A Dynamic Coordination Approach for Task Allocation in Disaster Environments under Spatial and Communicational Constraints

Xing Su, Minjie Zhang, Dayong Ye
School of Computer Science & Software Engineering
University of Wollongong, Australia
xs702@uowmail.edu.au
{dayong, minjie}@uow.edu.au

Quan Bai
School of Computing & Mathematical Sciences
Auckland University of Technology
New Zealand
quan.bai@aut.ac.nz

Abstract

Dynamic coordination for task allocation in disaster environments under spatial and communicational constraints is a challenging issue in both research and applications. To this end, this paper presents a coordinated task allocation approach for disaster environments by considering spatial and communicational constraints, dynamic features of environments as well as heterogenous capabilities of agents. The proposed approach consists of an information collection mechanism, a group task allocation mechanism and a group coordination mechanism. Initially, the information collection mechanism is applied to help agents in communication networks to prune their communication connections and elect one agent in each communication network to be the network leader in a decentralised manner so as to facilitate the network leader to collect information for task allocation under communicational constraints. Then, the group task allocation mechanism is employed by each network leader to allocate tasks and agents in its network to groups with suitable spatial ranges by considering spatial and communicational constraints and heterogenous capabilities of agents. During task execution, the group coordination mechanism is employed by isolated groups to periodically adjust group members (agents) at assembly points so as to achieve continuous coordination to handle dynamic features of environments. Experimental results demonstrate that the proposed approach can have better performance than some existing approaches in terms of information collection and coordination for task allocation in disaster environments under spatial and communicational constraints.

1 Introduction

Nowadays, coordination for task allocation techniques have been widely applied to many domains such as disaster rescue, space exploration and distributed computing, etc (Lesser 1999; Allouche and Boukhtouta 2010; Wu et al. 2011). In disaster environments, coordination for task allocation faces several challenging issues especially in the aspects of: **1) Spatial constraints**. In disaster environments, stationary tasks can be discovered at different locations if an agent wants to work on a task, it first needs to move to the location of the task (Barbulescu et al. 2010); **2) Communicational constraints**. In disaster environments, due to the destruction of local communication facilities, the amount of information transferred among agents (i.e., the constraint of communication capacities) and the communication distances of agents (i.e., the constraint of communication ranges) are limited (Ramchurn et al. 2010a). **3) Dynamic features of environments**. In disaster environments, agents can enter and leave the environments and tasks can be continuously discovered and finished in the environments (Smith, Gallagher, and Zimmerman 2007; Chapman et al. 2009); **4) Heterogenous capabilities of agents**. In disaster environments, agents are heterogenous and each agent has its own capabilities, which determine what kind of tasks it can conduct (Koes, Nourbakhsh, and Sycara 2005). **5) Local views of agents**. Due to the communicational constraints of disaster environments, each agent can only acquire the information of tasks nearby its location and the information of agents with which it can directly communicate (Reich 2006);

Various approaches have been proposed to achieve efficient task allocation in disaster environments from different perspectives (Musliner and Goldman 2006; Smith, Gallagher, and Zimmerman 2007). Some approaches deal with task allocation through a central point (Koes, Nourbakhsh, and Sycara 2005; Ramchurn et al. 2010b). In such approaches, the central controller (i.e., the agent in charge of task allocation) can create optimal solution for task allocation based on the global view about the environment. However, the main limitation for the application of centralised approaches in disaster environments is that communicational constraints make the central controller hard to timely collect information in the working environment.

In order to overcome the limitation brought by communicational constraints, some approaches enable agents to make decisions for task allocation by themselves based on their local views about the environment (Farinelli et al. 2008; Ramchurn et al. 2010a). In order to enlarge local views of agents, the max-sum algorithm is employed by decentralised approaches for message passing to enable agents to exchange information with each other based on the communication connections. However, in order for agents to collect comprehensive information to create optimal solutions for task allocation, agents need to exchange information with

all their direct neighbours, which needs a plenty of time to pass a large amount of information before task allocation and cannot suit the constraint of communication capacities as well as dynamic features of disaster environments. In addition, the prerequisite of employing the max-sum algorithm for message passing is that there must be communication paths among agents. If two agents are isolated (i.e., there is no communication path between two agents), the max-sum algorithm cannot achieve information exchange between them under communicational constraints.

In general, current approaches have the following drawbacks: 1) most centralised approaches cannot achieve timely information collection for task allocation under communicational constraints; 2) most decentralised approaches are hard to handle the dynamic features of disaster environments; 3) most of centralised and decentralised approaches cannot handle all spatial and communicational constraints, dynamic features of disaster environments as well as heterogenous capabilities of agents; and 4) most of centralised and decentralised approaches cannot achieve information exchange between isolated agents under communicational constraints.

To overcome the above drawbacks, this paper presents a dynamic coordination approach for task allocation in disaster environments under spatial and communicational constraints. The proposed approach includes the following mechanisms: **1) An information collection mechanism** is applied to help agents in communication networks to prune their communication connections and elect one agent for each communication network to be the network leader in a decentralised manner; **2) A group task allocation mechanism** is employed by each network leader to allocate tasks and agents in its network to groups with suitable spatial ranges; and **3) A group coordination mechanism** is employed by isolated groups to periodically adjust group members (agents) at assembly points.

The merits of the proposed approach include that 1) the proposed approach can reduce the overhead information exchange for information collection so as to handle communicational constraints of disaster environments; 2) the proposed approach can help the network leader to allocate tasks and agents to groups with suitable spatial ranges so as to handle spatial and communicational constraints of disaster environments and heterogenous capabilities of agents; 3) the proposed approach can achieve continuous coordination of isolated groups in the environment so as to handle dynamic features of disaster environments.

The rest of the paper is organized as follows. The problem is described and definitions are given in Section 2. The basic principle of the proposed approach is introduced in Section 3. The experiments are given and the results are analyzed in Section 4. The related work is introduced in Section 5. The paper is concluded and the future work is outlined in Section 6.

2 Problem Description and Definitions

In general, a task allocation problem includes a set of agents, which can be described as $\{A_1, A_2, ..., A_i, ..., A_m\}$, where A_i represents an agent in the set. A task can only be discovered by the agents which are close to the location of it. A task is represented by $T_{(i,j)}$, which means the j^{th} task discovered by A_i. The definitions of an agent and a task are given as follows.

Definition 1: An *Agent* (A_i) is defined as a two-tuple A_i=<Loc_i, $\vec{Cap}_i$>, where Loc_i is the current location of A_i; and $\vec{Cap}_i$ is the capabilities of A_i, which is described as a vector $\vec{Cap}_i$=(c_i^1, c_i^2, ..., c_i^R), where c_i^r is the indicator of the r^{th} capability, which indicates whether A_i has the r^{th} capability, if A_i has the r^{th} capability, $c_i^r = 1$, otherwise, $c_i^r = 0$.

Definition 2: A *Task* ($T_{(i,j)}$) represents the j^{th} task discovered by A_i, which is defined as a two-tuple $T_{(i,j)}$=<$Loc_{(i,j)}$, $\vec{RCap}_{(i,j)}$>, where $Loc_{(i,j)}$ is the location of $T_{(i,j)}$; and $\vec{RCap}_{(i,j)}$ is the required capabilities of $T_{(i,j)}$, which is described as a vector $\vec{RCap}_{(i,j)}$=($c^1_{(i,j)}$, $c^2_{(i,j)}$, ..., $c^R_{(i,j)}$), where $c^r_{(i,j)}$ is the indicator of the r^{th} capability, which indicates whether $T_{(i,j)}$ requires the r^{th} capability to finish, if $T_{(i,j)}$ requires the r^{th} capability to finish, $c^r_{(i,j)} = 1$, otherwise, $c^r_{(i,j)} = 0$.

The objective of coordination for task allocation in disaster environments is to maximize the total number of finished tasks, which can be described as follow.

$$Objective = max \sum_{\forall T_{(i,j)}} Finish(T_{(i,j)}), \qquad (1)$$

where $Finish(T_{(i,j)})$ is a Boolean-return function, if $T_{(i,j)}$ is finished, $Finish(T_{(i,j)})$=1.

In the proposed approach, tasks and agents are allocated to groups, each of which includes a set of tasks and agents and each task and agent can only belong to only one group. The definition of the group information of a group is given as follow.

Definition 3: The *Group Information* ($GInf_k$) of a group (G_k) is defined as a five-tuple $GInf_k$=<$TSet_k$, $UTSet_k$, $ASet_k$, $IASet_k$, Rep_k>, where $TSet_k$ is the set of tasks of G_k; $UTSet_k$ is the set of unfinished tasks of G_k, where $UTSet_k \subseteq TSet_k$; $ASet_k$ is the set of agents of G_k; $IASet_k$ is the set of idle agents of G_k, where $IASet_k \subseteq ASet_k$; and Rep_k is the representative agent of G_k, which is in charge of coordination with other isolated groups.

In the proposed approach, groups periodically coordinate with each other at assembly points and one round of coordination is finished in TP time units. In the proposed approach, there are two kinds of coordination, which are the top-layer coordination and the bottom-layer coordination carried out at two kinds of assembly points: the *assembly point of the environment* and the *assembly points of the network*, the definitions of which are given as follows.

Definition 4: The *assembly point of the environment* (APe) is defined as the only location in a disaster environment, which is set for the top-layer coordination beforehand and can be well known by agents in the environment.

Definition 5: An *assembly point of the network* (APn_p) is

defined as a location in a disaster environment, which is set during emergency for the bottom-layer coordination of groups from the same communication network.

3 The Basic Principle of the Proposed Approach

The proposed approach consists of three mechanisms, which are 1) the information collection mechanism; 2) the group task allocation mechanism; and 3) the group coordination mechanism. The basic principle of the proposed approach is shown in Figure 1.

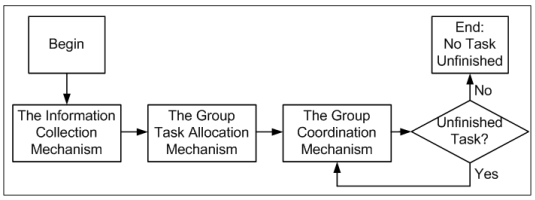

Figure 1: The basic principle of the proposed approach

At the beginning, the information collection mechanism can help agents in communication networks to prune their communication connections and elect a network leader for each network, which is in charge of information collection and task allocation in its network. Based on the pruned communication connections, agents can pass the information of their nearby tasks and the status of themselves to the network leader. After that, according to the group task allocation mechanism, each network leader allocates tasks and agents in its network to groups with suitable spatial ranges according to the locations of tasks and capabilities of agents. The information collection mechanism and the group task allocation mechanism are performed for just once and network leaders are dismissed after the group task allocation mechanism. During task execution, due to dynamic features of environments, the original allocation (by the group task allocation mechanism) of tasks and agents may be unsuitable. Therefore, the group coordination mechanism is periodically (in every TP time units) employed by groups to adjust group members (agents) at assembly points so as to achieve continuous coordination for task allocation until there is no unfinished task in the environment.

3.1 The Information Collection Mechanism

The objective of the information collection mechanism is to help agents to prune their communication connections and elect a network leader to collect information for task allocation in a decentralised manner, during which agents in the communication network eliminate all their communication connections except the one leading to the network leader. Therefore, after applying the information collection mechanism for an m number of agents communication network, only $m-1$ number of connections are kept in the network. By doing so, each agent only needs to pass the information it has to the network leader through its only direct neighbour so as to reduce the overhead information exchange for task allocation.

In the information collection mechanism, three neighbour related parameters are defined for each agent, which are the parent agent (represented by PA), the network leader (represented by L) and the number of direct neighbours of the network leader (represented by NNL). For example, for an agent A_i, three neighbour related parameters can be denoted as $A_i.PA$, $A_i.L$ and $A_i.NNL$, respectively. The information collection mechanism is described in *Algorithm 1*.

Algorithm 1: The information collection mechanism

1 **for *each*** *agent (e.g., A_i)* **do**
2 $A_i.PA \leftarrow A_i$; $A_i.L \leftarrow A_i$; $A_i.NNL \leftarrow$ the number of direct neighbours of A_i
3 Broadcasts its three neighbour related parameters
4 **for *each*** *agent (e.g., A_i)* **do**
5 Gets A_u from its received parameters, where $A_u.NNL$ is maximum
6 **if** $A_u.NNL > A_i.NNL$ **then**
7 $A_i.PA \leftarrow A_u$; $A_i.L \leftarrow A_u.L$; $A_i.NNL \leftarrow A_u.NNL$
8 Broadcasts its new neighbour related parameters

At the beginning of Algorithm 1, three neighbour related parameters of each agent (e.g., A_i) are initialised as follow: $A_i.PA$ is set to A_i, $A_i.L$ is set to A_i and $A_i.NNL$ is set to the number of direct neighbours of A_i (Lines 1 and 2). Then, agents broadcast their three neighbour related parameters to its direct neighbours (Line 3). When an agent (e.g., A_i) received parameters from its direct neighbours, it repeats the following two steps. **Step 1:** A_i finds the agent (e.g., A_u) with the highest value of the parameter NNL (Lines 4 and 5); **Step 2:** If $A_u.NNL$ is higher than the value of $A_i.NNL$, three neighbour related parameters of A_i are updated as follow: $A_i.PA$ is set to A_u, $A_i.L$ is set to $A_u.L$ and $A_i.NNL$ is set to $A_u.NNL$ (Lines 6 and 7) and broadcast its updated three neighbour related parameters to its direct neighbours (Line 8); The above two steps (Lines 4 to 8) will be repeated by each agent until no further updating for three neighbour related parameters of any agent.

In the final stage of the information collection mechanism, each connected agent (e.g., A_i) can pass the information of its nearby tasks and status of itself to its network leader (i.e., $A_i.L$) through its parent agent (i.e., $A_i.PA$) set in the mechanism. In addition, since a task can be discovered by multiple agents, agents need to identify superfluous information of tasks and abandon that information during information passing. In addition, if there are isolated agents in an environment, more than one communication networks exists in the environment and more than one network leaders are elected according to the information collection mechanism.

3.2 The Group Task Allocation Mechanism

The group task allocation mechanism helps each network leader to allocate tasks and agents in its network to groups with suitable spatial ranges under the consideration of task

allocation and execution. The group task allocation mechanism is executed by each network leader, which includes three steps: 1) task allocation, 2) agent allocation and 3) assembly point of the network setting. After the group task allocation mechanism, network leaders are dismissed.

Task allocation In task allocation step, tasks of each network are allocated to groups with suitable spatial ranges according to locations of tasks (i.e., $Loc_{(i,j)}$, see Definition 2) and communication ranges of agents (i.e., CR). The objective of this step is that though restricting the spatial range of each group, the moving ranges of agents allocated to each group can be reduced, which can ensure that agents working in the same group can always communicate with each other during task execution. By doing so, within each group, the centralised task allocation approaches can be employed by agents, such as (Koes, Nourbakhsh, and Sycara 2005; Ramchurn et al. 2010b).

To achieve above objectives, the mean-shift algorithm (Comaniciu and Meer 2002) is employed by each network leader to allocate tasks in its network to groups. The only parameter of the mean-shift algorithm h represents the radius of the window, which decides the spatial ranges of groups. In order to enable agents working in the same group to always communicate with each other during task execution, h is set equal to CR. For an n tasks grouping problem in a 2-dimensional space, the multi-kernel density function can be calculated as follow.

$$f(x) = \frac{1}{n \cdot CR^2} \sum_{\forall T_{(i,j)} \in window} K(\frac{x - Loc_{(i,j)}}{CR}), \quad (2)$$

where $K(x)$ is the kernel function, $Loc_{(i,j)}$ is the location of $T_{(i,j)}$ within the window, x is the centre (mean) of a window. In the proposed approach, $K(x)$ can be described by the Euclidean distance between two locations. Based on the multi-kernel density function, the centre (mean) of the window always moves to the point with the greatest density value.

Agent allocation In agent allocation step, agents of each network are allocated to suitable groups according to missing required capabilities of tasks of each groups (e.g., G_k) and capabilities of each unallocated agent (e.g., A_u). In order to find missing required capabilities of tasks of G_k, we first find required capabilities of tasks (i.e., $T_{(i,j)} \in G_k$) and capabilities of allocated agents (i.e., $A_i \in G_k$) of G_k, which can be calculated as follow.

$$\vec{RCap_k} = \sum_{T_{(i,j)} \in G_k} \vec{RCap_{(i,j)}}, \quad \vec{ACap_k} = \sum_{A_i \in G_k} \vec{Cap_i}, \quad (3)$$

where $\vec{RCap_{(i,j)}}$ is the vector of required capabilities of $T_{(i,j)}$ in G_k; and $\vec{Cap_i}$ is the vector of capabilities of A_i in G_k;

Then, the vector of missing required capabilities of tasks of G_k can be calculated as follow:

$$\vec{MRCap_k} = norm(\vec{RCap_k}) - norm(\vec{ACap_k}) \quad (4)$$

where $norm(\vec{RCap_k})$ and $norm(\vec{ACap_k})$ are the normalised vector of required capabilities of tasks and the normalised vector of capabilities of allocated agents of G_k, respectively; and mc_k^r is the indicator of the r^{th} capability, which describes to what extent G_k requires the r^{th} capability, if $mc_k^r \leq 0$, $mc_k^r = 0$, otherwise, $mc_k^r = mc_k^r$.

Finally, the similarity value between the vector of missing required capabilities of tasks of G_k (i.e., $\vec{MRCap_k}$) and the normalised vector of capabilities of an unallocated agent A_u (i.e., $norm(\vec{Cap_u})$=(nc_u^1, nc_u^2, ..., nc_u^R)) can be calculated by the *dot product* of two vectors (Kang and Müller 2011) and A_u will be allocated to the group with the highest similarity value.

Assembly point of the network setting At the final step of the group task allocation mechanism, the assembly point of the network (i.e., APn_p, see Definition 5) is set for the coordination of groups from the same communication network. In order for representative agents of the groups to arrive at APn_p at the same time, the location of APn_p will be set at the centroid (Altshiller-Court 2007) of centres (means) of the groups.

3.3 The Group Coordination Mechanism

During task execution, due to dynamic features of environments, the original allocation (by the group task allocation mechanism) of tasks and agents in groups may be unsuitable, where agents in some groups finish all tasks and are idle, while agents in some groups are busily working on unfinished tasks. In addition, it is hard to employ the existing centralised or decentralised approach to coordinate these groups, since most of groups are isolated with other groups under communication constraints. To dynamically adjust group members (agents) among isolated groups under communicational constraints, during task execution, the group coordination mechanism is periodically executed by the representative agent (i.e., Rep_k, see Definition 3) of each group. One round of coordination can be finished in TP time units and includes following steps

1. The representative agent departure: The representative agent (e.g., Rep_k) of each group (i.e., G_k) takes the latest group information (i.e., $GInf_k$, see Definition 3) and begins to move to the assembly point of the network (APn_p, see Definition 5).
2. The representative agent wait: Since different representative agents might have different moving speed or different distances to APn_p, if Rep_k arrives at APn_p earlier, it needs to wait for the representative agents that do not arrive until the $\frac{TP}{2}$ time units.
3. The representative agent coordination: The representative agents at APn_p begins to coordinate with each other and updates their $GInf_k$ accordingly. The detailed process of coordination will be introduced in Subsection 3.3.

4. The representative agent return: Rep_k returns to G_k and adjusts the work of its group members according to the updated $GInf_k$.

The two-layer coordination The group coordination mechanism has a two-layer structure: the top-layer coordination and the bottom-layer coordination. **Bottom-layer coordination:** the representative agents (i.e., Rep_k, see Definition 3) of groups from the same communication network first coordinate with each other at the assembly point of the network (i.e., APn_p, see Definition 5); **Top-layer coordination:** after coordination at APn_p, if one Rep_k at each APn_p collects all group information (i.e., $GInf_k$, see Definition 3), moves to the assembly point of the environment (i.e., APe, see Definition 4) and coordinates with representative agents from other assembly points of the network.

An example of the two-layer group coordination mechanism is shown as Figure 2

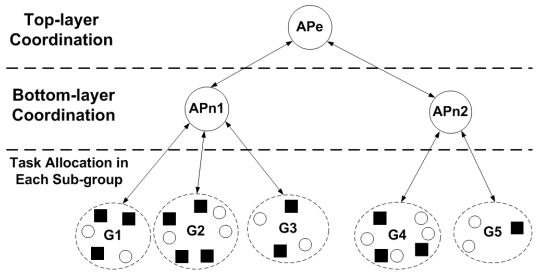

Figure 2: The two-layer group coordination mechanism

In Figure 2, black squares and white circles represent tasks and agents in an environment, respectively. APe is the assembly point of the environment (see Definition 4). There are five groups (i.e., G_1 to G_5) in the environment, where G_1, G_2, and G_3 are from the same communication network and their assembly point of the network is APn_1 (see Definition 5); G_4 and G_5 are from the same communication network and their assembly point of the network is APn_2 (see Definition 5). At beginning, representative agents (i.e., Rep_1 to Rep_3) of G_1 to $G3$ move to APn_1 and coordinate with each other there. At the same time representative agents (i.e., Rep_4 and Rep_5) of G_4 to G_5 move to APn_2 and coordinate with each other there (i.e., bottom-layer coordination). After that, if there are still unfinished tasks or idle agents in G_1 to G_3 and G_4 to G_5, one representative agent in each assembly point of the network (e.g., Rep_1 from APn_1 and Rep_4 from APn_2) move to APe and coordinate with each other (i.e., top-layer coordination).

Adjustment at an assembly point The adjustment of group members (agents) is to allocate suitable idle agents in some groups to unfinished tasks in other groups. The suitability of an idle agent (e.g., A_i) allocated to an unfinished task (e.g., $T_{(i,j)}$) can be evaluated by the similarity value between the vector of the required capabilities of the unfinished task (i.e., $\vec{RCap}_{(i,j)}$, see Definition 2) and the vector of the capabilities of the idle agent (i.e., $\vec{Cap}_i$, see Definition 1), which can be calculated by the *dot product* of two vectors. At the assembly point, many task allocation mechanisms can be employed by agents to adjust group members (agents). One of simply and quick ways is the Contract-Net Protocol (Smith 1980), which is described in *Algorithm 2*.

Algorithm 2: Adjustment at an assembly point

```
1  for each Rep_k whose UTSet_k ≠ ∅ do
2      Broadcasts each T_(i,j) ∈ UTSet_k
3  for each Rep_u whose IASet_u ≠ ∅ do
4      for each received T_(i,j) do
5          for each A_i ∈ IASet_u do
6              Calculates Sim_((i,j),i) between RCap_(i,j)
               of T_(i,j) and Cap_i of A_i
7          Records A_i with Max(Sim_((i,j),i)) in RRes_u
8      Sends RRes_u to Rep_k.
9  for each Rep_k whose UTSet_k ≠ ∅ do
10     for each T_(i,j) ∈ UTSet_k do
11         finds suitable A_i in RRes_u and informs Rep_u
12         Rep_k and Rep_u updates GInf_k and GInf_u,
           respectively.
```

At the initial stage, the representative agent (e.g., Rep_k) of each group (e.g., G_k) with unfinished tasks (i.e., $UTSet_k \neq \varnothing$, see Definition 3) broadcasts its unfinished tasks (i.e., $\forall T_{(i,j)} \in UTSet_k$) to other representative agents at the assembly point (Lines 1 and 2). When a representative agent (e.g., Rep_u) of a group (e.g., G_u) receives unfinished tasks (e.g., $T_{(i,j)} \in UTSet_k$) and G_u has idle agents (i.e., $A_i \in IASet_u$), Rep_u calculates the similarity value (i.e., $Sim_{((i,j),i)}$) between the required capabilities of the received $T_{(i,j)}$ and the capabilities of idle A_i (Lines 3 to 6). For each $T_{(i,j)}$ records A_i with the highest $Sim_{((i,j),i)}$ in resource response $RRes_u$, which is sent back to Rep_k (Lines 7 to 8). After receiving all resource responses, Rep_k chooses suitable A_i for each $T_{(i,j)}$ and informs the owner (i.e., Rep_u) of each A_i (Lines 9 to 11). Rep_k and Rep_u update $Glnf_k$ and $GInf_u$, respectively (Line 12).

4 Empirical Experiments and Analysis

The purpose of this experiment is to evaluate the performance of the proposed approach on task allocation in disaster environments. In this experiment, the proposed approach is compared with the MILP based approach (Koes, Nourbakhsh, and Sycara 2005) and max-sum based approach (Ramchurn et al. 2010a).

4.1 Experimental Settings

In this experiment, 100 tasks (40 tasks are discovered at the beginning of the experiment and 60 are discovered during task execution) and 15 agents are employed in a 50×50 (square units of distance) area. Each agent can only finish move $1 \sim 5$ units of distance per time unit. There are four kinds of capabilities required by tasks. Each task requires one or two kinds of capabilities and each agent has two random kinds of capabilities. The MILP based approach (represented by 'MILP') is a centralised task allo-

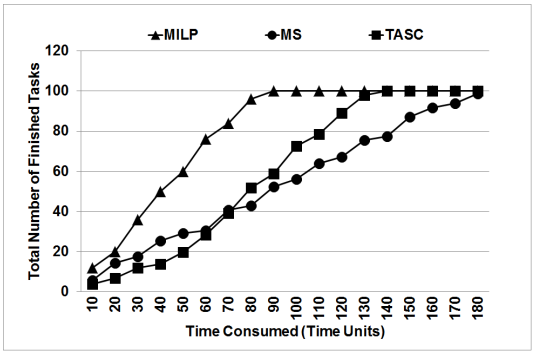

Figure 3: The experimental results of experiment

cation approach without considering communicational constraints. Therefore, in the experiment, we assume that there is no communicational constraints for 'MILP'. For the max-sum based approach (represented by 'MS') and the proposed approach (represented by 'TASC'), the communication ranges of agents (i.e., CR) are fixed to 20 units of distance. In 'TASC', one agent in each group is chosen to be the representative agent, which periodically (every 25 time units) and moves to assembly points to coordinate with representative agents of other groups.

4.2 Experimental Results and Analysis

The experimental results for the experiment are shown in Figure 3. The X-axis of Figure 3 are the consumed time units. The Y-axis of Figure 3 are the total number of finished tasks. From Figure 3, it can be seen that without communicational constraints, 'MILP' has the best performance on task allocation. That is because 'MILP' can always create the optimal solution for task allocation based on the global knowledge about the environment, which can be taken as the benchmark in this experiment. With communicational constraints, the performance of task allocation of 'MS' is better than that of 'TASC' at the beginning of the experiment. That is because all agents in 'MS' participate in task execution, while representative agents of the proposed approach are only in charge of coordination with other groups. However, after one round of coordination, 'TASC' can quickly adjust group members (agents) among isolated groups and the group coordination mechanism can achieve continuous coordination in 'TASC', which keep the performance of 'TASC' better than that of 'MS' in the following task allocation. Without a coordination mechanism, communicational constraints and dynamic features of the environment limits the performance of 'MS' in long-term task allocation in the experiment.

5 Related Work

Some approaches handle the coordination for the task allocation problem in a centralised manner, such as the MILP based approaches (Ramchurn et al. 2010b; Koes, Nourbakhsh, and Sycara 2005). The centralised approaches can guarantee an optimal solution for task allocation if the central controller has the global view of the environment. However, due to communicational constraints, it is hard for the central controller to have such view of the environments. According to the proposed approach, tasks and agents of communication network are divided into groups with suitable space ranges, within which agents can always communicate with each other so that the centralised task allocation approach can be employed by agents of each group for task allocation.

Some approaches handle the coordination for the task allocation problem in a decentralised manner, such as the max-sum algorithm based approaches (Farinelli et al. 2008; Ramchurn et al. 2010a). These decentralised approaches enable agents to make decisions for task allocation based on information exchange of the max-sum algorithm. However, in order to collect comprehensive information for task allocation, each agent needs to exchange information with its direct neighbours, which needs a plenty of time to achieve and cannot suit dynamic features of the environments. In our information collection mechanism, a connected agent (e.g. A_i) only needs to pass information for task allocation to its network leader (i.e., $A_i.L$) through its parent agent (i.e., $A_i.PA$), which saves much time and resource for information exchange for task allocation.

In recent years, the DARPA coordinators program has been a popular simulation environment for task allocation approaches (Smith, Gallagher, and Zimmerman 2007; Barbulescu et al. 2010). The main difference between the environments of the DARPA coordinators program and the proposed approach is that the environment of the DARPA coordinators program does not take either spatial or communicational constraints into account, while the proposed approach deals with coordination for task allocation problem in disaster environments by considering spatial and communicational constraints, dynamic features of the environments as well as heterogenous capabilities of agents.

6 Conclusion and Future Work

In this paper, an innovative dynamic coordination approach for task allocation in disaster environments under space and communicational constraints is proposed. The proposed approach first prunes communication networks and collects information for task allocation in a decentralised manner according to the information collection mechanism. Then, tasks and agents of each communication network are divided into groups with suitable spatial ranges according to the group task allocation mechanism. In order to adjust group members (agents) among isolated groups during task execution, groups periodically coordinate with each other at assembly points so as to achieve continuous and dynamic coordination for task allocation by the use of the group coordination mechanism. In the future, we would like to adjust the proposed approach to handle task allocation in competitive environments, such as market-based environments, such as (Chapman et al. 2009).

References

Allouche, M. K., and Boukhtouta, A. 2010. Multi-agent coordination by temporal plan fusion: Application to combat search and rescue. *Information Fusion* 11(3):220–232.

Altshiller-Court, N. 2007. *College Geometry: An Introduction to the Modern Geometry of the Triangle and the Circle*. Dover Books on Mathematics. Dover Publications.

Barbulescu, L.; Rubinstein, Z. B.; Smith, S. F.; and Zimmerman, T. L. 2010. Distributed coordination of mobile agent teams: the advantage of planning ahead. In *AAMAS '10*, 1331–1338.

Chapman, A. C.; Micillo, R. A.; Kota, R.; and Jennings, N. R. 2009. Decentralised dynamic task allocation: a practical game: theoretic approach. In *AAMAS '09*, volume 2.

Comaniciu, D., and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5):603–619.

Farinelli, A.; Rogers, A.; Petcu, A.; and Jennings, N. R. 2008. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In *AAMAS '08*, volume 2, 639–646.

Kang, R. J., and Müller, T. 2011. Sphere and dot product representations of graphs. In *Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry*, SoCG '11, 308–314. New York, NY, USA: ACM.

Koes, M.; Nourbakhsh, I.; and Sycara, K. 2005. Heterogeneous multirobot coordination with spatial and temporal constraints. In *AAAI'05*, 1292–1297. AAAI Press.

Lesser, V. 1999. Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering* 11:133–142.

Musliner, D. J., and Goldman, R. P. 2006. Coordinated plan management using multiagent mdps. In *AAAI '06*, 73–80. AAAI Press.

Ramchurn, S. D.; Farinelli, A.; Macarthur, K. S.; and Jennings, N. R. 2010a. Decentralized coordination in robocup rescue. *The Computer Journal* 53(9):1447–1461.

Ramchurn, S. D.; Polukarov, M.; Farinelli, A.; Truong, C.; and Jennings, N. R. 2010b. Coalition formation with spatial and temporal constraints. In *AAMAS 2010*, 1181–1188.

Reich, J. 2006. Toward automatic reconfiguration of robot-sensor networks for urban search and rescue. In *In First International Workshop on Agent Technology for Disaster Management (ATDM): Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM.

Smith, S. F.; Gallagher, A.; and Zimmerman, T. 2007. Distributed management of flexible times schedules. In *AAMAS '07*, volume 74, 1–8. New York, NY, USA: ACM.

Smith, R. G. 1980. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput.* 29(12):1104–1113.

Wu, J.; Xu, X.; Zhang, P.; and Liu, C. 2011. A novel multi-agent reinforcement learning approach for job scheduling in grid computing. *Future Generation Computer Systems* 27(5):430–439.