with other agents.
4) Zipf, where each agent distributes its available time inversely proportional to the rank of the benefit of cooperation with other agents.

Equal and random strategies are naive and will be used as a benchmark against which the two other strategies are evaluated. The three first strategies are obvious from their definitions, the fourth strategy may require some clarification. Zipf's law or *principle of least effort* [15], is an empirical law that states given some large and structured set of texts in a *natural language*, the frequency of any word is inversely proportional to its rank in the frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word and three times as often as the third most frequent word, etc. Zipf's law has shown to be applicable to many types of data studied in physical and social sciences. Applying this principle to agents in a team, we assume that each agent ranks the others according to the benefit of cooperation and is willing to work with them inversely proportional to their ranks.

Algorithm 1 shows how these 4 different strategies are implemented. The input of the algorithm is the benefit matrix $\mathcal{B}$, the unscheduled time of agents $[\alpha_i]$, and the common strategy of agents. The algorithm returns a matrix, which we call *preference matrix*. This matrix is the desire of agents for cooperation.

In the first inner for-loop of the algorithm (lines $4 - 18$), depending on the given strategy, elements p_{ij} of the row $\mathcal{P}[i]$ are initialized. Lines 2 and 3 of the algorithm are required only for the Zipf strategy (lines $11 - 17$). In line 19 the sum of each row of the matrix $\mathcal{P}$ is calculated and the unscheduled time of each agent α_i is divided in proportion to the p_{ij} between other agents (line 22). If all elements of $\mathcal{P}[i]$ are zero, the time is equally divided between all agents (line 24).

To illustrate the results of Algorithm 1, assume we have a team of 5 agents, with 4 capabilities, having the following capability matrix,

$$C = \begin{pmatrix} 1.5 & 3.25 & 2.5 & 2.0 \\ 3.0 & 3.0 & 4.0 & 0.75 \\ 1.0 & 1.75 & 1.0 & 2.0 \\ 2.5 & 2.0 & 2.0 & 4.0 \\ 1.75 & 2.0 & 1.75 & 2.0 \end{pmatrix}.$$

These agents are performing a task with the weight vector $W = \big(0.25,\, 0.75,\, 1.0,\, 2.0\big)^{-1}$.

We use the given weight vector and the benefit function in (12) and calculate the benefit matrix $\mathcal{B} = [b_{ii'}]_{5\times5}$. Each element $b_{ii'}$ in the matrix is equal to $\sum_{j=1}^{4} b(c_{ij}, c_{i'j})w_j$.

The complete matrix $\mathcal{B}$ is

$$\mathcal{B} = \begin{pmatrix} 0 & 1.12 & 0 & 2.15 & 0.05 \\ 1.11 & 0 & 0.94 & 1.22 & 0.94 \\ 1.29 & 1.46 & 0 & 2.81 & 0.7 \\ 0.98 & 1.60 & 0 & 0 & 0 \\ 1.10 & 1.67 & 0 & 2.35 & 0 \end{pmatrix}.$$

We assume that all $\alpha_i = 1.0$, which means agents have no predetermined schedule. By using Algorithm 1, the preference matrix for the proportional and Zipf strategies are calculated. For the proportional strategy, it means only that the matrix $\mathcal{B}$ is normalized such that the sum of the elements in each row is equals to $\alpha_i = 1.0$. This operation yields the following matrix

$$\mathcal{P}_p = \begin{pmatrix} 0 & 0.34 & 0 & 0.64 & 0.02 \\ 0.27 & 0 & 0.22 & 0.29 & 0.22 \\ 0.21 & 0.23 & 0 & 0.45 & 0.11 \\ 0.38 & 0.62 & 0 & 0 & 0 \\ 0.21 & 0.33 & 0 & 0.46 & 0 \end{pmatrix}.$$

By using Zipf's law and ranking the elements of the matrix $\mathcal{B}$, one can assign the inverse of the rank of each element to it and produce the matrix

$$\begin{pmatrix} 0 & 0.5 & 0 & 1 & 0.33 \\ 0.5 & 0 & 0.33 & 1 & 0.25 \\ 0.33 & 0.5 & 0 & 1 & 0.25 \\ 0.5 & 1 & 0 & 0 & 0 \\ 0.33 & 0.5 & 0 & 1 & 0 \end{pmatrix}.$$

The last part of Algorithm 1, (lines $19-26$) normalizes this matrix and yields

$$\mathcal{P}_z = \begin{pmatrix} 0 & 0.27 & 0 & 0.55 & 0.18 \\ 0.24 & 0 & 0.16 & 0.48 & 0.12 \\ 0.16 & 0.24 & 0 & 0.48 & 0.12 \\ 0.33 & 0.67 & 0 & 0 & 0 \\ 0.18 & 0.27 & 0 & 0.55 & 0 \end{pmatrix}.$$

A comparison of the two matrices $\mathcal{P}_p$ and $\mathcal{P}_z$ shows that elements of these two matrices are both zero in the same positions and they follow the same trend. Even if the calculation of $\mathcal{P}_z$ is more complicated in Algorithm 1, this strategy appears to be more plausible since agents do not require to calculate exactly the benefits of cooperation with other agents. To produce $\mathcal{P}_z$, agents only need to have a rough estimation of other agents and rank them according to their capabilities. Bounded rational agents, which have limited computational resources, are considered to use simplified models and approximate solutions instead of seeking the optimal one [16].

VI. Negotiation of agents

Algorithm 1 ensures that the sums of the rows of the calculated $\mathcal{P}$ matrices are equal to the agents' available time α_i. However, the sums of the columns are not equal to these values. A value greater than α_i for a column indicates that other agents request cooperativeness from the corresponding agent more than it is capable to provide, that is the agent is overstressed. A value less than α_i shows that the corresponding agent is understressed. A necessary condition for agents to be fairly stressed is that the sum of each column is equal to α_i as well. However, this condition is not sufficient and it is required that all agents in all subgroups reach a consensus on the time they are cooperating. Here, there are two agents in each subgroup and it requires that $\alpha_{ij} = \alpha_{ji}$, $\forall i, j$, i.e. $\mathcal{P}$ must be a *symmetric matrix*.

A special case of the problem is when $\alpha_i = 1.0$, for all i. That is the matrix is *stochastic*. A necessary (not sufficient) requirement for a stochastic matrix to be symmetric is that the matrix is *doubly stochastic*. A doubly stochastic matrix (also called bistochastic), is defined to be a square matrix of non-negative real numbers such that each row and each column sums to one [17].

Algorithm 1: Devising Preference Matrix

```
given   benefit matrix: $\mathcal{B} = [b_{ij}]_{m\times m}$
        strategy used by agents: strategy
        agents unscheduled time: $[\alpha_i]_{m\times 1}$
return  preference matrix: $\mathcal{P} = [p_{ij}]_{m\times m}$

1   for $i \in [1, \dots m]$
2       $tmp \leftarrow \mathcal{B}[i]$
3       $descending\_sort(tmp)$
4       for $j \in [1, \dots m]$
5           if strategy = equal
6               $p_{ij} \leftarrow 1.0$
7           else if strategy = random
8               $p_{ij} \leftarrow rnd$
9           else if strategy = proportional
10              $p_{ij} \leftarrow b_{ij}$
11          else if strategy = zipf
12              $index \leftarrow 1$
13              while $tmp[index] > b_{ij}$
14                  $index \leftarrow index + 1$
15              end while
16              $p_{ij} \leftarrow 1.0/index$
17          end if
18      end for
19      $sum \leftarrow$ sum of elements in row $\mathcal{P}[i]$
20      for $j \in [1, \dots m]$
21          if $sum \neq 0$
22              $p_{ij} \leftarrow (p_{ij} * \alpha_i)/sum$
23          else
24              $p_{ij} \leftarrow \alpha_i/m$
25          end if
26      end for
27  end for
28  return $\mathcal{P}$
```

We propose a negotiation model in which agents follow two rules: 1) if there is a conflict between two agents about the amount of time they wish to cooperate, they compromise by accepting the average of the suggested values; 2) if an agent's total time in different subgroups is not equal to its available time, the times are normalized to fulfill the constraint. The procedure is repeated until the matrix converges successively to a value. The resulted matrix is a successive manipulation of the initial original matrix, toward a schedule matrix in which agents have compromised with other agents on their different desires, while trying to hold the schedule consistent with their time constraints.

A procedure that follows these principles and provides a final symmetric matrix is outlined in Algorithm 2. The input of the algorithm is the agents' preference matrix ($\mathcal{P}$) obtained from Algorithm 1 and its output is a symmetric matrix $\mathcal{P}$.

The algorithm first calculates the unscheduled time of each agent, i.e. vector α (lines $1-3$). Each iteration of the while-loop (lines $4-20$), contains two for-loops. In the first for-loop (lines $5-11$), the matrix $\mathcal{P}$ is replaced by $(\mathcal{P}+\mathcal{P}^t)/2$, i.e. each pair of symmetrically located elements of the matrix are substituted by the average of these two elements. Two agents that have different preferences about their cooperation compromise by choosing the average of the two proposed time amounts. We call this operation (lines $5-11$) *symmetrization*. The second for-loop (lines $12-19$) restores the main characteristic of the matrix $\mathcal{P}$, so that for each row i, the sum of the elements $p_{ij} \in \mathcal{P}[i]$ remains equal to α_i. We call this operation (lines $12-19$) *normalization*.

If none of the two agents a_i and a_j are initially interested in cooperation, that is $a_{ij}=0$ and $a_{ji}=0$, repetition of the symmetrization and normalization operations cannot change this value. If at least one of the agents is interested in cooperation, the algorithm can either force the other agent to cooperate or suppress the agent's initial interest for cooperation. An example of the latter case is the following,

$$\mathcal{P}_1 = \begin{pmatrix} 1.0 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 1.0 & 0 \end{pmatrix} \xrightarrow{793} \begin{pmatrix} 1.0 & 0 & 0 \\ 0 & 0 & 1.0 \\ 0 & 1.0 & 0 \end{pmatrix},$$

where the left hand side matrix after 793 iterations of the while-loop converges to the symmetric matrix at the right hand side of the arrow.

However, the sequence of the symmetrization and normalization does not manipulate all matrices to a symmetric matrix. For instance, consider the matrix

$$\mathcal{P}_2 = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 1.0 & 0 & 0 \\ 1.0 & 0 & 0 \end{pmatrix} \xrightarrow{1} \begin{pmatrix} 0 & 0.5 & 0.5 \\ 1.0 & 0 & 0 \\ 1.0 & 0 & 0 \end{pmatrix},$$

where each sequence of symmetrization and normalization operations results in the same matrix. The algorithm does not provide a symmetric solution for matrix $\mathcal{P}_2$ or any matrix that converges to $\mathcal{P}_2$, e.g.

$$\mathcal{P}_3 = \begin{pmatrix} 1.0 & 0 & 0 \\ 1.0 & 0 & 0 \\ 1.0 & 0 & 0 \end{pmatrix} \xrightarrow{43} \mathcal{P}_2 = \begin{pmatrix} 0 & 0.5 & 0.5 \\ 1.0 & 0 & 0 \\ 1.0 & 0 & 0 \end{pmatrix}.$$

These matrices cannot be manipulated to a symmetric solution, regardless of the negotiation protocol. The problem is that both agents a_2 and a_3 want to cooperate full-time with agent a_1, which leads to a *locked state*.

Although during our tests with simulated teams of different sizes, only few cases resulted in locked states, the negotiation algorithm is extended by a detection and recovery part (lines $21-26$). To recover from locked states, different strategies may be chosen. We use a simple method by adding a random noise to the diagonal of the original matrix (lines $22-24$) and restarting the negotiation once more (line 25). Generally, adding a little number to element p_{ij} of the matrix means to force agent a_i to reconsider cooperation with agent a_j. For example, by adding a small number, $\epsilon = 0.001$ to the diagonal of the matrix $\mathcal{P}_2$, all agents are encouraged to reconsider their initial wishes of not working alone. For $\mathcal{P}_2+\epsilon\mathcal{I}$, the negotiation algorithm provides a symmetric matrix after 40 iterations,

$$\mathcal{P}_2 + \epsilon\mathcal{I} = \begin{pmatrix} 0.001 & 0.5 & 0.5 \\ 1.0 & 0.001 & 0 \\ 1.0 & 0 & 0.001 \end{pmatrix} \xrightarrow{40} \begin{pmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \end{pmatrix}.$$

This result means that agents a_2 and a_3 are forced to work alone one-half of their time, while the desire of agent a_1 is completely accepted.

Algorithm 2: Negotiations of Agents

```
      given   preference matrix: $\mathcal{P} = [p_{ij}]_{m\times m}$
      return  symmetric matrix: $\mathcal{P} = [p_{ij}]_{m\times m}$

 1    for each row $i$ in matrix $\mathcal{P}$
 2        $\alpha[i] \leftarrow sum(\mathcal{P}[i])$
 3    end for
 4    while matrix $\mathcal{P}$ is changing
 5        for each row i in matrix $\mathcal{P}$
 6            for each column j in matrix $\mathcal{P}$ with $i > j$
 7                $average \leftarrow (p_{ij} + p_{ji})/2$
 8                $p_{ij} \leftarrow average$
 9                $p_{ji} \leftarrow average$
10            end for
11        end for
12        for each row i in matrix $\mathcal{P}$
13            $s \leftarrow sum(\mathcal{P}[i])$
15            if $s \neq 0$
14                for each $p_{ij} \in \mathcal{P}[i]$
16                    $p_{ij} = (p_{ij} * \alpha_i)/s$
18                end for
17            end if
19        end for
20    end while
21    if $\mathcal{P}$ not symmetric
22        for each $i \in [1,\ldots,m]$
23            $p_{ii} \leftarrow p_{ii} + random\ \epsilon$
24        end for
25        run Algorithm 2
26    end if
27    return $\mathcal{P}$
```

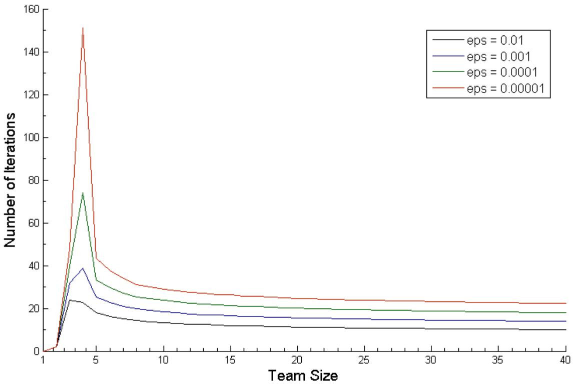

Fig. 2. Convergence of the preference matrix toward a symmetric matrix for different team sizes.

VII. Test of the Algorithms

To investigate how fast Algorithm 2 converges to a symmetric matrix, we let the algorithm manipulates random matrices with different sizes between 2 and 40. For each size 10000 random matrices are produced and the number of iterations in Algorithm 2 before each matrix converges to a symmetric matrix is measured. A matrix is considered to be symmetric if the differences between all pair of symmetrically located elements of the matrix are less than a given value. That is, $|\alpha_{ij} - \alpha_{ji}| < \epsilon, \forall\, i,\, j$, where ϵ is one of the four values $10^{-2}, 10^{-3}, 10^{-4}$ or 10^{-5}.

The results depicted in Fig. 2 show that for all matrix sizes the algorithm converges rapidly to a symmetric matrix. However, the convergence rate is much higher for larger matrices.

The results for the number of times Algorithm 2 reaches a locked state is given in TABLE I for the same set of data, i.e. 10000 different teams for each size (from 2 to 8). As seen from the results, the number of locked states decreases, for larger team sizes. For team size 8, this number is 0. It is notable that the results in Fig. 2 include number of iterations that was required after locked state detection and recovery.

TABLE I
Number of locked states reached in 10000 teams.

Team size	# Locked states	% Locked states
2	0	0.0
3	94	0.94
4	422	4.22
5	27	0.27
6	7	0.07
7	6	0.06
8	0	0.0

To test how algorithms 1 and 2 and the benefit model work together 100000 random tasks and 100000 random teams with different sizes from 2 to 9 agents were created. The four different cooperation strategies were tested and an average of the total benefit of the agents for each team size and each strategy were calculated, the results are summarized in TABLE II.

TABLE II
Total benefit of agents in average for different team sizes

Size	Proportional	Zipf	Equal	Random
2	1.706	1.706	0.853	0.901
3	2.551	2.563	1.707	1.743
4	3.500	3.661	2.564	2.590
5	4.426	4.630	3.415	3.437
6	5.351	5.707	4.271	4.289
7	6.275	6.730	5.127	5.142
8	7.188	7.799	5.974	5.988
9	8.104	8.846	6.824	6.837

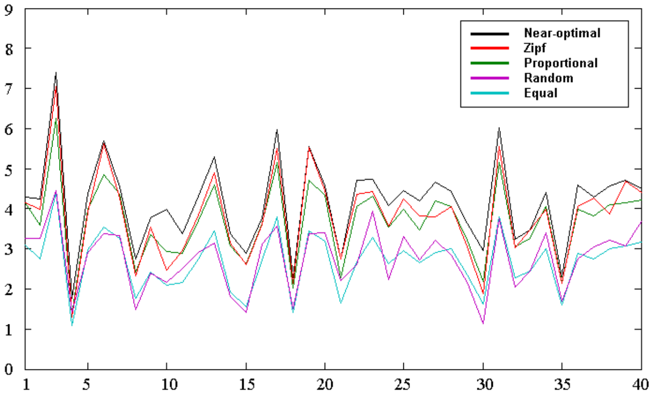

Fig. 3. Comparison of the increase in the value added for different cooperation strategies in a team of 4 agents.

As the results in TABLE II show, the average benefit of the team is higher for the proportional and Zipf strategies compared with the equal and random strategies. Common for proportional and Zipf is that in both strategies agents seek for cooperation with other agents with higher capabilities.

Another interesting observation is the comparison of the proportional and Zipf strategies. In the first case each agent tries to distribute its time exactly in proportion to the benefit of cooperation with other agents, but in the second case the agent needs only to have a rough estimate of other agents' capabilities and tries to cooperate with them according to their ranks. The simulation results show that in most cases the result of the Zipf strategy is slightly higher than distributing the time precisely in proportion to the benefits.

In another test, 40 random tasks and 40 random teams of 4, 6 and 8 agents using 4 different strategies were studied. The performances of teams for these 40 trials are shown in Fig. 3, 4 and 5. These results clearly confirm that the tendency shown by the average values are valid almost in every single trial.

Furthermore, these results are compared with a near-optimal solution. The near-optimal values are computed using a centralized approach and forcing agents to cooperate only with one other agent in a way that maximizes the benefit of the team. In most cases this method produces the optimal value; however there are cases where it is most beneficial that an agent cooperates with more than one agent, hence the name near-optimal. The results show that the strategy based on the Zipf's law (principle of the least effort) is not very far from the near-optimal value.

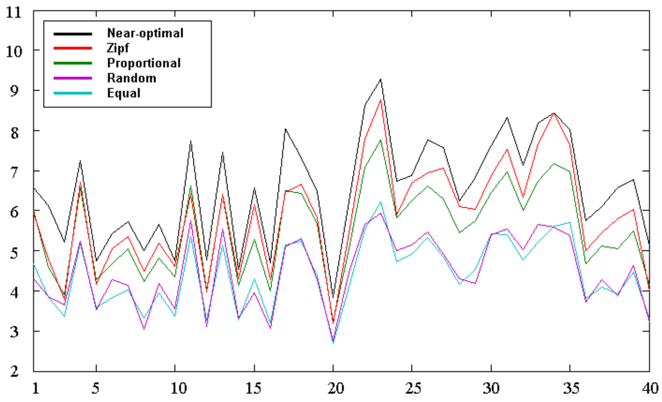

Fig. 4. Comparison of the increase in the value added for different cooperation strategies in a team of 6 agents

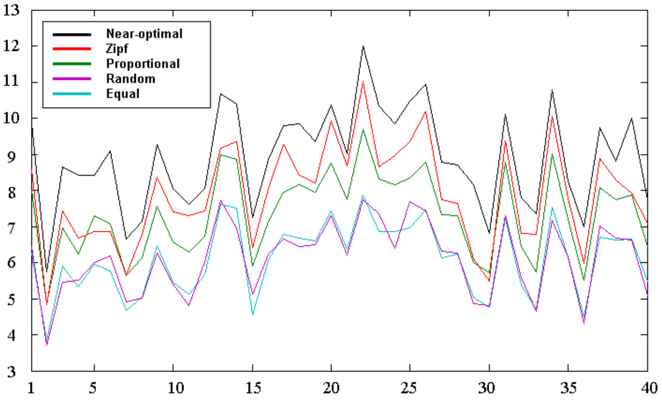

Fig. 5. Comparison of the increase in the value added for different cooperation strategies in a team of 8 agents

The good results of Zipf's strategy may be partially explained by the fact that agents who benefit from each other in almost the same extent, will rank each other in the same manner when following this strategy. Thus, the preference matrix is closer to "symmetric" and agents require less negotiation.

VIII. Conclusions

In this paper, we proposed a general framework for estimating the performance of a team of agents. A team is defined as a set of agents with different capabilities that may cooperate and interact with each other. Teams are partly self-organized and the working time of the agents is not completely scheduled. It means that during a task, different subgroups in which agents cooperate are built. The performance of the team is considered to be the sum of agent's individual contributions and subgroup contributions. In a subgroup, the capabilities of agents are increased under the influence of agents having the highest capabilities. We proposed a model to calculate the benefit of agents from the cooperation. Based on this model and according to a common strategy, agents define a preference matrix, which is a presentation of agents' desires for cooperation with other agents (Algorithm 1). In general, this matrix is not symmetric, which means there is no consensus among the agents about how much they are willing to cooperate with each other. A negotiation algorithm that resolves the conflicts between agents is presented (Algorithm 2). A series of simulation experiments is run to compare the performance of teams with different cooperation strategies. The results of these simulations show that a team in which the members cooperate according to the Zipf's law (principle of least effort) has the highest performance.

Acknowledgment

This paper is part of a project funded by the R&D program of the Swedish Armed Forces and conducted by the Swedish Defence Research Agency (FOI). The authors would like to thank all members of the project group, especially Johan Schubert for his comments and constructive critiques.

References

[1] F. Kamrani, R. Ayani, F. Moradi, and G. Holm, "Estimating performance of a business process model," in *Proc. Winter Simulation Conference (WSC'09)*, M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, Eds., Austin, TX, December 2009.

[2] F. Kamrani, R. Ayani, and A. Karimson, "Optimizing a business process model by using simulation," in *Proceedings of the 2010 Workshop on Principles of Advanced and Distributed Simulation*, Atlanta, GA, May 2010, pp. 40–47.

[3] J. R. Katzenbach and D. K. Smith, *The wisdom of teams : creating the high-performance organization*. Harvard Business School Press, Boston, Mass., 1993.

[4] C. R. Paris, E. Salas, and J. A. Cannon-Bowers, "Teamwork in multi-person systems: a review and analysis," *Ergonomics*, vol. 43, no. 8, pp. 1052–1075, 2000.

[5] E. Salas, C. S. Burke, and J. A. Cannon-Bowers, "Teamwork: emerging principles," *International Journal of Management Reviews*, pp. 339–356, December 2000.

[6] F. Kirschner, F. Paas, and P. A. Kirschner, "Individual and group-based learning from complex cognitive tasks: Effects on retention and transfer efficiency," *Computers in Human Behavior*, vol. 25, no. 2, pp. 306–314, 2009.

[7] E. Kilner and L. A. Sheppard, "The role of teamwork and communication in the emergency department: A systematic review," *International Emergency Nursing*, pp. 127–137, July 2010.

[8] D. Siassakos, T. Draycott, J. Crofts, L. Hunt, C. Winter, and R. Fox, "More to teamwork than knowledge, skill and attitude," *BJOG: An International Journal of Obstetrics & Gynaecology*, pp. 1262–1269, September 2010.

[9] D. deB. Beaver, "Collaboration and teamwork in physics," *Czechoslovak Journal of Physics*, pp. 14–18, 1986.

[10] G. H. Walker, N. A. Stantona, P. Salmona, and D. Jenkinsa, "How can we support the commander's involvement in the planning process? an exploratory study into remote and co-located command planning," *International Journal of Industrial Ergonomics*, vol. 39, pp. 456–464, March 2009.

[11] W. B. Rouse and R. Rouse, "Teamwork in the performing arts," *Proceedings of the IEEE: Special Issue on Engineering & Music*, vol. 92, no. 4, pp. 606–615, April 2004.

[12] A. Bikfalvi, "Teamwork in production: Implementation, its determinants, and estimates for German manufacturing," *Human Factors and Ergonomics in Manufacturing & Service Industries*, vol. 21, 2011.

[13] B. Prasad, *Concurrent Engineering Fundamentals, Volume I: Integrated Product and Process Organization*. New Jersey: Prentice-Hall, 1996.

[14] ——, "Decentralized cooperation: a distributed approach to team design in a concurrent engineering organization," *Team Performance Management*, vol. 4, pp. 138–165, 1998.

[15] G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison-Wesley, Ed., 1949.

[16] H. A. Simon, "A behavioral model of rational choice," *Quarterly Journal of Economics*, vol. 69, no. 1, pp. 99–118, 1955.

[17] E. W. Weisstein, "Doubly stochastic matrix," Available via http://mathworld.wolfram.com/DoublyStochasticMatrix.html [accessed June 14, 2010], From MathWorld–A Wolfram Web Resource, 2010.