How Can Robots Trust Each Other For Better Cooperation?
A Relative Needs Entropy Based Robot-Robot Trust Assessment Model

Qin Yang

Ramviyas Parasuraman

Abstract— Cooperation in multi-agent and multi-robot systems can help agents build various formations, shapes, and patterns presenting corresponding functions and purposes adapting to different situations. Relationship between agents such as their spatial proximity and functional similarities could play a crucial role in cooperation between agents. Trust level between agents is an essential factor in evaluating their relationships' reliability and stability, much as people do. This paper proposes a new model called Relative Needs Entropy (RNE) to assess trust between robotic agents. RNE measures the distance of needs distribution between individual agents or groups of agents. To exemplify its utility, we implement and demonstrate our trust model through experiments simulating a heterogeneous multi-robot grouping task in a persistent urban search and rescue mission consisting of tasks at two levels of difficulty. The results suggest that RNE trust-Based grouping of robots can achieve better performance and adaptability for diverse task execution compared to the state-of-the-art energy-based or distance-based grouping models.

I. INTRODUCTION

Trust describes the interdependent relationship between agents [1], which can help us better understand the dynamics of cooperation and competition, the resolution of conflicts, and the facilitation of economic exchange [2]. In different computing fields, trust is different based on various perspectives. However, in computational agents, these models do not involve the trustor's characteristics but focus on forming trust based on past behavior [3].

In *Automation*, authors in [4] describe trust in human-automation relationships as the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability. The trustor here is a human and the trustee usually is the automation system or an intelligent agent like robot. Those systems' primary purpose is to assess and calibrate the trustors' trust beliefs reflecting the automation system's ability to achieve a specific task.

In *Computing and Networking*, the concept of trust involves many research fields, such as artificial intelligence (AI), human-machine interaction, and communication networks [5], [6], [7]. They regard an agent's trust as a subjective belief which represents the reliability of the other agents' behaviors in a particular situation with potential risks. Their decisions are based on learning from the experience to maximize its interest (or utility) and minimize risk [3]. Especially in the social IoT domain [8], [9], trust is measured based on the information accuracy and agents' intentions

* The authors are with the Heterogeneous Robotics Research Lab (HeRoLab), Department of Computer Science, University of Georgia, Athens, GA 30602, USA. Email: {qy03103,ramviyas}@uga.edu.
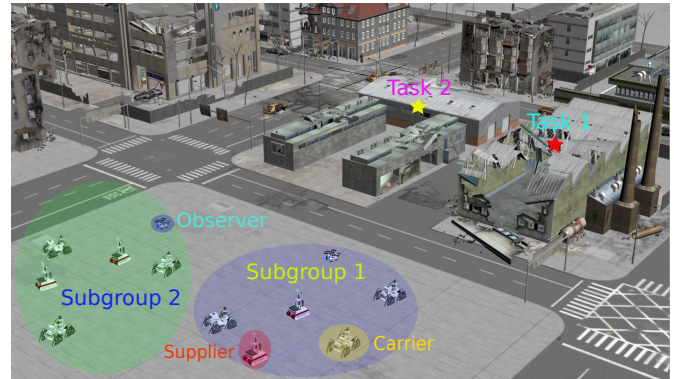
Fig. 1. Illustration of two heterogeneous robot teams (ground robots + aerial robots) cooperatively achieving tasks of two different difficulty levels in a post-nuclear leak rescue mission.

(friendly or malicious) according to the current environment to avoid attacks on the system.

In *Multi-Robot Systems (MRS)*, trust by one agent on another agent reflects the trusting agents' mission satisfaction, organizational group member behaviors and consensus, and task performance at the individual level. Moreover, trust has been associated with system-level analysis, such as reward and function at the organized group structure in a specific situation, the conflict solving of heterogeneous and diverse needs, and the system evolution through learning from interaction and adaptation. On the other hand, the act of trusting by an agent can be conceptualized as consisting of two main steps: 1) *trust evaluation*, in which the agent assesses the trustworthiness of potential interaction partners; 2) *trust-aware decision-making*, in which the agent selects interaction partners based on their trust values [10]. Also, assessing the performance of agent trust models is of concern to agent trust research [11].

Considering the interactions between human agents and artificial intelligence agents like human-robot interaction (HRI), building stable and reliable relationships is of utmost importance in MRS cooperation, especially in adversarial environments and rescue missions [12]. Fig. 1 illustrates two heterogeneous robot teams cooperating to achieve two different challenging tasks in a post-nuclear leak rescue mission. In such multi-agent missions, appropriate trust in robotic collaborators is one of the leading factors influencing HRI performance [13]. In HRI, factors affecting trust are classified into three categories [14]: 1) *Human-related factors* (i.e., including ability-based factors, characteristics); 2) *Robot-related factors* (i.e., including performance-based factors and attribute-based factors); 3) *Environmental factors*

(i.e., including team collaboration, tasking, etc.).

Although modeling trust has been studied from various perspectives and involves many different factors, it is still challenging to develop a conclusive trust model that incorporates all of these factors. Therefore, future research into trust modeling needs to be more focused on developing general trust models based on measures other than the countless factors affecting trust [13]. Specifically, no work from the literature considered trust in socio-intelligent systems from an artificial intelligent agent-perspective (robots) to model trust with agents having similar needs and interests to the best of our knowledge. Balch [15] laid the foundations in introducing behavioral difference and social entropy metrics to evaluate diversity in societies of mechanically similar agents from the agent's behaviors perspective in an MRS. However, it only looks from a diversity point of view and does not establish a substantial and clear connection between the agent's behaviors and motivations, especially trust.

To bridge this important gap, in this paper, we introduce a general agent-agent trust model based on *Relative Needs Entropy (RNE)*. Here, we extend our previous work [16], where we defined needs hierarchy for robots similar to Maslow's law of human needs. Using the current state of the needs of the agents, it is possible to devise a relative needs entropy value, which can help measure social trust between agents. We verify this trust model in a simulated urban search and rescue (USAR) mission to achieve multi-robot grouping based on trust level within the robots in a group. We analyze the relative mission performance against state-of-the-art methods which use energy capability (battery level) [17] or the spatial proximity (i.e., the inter-distance) between the robots for distributed task allocation [18]. The specific contributions made in this paper are as follows:

1) **RNE-based robot-robot[1] and multi-robot group trust models** Considering intelligent agents representing different needs in interaction, especially for the heterogeneous MRS, we introduce a new RNE-based trust assessment model to describe trust levels between agents or groups. It presents the similarity of their needs.

2) **RNE trust based heterogeneous multi-robot grouping mechanism** We propose a hierarchical approach for RNE trust-based formation of groups in a heterogeneous MRS. This method will first form homogeneous agent subgroups, then hierarchically merges according to a specific sequence until forming the final heterogeneous groups' combination.

We provide examples of the proposed trust model and demonstrate its effectiveness in heterogeneous multi-robot grouping for a simulated urban search and rescue (USAR) mission, where the robots should form two specific groups to finish two tasks at easy and hard difficulty levels based on the environment. We hypothesize that RNE trust-based group formation would achieve better cooperation (lower costs and higher group rewards) by balancing and aligning their motivations (i.e., agent needs).

---

[1]We use the terms robot and agent interchangeably

II. BACKGROUND AND PRELIMINARIES

This section briefly reviews the *relative Entropy*, and our previous study on *robot needs hierarchy*. The relative definitions and notations are described below.

A. Relative Entropy

The *relative entropy* or the Kullback Leibler distance is a measure of distance between two distributions. In statistics, it arises as an expected logarithm of the likelihood ratio. The relative entropy $D(p||q)$ is a measure of the inefficiency of assuming that the distribution is q when the true distribution is p [19]. The corresponding mathematical definition is shown as Eq. (1).

$$
\begin{aligned}
D(p||q) &= \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \\
&= E_p \log \frac{p(X)}{q(Y)}.
\end{aligned}
\tag{1}
$$

Furthermore, the *mutual information* [19] represents a measure of the amount of information that one random variable contains about another random variable and the reduction in the uncertainty of one random variable. Supposing we consider two random variables X and Y with a joint probability mass function $p(x, y)$ and marginal probability mass function $p(x)$ and $p(y)$. The *mutual information* $I(X; Y)$ is the *relative entropy* between the joint distribution and the product distribution $p(x)p(y)$ as Eq. (2).

$$
\begin{aligned}
I(X; Y) &= \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \\
&= D(p(x, y)||p(x)p(y)) \\
&= E_{p(x,y)} \log \frac{p(X, Y)}{p(X)P(Y)}.
\end{aligned}
\tag{2}
$$

B. Robot Needs Vector

In *Robot Needs Hierarchy* [16], [17], the agent's needs expressed as the *Needs Expectation*, which can be calculated through its behaviors' weight and corresponding safety probability based on the data of perception and communication. The individual safety needs (Eq. (3)) are the precondition for calculating the basic needs (Eq. (4)). Similarly, only after fitting the safety and basic needs can consider its capability needs (Eq. (5)) and the teaming needs (Eq. (6))

$$
Safety \ \ Needs: \ \ N_{s_{t_i}} = \sum_{i=1}^{s_{t_i}} W_i \cdot \mathbb{P}(W_i|P, C);
\tag{3}
$$

$$
Basic \ \ Needs: \ \ N_{b_i} = \sum_{i=1}^{b_i} W_i \cdot \mathbb{P}(W_i|P, C, N_{s_{t_i}});
\tag{4}
$$

$$
Capability \ \ Needs: N_{c_i} = \sum_{i=1}^{c_i} A_i \cdot \mathbb{P}(A_i|T, P, N_{b_i});
\tag{5}
$$

$$
Teaming \ \ Needs: \ \ N_{t_i} = \sum_{i=1}^{n} U_i \cdot \mathbb{P}(U_i|P, C, N_{c_i});
\tag{6}
$$

Here, P and C represent the data of the agent's perception and communication separately; T represents the task requirement space; U_i represents the utility value of agent i in the

group; W represents corresponding weights; A represents the level of the agent's corresponding capabilities based on task requirements; b_i, s_{t_i}, and c_i represent the size of agent i's basic, safety, and capability needs solution space respectively; n represents the number of agents in the group.

Through the above analysis, we adopt *Utility Theory* [20], [21] to define the agent's fourth level needs – *Teaming Needs* (Eq. (6)), representing higher-level needs for an intelligent agent. It can be regarded as a kind of motivation or requirement for cooperation achieving specific goals or tasks to satisfy the individual or group's certain expected utilities.

C. Relative Needs Entropy

Similar to the information entropy, we define the needs entropy as the difference or distance of needs distribution between agents in a specific scenario for an individual or groups. Here, needs of the robots are regarded as their motivations. From a statistical perspective, the RNE can be regarded as calculating the similarity of high-dimensional samples from the robot needs vector. **A lower RNE value means that the trust level between agents or groups is higher because their needs are well-aligned and there is low difference (distance) in their needs distributions**. Similarly, a higher RNE value will mean that the needs distributions are diverse, and there exists a low trust level between the agent or groups because of their misalignment in their motivations, which are similar to each other. More specifically, we formalize the RNE based trust model on three different situations in Sec. III.

III. PROPOSED RNE BASED TRUST MODELS

Definition 1 (Trust between Agents): Supposing the needs' vectors of R_1 and R_2 are $N_{R_1}(n_{11}, \ldots, n_{1j})$ and $N_{R_2}(n_{21}, \ldots, n_{2j})$, where j is the number of specific needs (categories) in the needs space. Then, through the corresponding weight vector $W(w_1, \ldots, w_j)$, we get the needs' distribution of two agents are $D_{R_1}(d_{11}, \ldots, d_{1j})$ and $D_{R_2}(d_{21}, \ldots, d_{2j})$ respectively. We can present the RNE based Trust value from R_1 to R_2 as Eq. (7). Here, d_{1k} and d_{2k} are calculated as Eq. (8). ($j, k \in Z^+$)

$$\mathbb{T}(R_1\|R_2) = \sum_{k=1}^{j} D_{R_{1_k}} \cdot \log \frac{D_{R_{1_k}}}{D_{R_{2_k}}} \tag{7}$$

$$D_{R_{1/2}} \ni d_{1k/2k} = n_{1k/2k} \cdot w_k / \sum_{k=1}^{j} (n_{1k/2k} \cdot w_k) \tag{8}$$

Note, $\mathbb{T}(R_1\|R_2) \not\equiv \mathbb{T}(R_2\|R_1)$ since we consider relative entropy. That is, robot R_1's trust on the robot R_2 is not necessarily reciprocal (equal to the robot R_2's trust on R_1) because the reference "true" distributions are different in both cases. Moreover, the robot's information on other robot's needs expectations of current levels need not be accurate because of information uncertainty. This makes the RNE trust model applicable in most realistic scenarios where robots rely on perception and communication to gather information about other robots in the scenario.

Example 1: Let a robot has three needs value to represent its safety, energy, and health levels. Assume the needs' vectors of three robots R_1, R_2, and R_3 are [86, 120, 30], [20, 30, 10], and [80, 115, 25], respectively. The corresponding weight vector is [6, 4, 2][2], which represents the priority each needs level belongs to. According to Eq. (8), we can get their needs' distributions as D_{R_1}(0.4884, 0.4545, 0.0568), D_{R_2}(0.4615, 0.4615, 0.0769), and D_{R_3}(0.4848, 0.4646, 0.0505). Then,

$$\mathbb{T}(R_1\|R_2) = 0.4886 \cdot \log \frac{0.4886}{0.4615} + 0.4545 \cdot \log \frac{0.4545}{0.4615} + \\ 0.0568 \cdot \log \frac{0.0568}{0.0769} = 0.00373 \tag{9}$$

$$\mathbb{T}(R_1\|R_3) = 0.4886 \cdot \log \frac{0.4886}{0.4848} + 0.4545 \cdot \log \frac{0.4545}{0.4646} + \\ 0.0568 \cdot \log \frac{0.0568}{0.0505} = 0.0005 \tag{10}$$

By calculating the RNE values between R_1 and R_2 Eq. (9), and R_3 Eq. (10), the trust values $\mathbb{T}(R_1\|R_3) < \mathbb{T}(R_1\|R_2)$. Therefore, as per the definition of RNE in Sec. II-C, robot R_1 has a higher trust level with R_3 compared to its trust on R_2 because robot R_1 shares similarities with the robot R_3 in terms of needs distributions and hence the RNE value is lower.

Definition 2 (Trust between Agent and Groups): Supposing the needs' vector of the agent R is N_R, and the needs matrix of the group R_G is $(N_1, \ldots, N_i)$, where i represents the number of robots in the group with corresponding weights $W(w_1, \ldots, w_j)$ for the needs categories. According to Def. 1, we can represent the needs' distribution vector of R as D_R. Considering R_G needs' distribution vector as D_{R_G} Eq. (11), the RNE Trust value from the robot R to the group R_G can be represented as Eq. (12). ($i, j \in Z^+$)

$$D_{R_G} = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_i \end{bmatrix}_{i \times j} . \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_j \end{bmatrix}^T ./ \left[\sum_{k=1}^{i} (N_k \cdot W_k) \right]_{j \times 1} \tag{11}$$

$$\mathbb{T}(D_R\|D_{R_G}) = \sum_{k=1}^{j} D_{R_k} \cdot \log \frac{D_{R_k}}{D_{R_{G_k}}} \tag{12}$$

Example 2: In addition to the Robot R_1 from Example 1, assume we have two robot groups R_{G_1} and R_{G_2} with two robots in each group. Their needs' matrices are {[82, 114, 24]; [79, 117, 23]} and {[40, 56, 12]; [56, 48, 15]} respectively. The weight vector of needs is the same as in Example 1. Through the Eq. (11), we get their needs' distributions as $D_{R_{G_1}}$ (0.4869, 0.4657, 0.0474) and $D_{R_{G_2}}$ (0.5507, 0.3977, 0.0516). Then we can get the RNE values

---

[2]Due to the difference in individual needs and situations, the weight vector might be different correspondingly. Here, we consider equal weights.

between robot R_1 and group R_{G_1} Eq. (13), and R_{G_2} Eq. (14) with the help of agent-group RNE trust in Eq. (12) correspondingly.

$$\mathbb{T}(R_1||R_{G_1}) = 0.4886 \cdot \log \frac{0.4886}{0.4869} + 0.4545 \cdot \log \frac{0.4545}{0.4657} +$$
$$0.0568 \cdot \log \frac{0.0568}{0.0474} = 0.000915 \tag{13}$$

$$\mathbb{T}(R_1||R_{G_2}) = 0.4886 \cdot \log \frac{0.4886}{0.5507} + 0.4545 \cdot \log \frac{0.4545}{0.3977} +$$
$$0.0568 \cdot \log \frac{0.0568}{0.0516} = 0.007674 \tag{14}$$

Here, since $\mathbb{T}(R_1||R_{G_1}) < \mathbb{T}(R_1||R_{G_2})$ the trust of robot R_1 on the group R_{G_1} is higher than its trust on the group R_{G_2} because robot R_1 share similar needs with the average needs distribution of the group R_{G_1}.

Definition 3 (Trust between Groups): For certain state $s_1 \in S$, supposing two groups R_{G_1} and R_{G_2} have the current needs matrix $(N_{11}, \ldots, N_{1i})$ and $(N_{21}, \ldots, N_{2i})$, Eq. (15). Considering each group member's needs vector $(n_{1k_1/2k_1}, \ldots, n_{1k_j/2k_j})$ consists of different needs elements with corresponding weights $W(w_1, \ldots, w_j)$. Let D_{N1} $(D_{11}, \ldots, D_{1j})$ and D_{N2} $(D_{21}, \ldots, D_{2j})$ present the agents needs' distribution of two groups. The RNE value (Trust) from group one to two in the current scenario can be defined as Eq. (17).

$$\begin{bmatrix} N_{11/21} \\ N_{12/22} \\ \vdots \\ N_{1i/2i} \end{bmatrix} = \begin{bmatrix} n_{11_1/21_1} & n_{11_2/21_2} & \cdots & n_{11_j/21_j} \\ n_{12_1/22_1} & n_{12_2/22_2} & \cdots & n_{12_j/22_j} \\ \vdots & \vdots & \ddots & \vdots \\ n_{1i_1/2i_1} & n_{1i_2/2i_2} & \cdots & n_{1i_j/2i_j} \end{bmatrix} \tag{15}$$

$$D_{N1/N2} = \begin{bmatrix} D_{11/21} \\ D_{12/22} \\ \vdots \\ D_{1i/2i} \end{bmatrix} = \begin{bmatrix} N_{11/21} \\ N_{12/22} \\ \vdots \\ N_{1i/2i} \end{bmatrix}_{i \times j} . \times$$
$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_j \end{bmatrix}^T ./ \left[\sum_{k=1}^{i} (N_{1i_k/2i_k} \cdot W_k) \right]_{j \times 1} \tag{16}$$

$$\mathbb{T}(D_{N_1}||D_{N_2}) = \sum_{k=1}^{j} D_{N_{1_k}} \cdot \log \frac{D_{N_{1_k}}}{D_{N_{2_k}}} \tag{17}$$

Example 3: Let us calculate the RNE trust value from group R_{G_1} to R_{G_2} based on Example 2 after ceiling the needs distributions to two decimals.

$$\mathbb{T}(R_{G_1}||R_{G_2}) = 0.48 \cdot \log \frac{0.48}{0.55} + 0.46 \cdot \log \frac{0.46}{0.39} +$$
$$0.04 \cdot \log \frac{0.04}{0.05} = 0.0007,$$
$$\mathbb{T}(R_{G_2}||R_{G_1}) = 0.55 \cdot \log \frac{0.55}{0.48} + 0.39 \cdot \log \frac{0.39}{0.46} +$$
$$0.05 \cdot \log \frac{0.05}{0.04} = 0.0094. \tag{18}$$

From this example, we can see that the robot group R_{G_1} has a significantly higher level of trust on the group R_{G_2} than the vice versa. This aligns with the intended social needs and motivations for intelligent agents (robots) where each robot views another robot from its own perspective rather than grounding on a common reference.

According to the above definitions and examples, we can see that RNE assesses the trust levels of the specific agent or team to others, which stands on the current individual or group perspective analyzing the needs' similarity from itself to others. It means that different agents or groups might have distinct RNE values or trust levels based on their current needs, knowledge of the environment and other agents, and future interests, much as humans do. In the real world, we usually assign the more challenging tasks to the agents or groups with reliable capabilities and stable performance, and vice versa for simpler tasks where grouping of agents does not need to account for trust within the group. However, most real-world tasks are challenging and has varying spatio-temporal difficulty levels. Therefore, grouping agents with similar corresponding capabilities and needs to the related mission is crucial in practical applications.

IV. HETEROGENEOUS MULTI-ROBOT SYSTEMS

We design a robot-aided urban search and rescue (USAR) mission to implement and illustrate the heterogeneous multi-robot grouping concept and corresponding algorithms. The mission is to retrieve (or rescue) as many resources (e.g., victims) as possible from the task area where the resources are present. In our scenarios, a set of robots will be available. Each robot is classified as one of the following: *Carrier*, *Supplier*, and *Observer*. Each type of robot has a specific role and functionality [12]. Multiple robots cooperate to fulfill this rescue mission within a limited time [22]. When they form a group, there must be robots from each of these three categories, making the group heterogeneous in terms of functionalities.

This USAR mission is executed in multiple rounds by the heterogeneous MRS. Each round, the entire group will divide into two subgroups based on certain principles, executing the corresponding tasks. There are two kinds of adversaries (debris and radioactive resources) randomly located in our post-nuclear leak rescue mission environment. According to the task's difficulty, the distance from the initial position to the task and the number of adversaries distributed in the task area are different. See Fig. 2 for an illustration of the heterogeneous MRS in the USAR mission under study.

Considering the various individual agent's needs in different situations, we organize their needs based on *Robot Needs Hierarchy* as follows:

- Safety Needs: HP (Health Points), Speed, and Sensing Range (detecting adversaries);
- Basic Needs: Energy (battery);
- Capability Needs: Capacity for carrying, Resource for rescuing, and Observing Range (detecting rescuee and rescued objects);
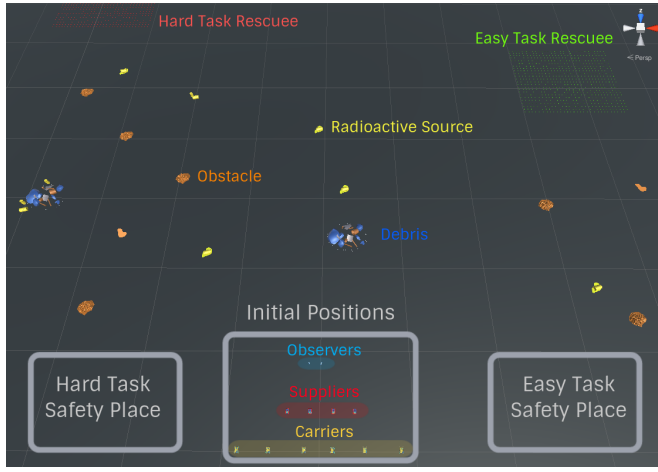
Fig. 2. Illustration of the simulation of a post-nuclear leak rescue mission.

Algorithm 1: RNE Trust-based Multi-robot Grouping

Input	:	Carrier, Supplier, and Observer group state matrixes: A_o, A_s, and A_o. The number of tasks: 2.
Output	:	The maximum RNE difference group partition (cso_1, cso_2).

1: initialization;
2: memory data set $V_{c/s/o/cs/cso} = \{\varnothing\}$;
3: $m = A_{c/s/o}$.Count / 2;
4: **for** *each* $i \in ASC(A_{c/s/o}, m)$ **do**
5:    | $V_{c/s/o}$.Add($GRC(i[0], i[1])$)
6: **end**
7: $G_{c/s/o}$ = Max($V_{c/s/o}$);
8: **for** *each* $i \in G_c$ **do**
9:    | **for** *each* $j \in G_s$ **do**
10:    | | V_{cs}.Add($GRC(i, j)$)
11:    | **end**
12: **end**
13: G_{cs} = Max(V_{cs});
14: **for** *each* $i \in G_{cs}$ **do**
15:    | **for** *each* $j \in G_o$ **do**
16:    | | V_{cso}.Add($GRC(i, j)$)
17:    | **end**
18: **end**
19: **return** G_{cso} = Max(V_{cso}).

- Teaming Needs: The number of rescuees or rescued objects (Group Utility).

In our USAR mission, we assume that the number of *Carrier*, *Supplier* and *Observer* are x, y and z ($x, y, z \in Z^+$). According to Section III and IV discussion, we define their *Needs Space*[3] as Eqs. (19), (20), and (21), respectively.

$$Carrier := N_c(hp_c, v_c, sen_c, eng_c, res_c, cap_c, obs_c); \quad (19)$$

$$Supplier := N_s(hp_s, v_s, sen_s, eng_s, res_s, cap_s, obs_s); \quad (20)$$

$$Observer := N_o(hp_o, v_o, sen_o, eng_o, res_o, cap_o, obs_o). \quad (21)$$

Here,

- hp, v, and sen represent the agent's safety needs like health points, velocity, and sensing range separately;
- eng represents the agent's basic energy needs such as battery level;
- cap, res, and obs represent the amount of the agent's carrying capacity and rescuing resources, and observing range for searching correspondingly.

When agents pass through the radioactive area, their health will be damaged continuously. For the safety needs, the individual needs should guarantee that the robot has enough HP supporting its normal function. From another perspective, through sensing perception and speed adaptation, they should avoid the collision between agents and adversaries like obstacles (debris) and radioactive sources. Similarly, they also need to satisfy corresponding energy fulfilling tasks [23]. Otherwise, it will return to the resting place for recharging its energy or refreshing its HP.

The individual robot should represent corresponding abilities for the task to satisfy the mission's capability needs, especially for the heterogeneous agents' cooperation. Our USAR mission assumes that saving one rescued object will cost one space (capacity) and one rescuing resource. Besides, each group member's speed will decrease by 30% without an observer involving the group, and group members can share their resources in the specific task.

[3]The needs space has 7 variable, i.e., j=7 in Eqs. (7), (12), and (17).

V. RNE Trust-based Multi-Robot Cooperation

Supposing we have two tasks in our USAR mission, which means that the entire group needs to divide into two subgroups fulfilling the corresponding task. These two tasks have different difficulty levels - easy and hard. Compared to the easy task, the hard task has more obstacles (debris) and radioactive sources in the disaster area, which means that agents might cost more energy avoiding obstacles and more HP resisting radiation emitting from radioactive resources. Besides, agents will spend more time and energy executing the challenging task because of the longer distance.

For the multi-robot grouping, we organize the group members and implement the *bottom-up* mechanism instead of partition without classification and hierarchy. More specifically, in the three homogeneous groups – *Carrier*, *Supplier* and *Observer*, we first consider separating them into two subgroups based on different combinations $C_x^{\frac{x}{2}}$, $C_y^{\frac{y}{2}}$, and $C_z^{\frac{z}{2}}$, respectively. Then, we can always find a combination with the most significant RNE value difference between those subgroups in the corresponding homogeneous group. Supposing the three groups' final combinations are $G_c(Car_1, Car_2)$, $G_s(Sup_1, Sup_2)$, and $G_o(Obs_1, Obs_2)$, using the same approach, we can merge the subgroups of carrier and supplier into a new combination $G_{cs}(CS_1, CS_2)$ recursively. Until we get the final group partition $G_{cso}(CSO_1, CSO_2)$, we finish the entire RNE trust-based grouping. The whole process can be represented as the main RNE trust-based multi-robot grouping algorithm in Alg. 1, with supporting functions and algorithms in Alg. 2 (obtaining hierarchical subgroup combinations), 3 (calculating group's RNE trust values), Alg. 4 (calculating needs distribution vector of a robot group), and Alg. 5 (iteratively solving deadlocks in robot's task assignments).

The multi-robot grouping implementation is based on our previous work in *Self-Adaptive Swarm System* (SASS) [17]. Here, the individual agent selecting a suitable group relies on the trust (RNE) calculation, which means that agents with

---

Algorithm 2: All Subgroup Combinations (ASC)

Input : Group members' state list A; The partition point: m.
Output : All subgroup combination list S_A.

1: initialization;
2: memory data set $agent$, V, $D = \{\varnothing\}$
3: **if** $A.Count == m$ And $m \neq 1$ **then**
4:     **return** S_A.Add(A)
5: **end**
6: **else if** $m == 1$ **then**
7:     **for** $each\ i \in A$ **do**
8:        V.Add(i);
9:        S_A.Add(V);
10:        $V = Null$.
11:     **end**
12:     **return** S_A;
13: **end**
14: **for** $each\ i \in (A.Count - m)$ **do**
15:     $agent = A[0]$;
16:     A.Remove(0);
17:     $D = ASC(A, m - 1)$;
18:     **for** $each\ j \in D.Count$ **do**
19:        V.Add($agent$);
20:        V.Add($D[j]$);
21:        S_A.Add(V);
22:        $V = Null$.
23:     **end**
24: **end**
25: **return** S_A.

Algorithm 3: Groups RNE Calculation (GRC)

Input : Group 1 and 2 state matrixes A_1 and A_2.
Output : The RNE value from group 1 to 2 RNE_{12}.

1: initialization;
2: memory data set $D_{1/2} = \{\varnothing\}$
3: $D_1 = NDC(A_1.needs,\ A_1.weights)$;
4: $D_2 = NDC(A_2.needs,\ A_2.weights)$;
5: **for** $each\ i \in Max(D_1.Count, D_2.Count)$ **do**
6:     $RNE_{12}\ += D_1[i] \times \log(D_1[i]/D_2[i])$
7: **end**
8: **return** RNE_{12}

Algorithm 4: Needs' Distribution Calculation (NDC)

Input : Group members' needs and weight matrix: N and W.
Output : The group needs' distribution vector D.

1: initialization;
2: memory data set V, $D = \{\varnothing\}$
3: **for** $each\ i \in N.Count$ **do**
4:     V.Add($N[i] \times W[i]$)
5: **end**
6: **for** $each\ i \in V.Count$ **do**
7:     D.Add($V[i]\ /\ V$.Sum())
8: **end**
9: **return** D.Sort()

Algorithm 5: Deadlock Solving Mechanism (DSM)

Input : Sorted collision agents' state list C based on agent's needs.
Output : Solving the deadlock.

1: % Detect the deadlock.
2: **while** $length(C) \neq 0$ And $\Delta C.velocity == 0$ **do**
3:     i++;
4:     % Switch the execution order from the high to the low priority agent.
5:     $C_n = Swap(C, 0, C[i])$;
6:     % Execute the new collision list in the collision avoiding planning.
7:     CollisionAvoiding(C_n);
8:     % Check whether the deadlock is existing.
9:     $DSM(C_n)$.
10: **end**
11: $i = 0$;
12: $C_n = \{\varnothing\}$;
13: **return**

---

similar needs levels will form a group executing the corresponding task. Besides, through the *Negotiation-Agreement Mechanism* and route planning, agents can efficiently avoid conflicts and collisions between agent to agent and agent to static obstacles (debris). More specifically, for the SASS *Collision Avoidance* planning, the agent with the highest priority in the collision list will move first, and the rest of the agents stop until it out of the list. Then, they keep on executing the process recursively until the collision list is empty. Especially for this deadlock solving problem, when all the collision list agents can not move, we build a dynamic priority-based switching mechanism providing each agent has a fair opportunity to move until solving the deadlocks (see Alg. 5).

VI. Evaluation through Simulation Studies

Considering the cross-platform, scalability, and efficiency of the simulations, we chose the "Unity" game engine to simulate the USAR mission. In our simulations, we consider six carriers, four suppliers, and two observers in total which separated into two equally-numbered subgroups executing two complex tasks in a post-nuclear rescue mission to rescue the resources and valuable items as much as possible within a limited time period. We compare the RNE trust-based grouping method with other methods from the literature.

A. Performance Metrics

We consider the following performance metrics in this study for a comparative analysis.

No. of Rescuees: We calculate the total number of resources the robot groups have rescued in both the tasks in the USAR mission for ten minutes per trial.

Energy Cost Per Rescuee: This is the total battery level spent by all robots divided by the total number of rescuees. From the agent basic needs perspective, we calculate the energy cost per rescuee for each category task and analyze the integrated energy cost per rescuee per trial.

HP Cost Per Rescuee: This is the total Health points used by all robots divided by the total number of rescuees. Here, we analyze the HP cost per rescuee performance similar to the *Energy Cost Per Rescuee*.

B. Simulation Studies

Until the timeout is reached, multiple rounds of rescue mission is executed. After each round, when all agents are back to the initial position, the whole group members will regroup based on agents' current grouping principle, then distributed to the corresponding task continuing the mission. From the individual perspective, if it passes through the radioactive area, its HP will cost 0.0003% per step, and we assume that each agent costs 0.0045% energy (battery point) for every moving step. Besides, to satisfy the agent's safety and basic needs, we assume that if the individual battery (energy) or HP level is below 30%, it needs to go to the pre-defined rest position for recharging energy (or for refreshing its health HP) 30 seconds, then back to the initial position waiting for the next round to start.

In the experiments, the agent's HP level affects its speed, sensing, and observing range, which means that the lower its HP, the smaller its velocity, sensing, and observing range. Besides, without observer involvement, each group member's speed will decrease by 30%. Moreover, each group member can share their resources and information between a heterogeneous subgroup to achieve the specific task cooperatively in every round. We present the needs' relationships between *Carrier*, *Supplier*, and *Observer* as follows:

- *Safety Needs*: $hp_s = hp_c = 2hp_o$; $v_c = v_s = 1.5v_o$; $sen_o = 6sen_c = 6sen_s$.
- *Basic Needs*: $eng_s = eng_c = 1.5eng_o$;
- *Capability Needs*: $cap_c = 6cap_s$, $cap_o = 0$; $res_s = 10res_c$, $res_o = 0$; $obs_o = 1000obs_c = 1000obs_s$.

Moreover, each subgroup (i.e., each task) has one observer, two suppliers, and three carriers. We conduct ten simulation trials for each principal case (method) in the whole experiment. In every case, we use the same ten different initial battery (energy) and HP levels sampled from two Gaussian distributions with means of 80% and 90%, and standard deviations of 20% and 10% respectively.

C. Compared methods from the literature

To evaluate our RNE Trust model for grouping, we implement the state-of-the-art energy-based (ENG) method [17] and use the distance-based (DIS) grouping approach as the baseline in the performance comparison. Besides, we combine the agent's HP level and distance (HP$_{DIS}$) as another grouping principle applied in the experiments. We discuss the specific details for each case (method) as follows:

DIS - Distance-based: Each agent will compare the distances between its initial position and the goal points of two tasks, then selects the nearest one forming a group to execute. They continue to form a group until the necessary count of robots has reached for that group. This is a typical mode of multi-robot grouping used in the literature. For instance, in [18], the authors used shortest distance to the task as the measure to determine the group membership.

ENG - Energy-based: According to our previous work in [17], [16], where the agents with the same category will sort based only on their current energy (battery) levels at the initial positions, then divide into two parts with equal numbers. By selecting the high or low parts in each category to form the corresponding two heterogeneous subgroups, we assign the higher energy group to the challenging task and the lower one to the easy one, respectively.

HP$_{DIS}$ - Health points combined with the distance based: Similar to the energy-based grouping, here we use the health level of the robots. First, consider sorting the robots in each category (type) based on their current health points (HP) at the initial position. For the agents with the same HP level, we prioritize them by the DIS approach.
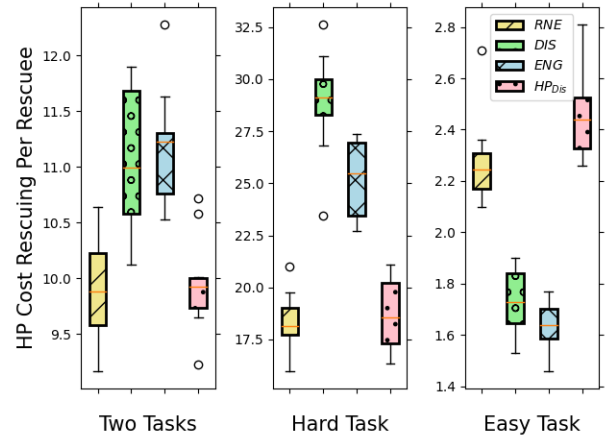
RNE Trust: By calculating the proposed RNE values between agents at the initial position, the whole group classifies as two subgroups with maximum trust level difference between each other. It means that the subgroup gathers agents with similar higher capabilities and needs levels. Then, the
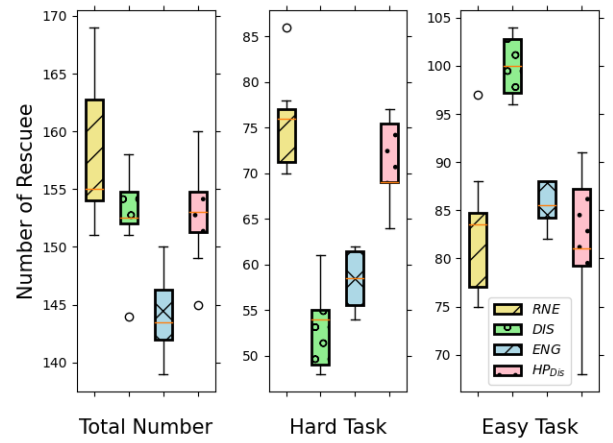
(a) Group Utility (Teaming Needs) Analysis
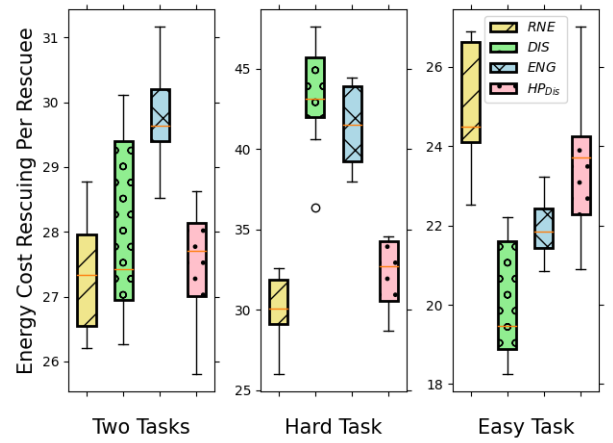
(b) Energy Cost Analysis

(c) Health (HP) Cost Analysis

Fig. 3. Performance comparison of different multi-robot cooperation (grouping) models in USAR missions with two difficulty tasks.

subgroup with higher trust level within the group executes the hard task, and the lower trust group fulfills the easy task.

VII. RESULTS AND DISCUSSION

Figure 3 show the final results of all the methods compared in this study showing all three performance metrics.

The experiment demonstration video is available at our website http://hero.uga.edu/research/trust/. Through Fig. 3(a), we can notice that RNE has the best performance for the total number of rescuees than other methods. More specifically, RNE presents more advantages in the challenging task but inferior results in the easy task, where pure distance-based group formation is highly effective. The RNE trust-based grouping mechanism balances various needs between agents and organizes more reliable and stable groups for achieving the tough or emergent task, and uses the lower trusted agents for easy task.

From the perspective of optimizing the efficiency of system resources, Figs. 3(b) and 3(c) show that RNE essentially decreases the energy and HP cost rescuing per rescuee, especially in the challenging task. The RNE trust model helps the system reassign the resources and gathers agents with similar capabilities, needs, and interests to fulfill the suitable tasks, which improves the system performance and lets each group member's abilities be fully utilized. Like human society, the multi-robot system can make the best possible use of resources and materials based on trust-based cooperation models.

Comparing with the natural intelligent agent, when the artificial intelligence (AI) agent becomes more advanced and smart, it also represents more complex, multilayered, and diverse needs in evolution such as individual security, health, friendship, love, respect, recognition, and so forth. When we consider intelligent agents, like robots, working as a team or cooperating with human beings, organizing their needs building certain reliable and stable relationships such as trust is a precondition for robot-robot and human-robot collaboration in complex and uncertain environments [12].

VIII. CONCLUSION

In this paper, we introduce a general agent trust model based on *Relative Needs Entropy (RNE)* to measure and analyze the trust levels between agents and groups, representing the similarity of their diverse needs in a specific situation for heterogeneous multi-robot cooperation. Then, we illustrate how the RNE trust can be used in multiagent decision-making applications. Specifically, we propose an RNE trust-based effective heterogeneous multi-robot cooperation method to form multiple robot groups based on trust levels within the groups. The proposed model is evaluated through extensive simulations under different difficulty tasks in a post-nuclear radiation leak-like urban search and rescue scenario. We also developed a dynamic priority switching mechanism to solve conflicts in multi-robot cooperation.

The experimental analysis showed that the RNE trust-based grouping model outperformed state-of-the-art energy-based and distance-based methods in maximizing group utilities and represented lower system costs. Trust based on relative needs distributions presents opportunities for improvements and interesting applications. Ultimately, we envision a harmonious team of robots in future multi-robot missions in which each robot values trust on each other robot.

REFERENCES

[1] R. L. Swinth, "The establishment of the trust relationship," *Journal of conflict resolution*, vol. 11, no. 3, pp. 335–344, 1967.

[2] R. J. Lewicki, E. C. Tomlinson, and N. Gillespie, "Models of interpersonal trust development: Theoretical approaches, empirical evidence, and future directions," *Journal of management*, vol. 32, no. 6, pp. 991–1022, 2006.

[3] J.-H. Cho, K. Chan, and S. Adali, "A survey on trust modeling," *ACM Computing Surveys (CSUR)*, vol. 48, no. 2, pp. 1–40, 2015.

[4] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human factors*, vol. 46, no. 1, pp. 50–80, 2004.

[5] J.-H. Cho, A. Swami, and R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE communications surveys & tutorials*, vol. 13, no. 4, pp. 562–583, 2010.

[6] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, pp. 1–33, 2013.

[7] J. Wang, X. Jing, Z. Yan, Y. Fu, W. Pedrycz, and L. T. Yang, "A survey on trust evaluation based on machine learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–36, 2020.

[8] R. Chen, F. Bao, and J. Guo, "Trust-based service management for social internet of things systems," *IEEE transactions on dependable and secure computing*, vol. 13, no. 6, pp. 684–696, 2015.

[9] V. Mohammadi, A. M. Rahmani, A. M. Darwesh, and A. Sahafi, "Trust-based recommendation systems in internet of things: a systematic literature review," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–61, 2019.

[10] H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser, "A survey of multi-agent trust management systems," *IEEE Access*, vol. 1, pp. 35–50, 2013.

[11] K. K. Fullam, T. B. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss, "A specification of the agent reputation and trust (art) testbed: experimentation and competition for trust in agent societies," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005, pp. 512–518.

[12] Q. Yang and R. Parasuraman, "Needs-driven heterogeneous multi-robot cooperation in rescue missions," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2020, pp. 252–259.

[13] Z. R. Khavas, S. R. Ahmadzadeh, and P. Robinette, "Modeling trust in human-robot interaction: A survey," in *International Conference on Social Robotics*. Springer, 2020, pp. 529–541.

[14] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. Chen, E. J. De Visser, and R. Parasuraman, "A meta-analysis of factors affecting trust in human-robot interaction," *Human factors*, vol. 53, no. 5, pp. 517–527, 2011.

[15] T. Balch, "Social entropy: a new metric for learning multi-robot teams," 1997.

[16] Q. Yang, Z. Luo, W. Song, and R. Parasuraman, "Self-reactive planning of multi-robots with dynamic task assignments," in *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2019, pp. 89–91.

[17] Q. Yang and R. Parasuraman, "Hierarchical needs based self-adaptive framework for cooperative multi-robot system," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 2991–2998.

[18] R. Parasuraman, J. Kim, S. Luo, and B.-C. Min, "Multipoint rendezvous in multirobot systems," *IEEE transactions on cybernetics*, vol. 50, no. 1, pp. 310–323, 2018.

[19] T. M. Cover and J. A. Thomas, "Entropy, relative entropy and mutual information," *Elements of information theory*, vol. 2, no. 1, pp. 12–13, 1991.

[20] P. C. Fishburn, "Utility theory for decision making," Research analysis corp McLean VA, Tech. Rep., 1970.

[21] M. J. Kochenderfer, *Decision making under uncertainty: theory and application*. MIT press, 2015.

[22] S. Luo, J. Kim, R. Parasuraman, J. H. Bae, E. T. Matson, and B.-C. Min, "Multi-robot rendezvous based on bearing-aided hierarchical tracking of network topology," *Ad Hoc Networks*, vol. 86, pp. 131–143, 2019.

[23] R. Parasuraman, P. Pagala, K. Kershaw, and M. Ferre, "Energy management module for mobile robots in hostile environments," in *Conference Towards Autonomous Robotic Systems*. Springer, 2012, pp. 430–431.