An Autonomous Task Allocation for Multi-robot System

Limei JIANG†, Rubo ZHANG

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Abstract

This paper presents an approach of dynamic task allocation that enables the robot to automatically perform task in a distributed manner within unknown environments. The proposed cooperative system is based on swarming behaviors inspired by biological colonies. We assume that the lightweight robot is incompetent and needs cooperation from other robots to perform a task. Communication is limited within a certain range, so each robot only communicates task information with its neighbors locally using probabilistic flooding algorithm. The robot associates each task that it is aware of with a utility value rather than pheromone in ACO-based task allocation algorithms. A new task utility function is developed which considers not only the respond threshold but also the task priority, travel cost and updated time. We also propose a concept of task social utility corresponding to task utility which is renamed as personal utility. The relative value is computed through social utility and personal utility which acts as the basis for the robot to select the most favorable task. Our experimental results indicate that our task allocation system performs better compared to other systems.

Keywords: Task Allocation; Swarming; Robotics; Distributed System

1. Introduction

Task allocation is an issue with considerable importance in the field of cooperative robotic system. So far, it is still a challenging topic to design an appropriate mechanism for multi-robot system to assign tasks adaptively in unknown dynamic environments for various applications [1]. Although the centralized task allocation can get a global optimal solution, it also introduces some problems, such as communication jam/delay, poor scalability with the number of robots, poor flexibility to changes, and susceptibility to attacks on the central centre. Market-based approaches are the mainstream with a distributed manner, involving various contract net protocols and auction algorithms [2]. Market-based approaches combine the efficiency of centralized approaches with advantages of distributed approaches, which can results in optimal solution theoretically [3]. At the same time, negotiations between robots make the system depend on communication to a great extent, which is not appropriate for the application where communication is severely constrained, such as multiple autonomous underwater vehicle system. Recently, inspired by the emergent phenomena of ants, bees, and other social insect colonies, more and more researchers have been studying the bio-inspired task allocation approaches in the dynamic unknown environment, including threshold based approaches (which are proposed by observing the labor division and task allocation processes in ant colonies [4]), ACO-based approaches [5][6], PSO-based approaches [7]and other emergent task allocation approaches [8]- [11]. Due to distributed local interaction among lots of relatively simple...
individuals and between individuals and environments, the swarm based system is characterized by robustness, scalability, simplicity and self-organization. However, the distributed system is predestinated to have worse global coordination capability. Conflict and interference will further decrease the overall performance, which are inherent vices for distributed system. Nevertheless, the swarm based techniques still provide an effective solving model.

In this paper, we propose a distributed dynamic task allocation approach that uses *swarming techniques* for multiple targets cleanup application. The lightweight robot is incompetent due to payload limitations and needs cooperation from other robots to perform a task. Communication is also limited and each robot only communicates with its neighbors locally using *probabilistic flooding algorithm* rather than stigmergy in consideration of practical applications. Instead of pheromone generally used in ACO-based algorithms, the paper associates each task with a utility. Social utility corresponding to personal utility is introduced which is produced in the process of information transmission. The paper aims to reduce the interferences between robots, enhance coordination and cooperation, and finally improve the overall efficiency.

The paper is organized as follows. After the motivation of this paper is introduced, a new task allocation approach is proposed in section 2, as well as a deeper analysis. Simulations and results are presented in section 3, followed by a brief discussion. We conclude in section 4 with summary and future directions.

### 2. Cooperative Task Allocation Strategy

In the ACO-based task allocation algorithms, the robot associates each task it is aware of with a certain amount of pheromone $\tau$ to express its affinity for the task and the pheromone will appeal to more other robots in return. In this paper the robot is also supposed to assign each task it is aware of with a number, i.e. utility. Both of them play a similar role from the viewpoint that they both act as the scale of measuring the task importance or appealing degree to robots. In the ACO-based task allocation algorithms, the pheromone value is decayed over time to preserve the temporal ordering of tasks. However, the value of utility does not “evaporates”, but is computed according to a utility function at some intervals or when the robot discovers a new task or receives new message about the task.

#### 2.1. Utility Function

The task information is represented by a six-tuple, $\psi_j = \langle \text{tid}, \text{type}, \text{loc}, \text{taj}, \text{tuj}, \text{naj} \rangle$, representing task ID, type, location, appearing time, last updated time, and the number of robots gathering around the task respectively. The referred parameters are listed as follows. $c_j$ is the fixed priority of task $j$. $nr_j$ is the number of robots needed to complete a task $j$. Both $c_j$ and $nr_j$ are related to the type of the task. $dl_j$ is the duration of lasting request of task $j$, $dl_j = t - t_{aj}$, where $t$ is the current time. The information of task $j$ stored in robot $i$ is marked by double subscript, e.g., $naij$ is the number of robots at task $j$ recorded by robot $i$, $\theta_{ij}$ is the response threshold of robot $i$ to task $j$, $d_{ij}$ is the distance from robot $i$ to task $j$.

\[
\text{utility}_{ij} = \text{conf}_{ij} \times \left( u_1 \times \text{fam}_{ij} + u_2 \times \text{pri}_{ij} + u_3 \times \text{dis}_{ij} \right). \tag{1}
\]

The utility of robot $i$ to task $j$ is defined by (1), where, $u_1$, $u_2$, and $u_3$ are weight coefficients. The utility function consists of four parts listed as follows.

1) Specialization is an appealing feature through which the insect individuals are allocated to different tasks adaptively and flexibly, resulting in an improvement of their fitness to the environments. So the
familiarity of robot \( i \) to task \( j \) is defined by (2) as \( \text{fami}_{ij} \). The response threshold \( \theta_{ij} \) is updated as the same as in [13]. The more the times that robot \( i \) complete task \( j \) and other tasks which belong to the same type as task \( j \), the lower the \( \theta_{ij} \) is. It also means that robot \( i \) becomes more familiar to the task of this type. The parameter \( k \) is used to adjust the effect of \( \theta_{ij} \).

\[
\text{fami}_{ij} = k \left( \frac{1}{k + \theta_{ij}} \right).
\] (2)

2) The priority of tasks must be considered if some tasks must be completed preceding to other tasks or some tasks are more emergent than others. The priority of task \( j \) represented by \( \text{pri}_j \) is defined by (3) varying from 0 to 1. The priority contains the fixed priority, which is defined in advance, and the dynamic priority, which varies over time continuously. Supposing there are \( n \) kinds of tasks, the fixed priority in this paper is divided into \( n \) levels uniformly between 0.1 and 0.9. The parameter \( \sigma \) adjusts the priority how to increase with \( t_{lj} \). The configuration of the priority of tasks coincides with the purpose of real-time application.

\[
\text{pri}_j = \frac{1}{2} \left( c_j + \min \left\{ \sigma \times t_{lj}, 1 \right\} \right).
\] (3)

3) It is common sense that the robot inclines to perform a nearby task rather than a farther one in consideration of travel cost and time. The \( \text{dis}_{ij} \) is the fitness of distance from robot \( i \) to task \( j \) defined by (4).

\[
\text{dis}_{ij} = d_{ij}^{\frac{1}{2}}.
\] (4)

4) In the ACO-based algorithm, task information is updated through pheromone evaporation. In this paper, we define the variable \( \text{conf}_{ij} \) to enable the robot estimate the task latest state. The confidence of robot \( i \) about whether task \( j \) is unfinished is computed by (5). The value \( \text{conf}_{ij} \) decreases with the elapsed time because robots \( i \) may consider that other robots have completed the task \( j \). \( 0 < \text{conf}_{ij} \leq 1 \). The change of \( \text{conf}_{ij} \) is show in figure 1.

\[
\text{conf}_{ij} = \exp \left( -\left( t - t_{ij} \right)^2 / 2 \left( \text{nr}_j - \text{na}_j \right)^2 \right).
\] (5)

Fig.1 It Shows Three Curves for \((\text{nr-na})=1, 2, 3\) Respectively and Time is an Independent Variable for Each Curve

2.2. Communication Mechanism and Information Fusion

The system using implicit communication can not complete higher coordination and cooperation without explicitly exchange of data and information. The multi-robot system in this paper uses explicit communication but only in local short range. The gossip-based mechanism is a local interaction with
decentralized control that can replace implicit communication [10][11]. In this paper, each robot employs a gossip mechanism to exchange information with other robots. The paper discusses the process in details from the aspects of sending and receiving.

The messages are categorized into two types: the messages about the tasks to be done and the messages about the tasks have been done, marked by TBD and ABD respectively. The format of message is shown in figure 2, where rid is the id of sender, TBD/ABD is the type of message, ψj is the information of task j, hop is the number of hops the message containing information about ψj before reaching robot i, and utilityj is the average utility calculated by the robots who have received this message.

![Fig.2 The Format of TBD and ABD Messages](image)

All the senders set up a sending message queue. They broadcast messages in the queue at intervals and need not to confirm whether the receivers can receive or not. The probabilistic flooding algorithm is used to forward a TBD message. The probability of forwarding a TBD message by a robot decreases with the distance from the message source. This results in that the probability of forwarding a TBD message by the robots near the message source is higher than other robots. probij is the probability of robot i to forward a message about task j.

\[
prob_j = 1 - \frac{\text{hop}}{H},
\]

where \( H \) is the maximum of hops.

For the ABD message, the probability of forwarding is 1 if \( \text{hop} < H \).

All the receivers create a received message queue. The receiver only accepts messages of interest, and decides whether or not to abandon the new received message. For the TBD messages, the receiver checks up task lists to see if it has already recorded the message about ψj by comparing tid. If so, the receiver updates the information about ψj. Otherwise, the receivers add the information about ψj to task list. Let \( \text{hop} \leftarrow \text{hop} + 1 \), and calculate the social utility of task j, that is \( \text{utility}_j \), by

\[
\text{utility}_j \leftarrow \left( \text{utility}_j \times \text{hop} + \text{utility}_j \right) / (\text{hop} + 1).
\]

The reason we name \( \text{utility}_j \) as the social utility is that \( \text{utility}_j \) is propagated in the robot swarm reflecting the robot society’s estimation to task j.

For the ABD message, the receiver checks up the task list to see if they have recorded the message about ψj. If so, delete it. Otherwise, do nothing with it.

2.3. Task Selection

The collision occurs while multiple robots compete for the same resource. Therefore, there is a balance between information sharing and collision avoidance [12]. From a different perspective, the selfish robot is
inclined to choose the task with larger utility for them. If the robot takes the state of other robots into account while making decision, the collisions will be reduced.

Corresponding to the social utility $utility_j$, we rename the utility $utility_{ij}$ computed by (1) as personal utility. The relative value is defined by

$$
\gamma_{ij} = \frac{utility_{ij}}{utility_j}.
$$

The centralized algorithm takes advantage of global information to get an optimum solution. In this paper, $utility_j$ is an average utility calculated in the process of message propagation which integrates the evaluation by other robots. $\gamma_{ij}$ is the relative advantage of robot $i$ to task $j$, which is called relative utility. $\gamma_{ij}$ utilizes the information of other robots in order to make decision more comprehensively.

The roulette method is used to select task based on $\gamma_{ij}$, which also expresses the internal competition in a group. Suppose $\gamma_i$ is the sum of all the $\gamma_{ij}$ recorded by robot $i$, the process is: 1) Let $sum \leftarrow 0$; 2) Generate a random number between 0 and $\gamma_i$, noted by $rand$; 3) Let $j \leftarrow 0$; 4) $sum \leftarrow sum + \gamma_{ij}$; 5) if $sum > rand$, task $j$ is selected, end; otherwise, turn to step 4).

Nevertheless, collisions are still inevitable in a distributed system. Collisions are solved by comparing each robot’s intensity. Intensity of robot $i$ is defined as the sum of tasks enlisted by robot $i$. The smaller the intensity is, the greater the probability of selecting task $j$ is. Deadlock is an extreme situation of conflict, where each robot insists on a task that is beyond its ability. So we set an upper limit of waiting time $T_w$. If a robot’s waiting time at a task for cooperation is over $T_w$, it will transfer to other state, searching or re-selecting.

3. Simulation Results

This section presents the quantitative evaluations of our task allocation strategy for a distributed targets cleanup application in an unknown environment. Robots were required to localize and deal with targets while they had no knowledge of the environment and the distribution of targets in advance. A single robot could discover and recognize a target but it could not deal with the targets individually due to limited capacity. The robot should cooperate with other robots to execute on a discovered target. In this paper, we concern with the task allocation mechanism, so we do not care about searching process, obstacle avoidance, and executing process.

Three metrics are employed to measure the system performance: $V-t$, the time required to finish all tasks; $V-r$, the number of goal switches due to redundant selection or timeout when waiting for cooperation; $V-w$, the time elapsed while waiting for cooperation. The parameters are set up as follows.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$k$</th>
<th>$\theta_{max}$</th>
<th>$\theta_{min}$</th>
<th>$H$</th>
<th>$T_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>20</td>
<td>0</td>
<td>100</td>
<td>20</td>
<td>25</td>
</tr>
</tbody>
</table>

The first scenario consisted of two types of targets, 10 for each type, distributed randomly inside a $50 \times 50$ environment, and 20 robots in the corner at beginning. 2 and 5 robots were required for each type task respectively. The sensing range of each robot is 2.5 and the communication range 7.5. The response threshold of all the robots were the same initially, equal to $k$. The 2-types task allocation in this paper can
be extended to \( n \)-type task allocation. Four cases were tested to validate the reasonableness of the proposed utility function. 1) \( u_1, u_2, u_3=1 \); 2) \( \text{conf}=1, u_1, u_2, u_3=1 \); 3) \( u_1=0, u_2, u_3=1 \); 4) \( \text{conf}=1, u_1=0, u_2, u_3=1 \). Here, we do not test the task priority and travel cost for simplicity, because it is common sense the robot inclines to choose the task with higher priority and low travel cost. Table 2 shows the averaging results obtained through 50 Monte Carlo simulations for each case. It is obvious that the performance of case 1 outperforms the others. Comparing case 1 with case 2, we infer that the information about the progress of the task plays a positive role which enables robots use updated information to make more precise decision. Comparing case 1 with case 3, we infer that the specialization based on respond threshold results in less redundant selection greatly. In all, every part of utility function is reasonable.

<table>
<thead>
<tr>
<th>Case</th>
<th>( V_r )</th>
<th>( V_w )</th>
<th>( V_t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case1</td>
<td>5.2</td>
<td>30.8</td>
<td>300</td>
</tr>
<tr>
<td>Case2</td>
<td>5.6</td>
<td>37.1</td>
<td>344</td>
</tr>
<tr>
<td>Case3</td>
<td>5.8</td>
<td>34.8</td>
<td>355</td>
</tr>
<tr>
<td>Case4</td>
<td>6.1</td>
<td>39.7</td>
<td>360</td>
</tr>
</tbody>
</table>

Next, we test the advantage of social utility and task selection based on relative value. We compared the proposed strategy with the one using task selection based on personal utility only. Consider the case where \( u_1, u_2, u_3=1 \), the results of the later are 5.7, 37.7, 375 in turn, which are inferior to the ones proposed in this paper.

Finally, we test if the proposed strategy still performs well while the number of robots increases by comparing it with a random method. The number of robots varies from 20 to 30 and 60. The results are shown in table 3. From table 3, the performance of the proposed strategy is always superior to the random method when the number of robots increases, so the proposed strategy shows a good scalability.

<table>
<thead>
<tr>
<th>StrategiesA</th>
<th>StrategiesB</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_r )</td>
<td>5.2</td>
</tr>
<tr>
<td>( V_w )</td>
<td>30.8</td>
</tr>
<tr>
<td>( V_t )</td>
<td>300</td>
</tr>
<tr>
<td>( V_r )</td>
<td>7.3</td>
</tr>
<tr>
<td>( V_w )</td>
<td>11.6</td>
</tr>
<tr>
<td>( V_t )</td>
<td>230</td>
</tr>
<tr>
<td>( V_r )</td>
<td>7.8</td>
</tr>
<tr>
<td>( V_w )</td>
<td>2.4</td>
</tr>
<tr>
<td>( V_t )</td>
<td>126</td>
</tr>
</tbody>
</table>

### 4. Conclusion

In this paper, we describe a dynamic task allocation strategy based on the swarming behavior for multiple
targets cleanup application. We apply the concepts of personal utility and social utility, which produce relative utility, to compose the basis of task allocation. The gossip mechanism based on the probabilistic flooding algorithm is used to exchange information. Experimental results show that the proposed strategy allows robots to perform task efficiently with a good scalability. Nevertheless, some problems still need further study, for example, evaluation criteria, strict theory research, and communication constraints.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China under Grant no. 60975071. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References