J Syst Sci Syst Eng (Apr 2018) 27(2):215-230
DOI: https://doi.org/10.1007/s11518-018-5365-9
ISSN: 1004-3756 (Paper) 1861-9576 (Online)
CN11-2983/N

TWO INNOVATIVE COALITION FORMATION MODELS FOR DYNAMIC TASK ALLOCATION IN DISASTER RESCUES

Xing Su[1] Yuechen Wang[1] Xibin Jia[1] Limin Guo[1*] Zhiming Ding[1]

[1] *Faculty of Information, Beijing University of Technology, Beijing, 100124, China*

guolimin@bjut.edu.cn (✉)

xingsu@bjut.edu.cn, wangyuechen@emails.bjut.edu.cn, jiaxibin@bjut.edu.cn, zmding@bjut.edu.cn

Abstract

This paper addresses the problem of multi-objective coalition formation for task allocation. In disaster rescue, due to the dynamics of environments, heterogeneity and complexity of tasks as well as limited available agents, it is hard for the single-objective and single (task)-to-single (agent) task allocation approaches to handle task allocation in such circumstances. To this end, two multi-objective coalition formation for task allocation models are proposed for disaster rescues in this paper. First, through coalition formation, the proposed models enable agents to cooperatively perform complex tasks that cannot be completed by single agent. In addition, through adjusting the weights of multiple task allocation objectives, the proposed models can employ the linear programming to generate more adaptive task allocation plans, which can satisfy different task allocation requirements in disaster rescue. Finally, through employing the multi-stage task allocation mechanism of the dynamic programming, the proposed models can handle the dynamics of tasks and agents in disaster environments. Experimental results indicate that the proposed models have good performance on coalition formation for task allocation in disaster environments, which can generate suitable task allocation plans according to various objectives of task allocation.

Keywords: Disaster rescues, multi-objective linear programming, cooperative agents, multi-stage task allocation

1. Introduction

Nowadays, disasters throughout the world such as 2004 Indian Ocean tsunami (Athukorala 2005), 2005 hurricane Katrina (Banipal 2005), 2008 Sichuan earthquake (Li 2009), etc. have become important social and political concerns. After disasters happened, many tasks need suitable agents (i.e., first responders or resources) to complete. In such circumstances, efficient and effective task allocation can significantly reduce casualties and economic losses. How- ever, task allocation for disaster rescue need to face several challenges, which include: **1) Temporal constraints.** In disaster rescue, tasks include saving survivors in debris, extinguishing fire of buildings, etc. Most of these tasks should have hard deadlines (i.e., the time points before which survives are still alive, buildings are still standing, etc.) (Ramamritham 1989, Ramchurn 2010). Tasks that are completed after their deadlines are considered as failure because of meaninglessness. **2) Complex tasks.** the complex task indicates the task with huge workloads or tight deadlines, which are hard for single agent to complete before its deadline. In such circumstance, agents have to form coalitions to cooperatively perform complex tasks (Shehory 1998, Predrag 2004). **3) Various task allocation objectives.** In disaster rescue, heterogeneous tasks have different allocation objectives (Yin 2007, Szabo 2012). For example, when allocating agents to save survivors from debris, in addition to the deadline, the task allocation should also guarantee the effect of completed tasks. However, when allocating agents to extinguish fire of buildings, completion time of tasks is the only important objective of the task allocation. **4) Different capabilities of agents.** Different agents have different capabilities, which not only determine the types of tasks that they can complete, but also affect the effect of their completed tasks (Koes 2005, Su 2014). **5) Limited available agents.** Due to blocking roads and uneven distribution, agents cannot quickly enter the disaster environments. Therefore, in a disaster environment, there are only limited available agents, which are much less than tasks (Mulcaire 2013). **6) Dynamic environments.** Disaster environments are highly dynamic, where tasks can be continuously detected and completed and agents can be arriving and departing (Smith 2007, Chapman 2009).

To achieve efficient and effective task allocation, various approaches have been proposed in the last twenty years. Some researchers consider the task allocation problem as the optimal assignment problem (OAP), where tasks and agents are allocated single-to-single (Nie 2010, Zhang 2013, Jalil 2016). In the approaches of OAP, it is easy to find optimal task allocation solution in polynomial time. However, as aforementioned, in disaster rescue, there are many complex tasks which cannot be completed by single agent. Since the approaches for OAP cannot handle such problem, they cannot achieve efficient task allocation in disaster rescues. Some researchers employ the max-sum algorithm to achieve task allocation in a decentralized manner (Farinelli 2008, Rogers 2011, Farinelli 2014). In max-sum based approaches, through exchanging and adjusting the utility function, agents can generate a consistent task allocation plan, where agents can cooperatively perform tasks. However, to simplify the process of utility function exchange and adjustment, the max-sum based approaches do not consider multiple task allocation objectives so they are hard to handle heterogeneous task allocation in disaster environments. Some researchers employ the mixed integer linear programming (MILP) to achieve efficient and effective task allocation in disaster rescue (Koes 2005, Davare 2006, Rueda-Medina 2013). Through such approaches, agents can find optimal task allocation plan to cooperatively perform tasks. However, finding an optimal task allocation plan through the MILP is a time-consuming process, so the MILP-based approaches do not suit the dynamics of disaster environments.

In general, the main limitations of existing approaches for task allocation in disaster environments can be summarized as follows. 1) In some approaches, the task allocation mechanism is single-to-single, which cannot handle complex tasks. 2) Most approaches do not consider the different objectives of tasks and use the number of completed tasks as the only criterion for task allocation. 3) Most approaches do not consider the different capabilities of agents and their suitability for task completion. 4) Some approaches generate optimal task allocation plan through increasing time consumption, which limits their application for task allocation in dynamic disaster environments.

To overcome the above limitations of existing approaches and achieve efficient task allocation in disaster environments, in this paper, two coalition formation models are proposed for dynamic task allocation in disaster environments. 1) The first model is a heuristic-based model, which employs a heuristic method to generate task allocation plans. 2) The second model is a multi-objective linear programming (MOLP) -based model, which employs the MOLP method to find the task allocation plan. The contributions of the proposed models are described as follows.

- The proposed models enable multiple agents to form coalitions to cooperatively perform tasks so as to handle complex tasks in disaster environments.
- The proposed models include multiple task allocation objectives. Through adjusting the weights of objectives, the proposed models can satisfy different task allocation requirements of tasks.
- The proposed models consider the different capabilities of agents and evaluate the effects of task allocation through the suitability between types of tasks and capabilities of agents.
- The proposed models employ the multi-stage task allocation mechanism of the dynamic programming to address the long-term task allocation in highly dynamic disaster environments.

The rest of the paper is organized as follows. Section 2 gives the definitions and the problem description. Section 3 introduces the basic principle of the proposed models. Section 4 introduces the heuristic method in detail. Section 5 introduces the MOLP method in detail. Section 6 demonstrates and analyzes experimental results. Section 7 compares the related work with the proposed models. Section 8 is the conclusion and future work.

2. Problem Description and Definition

In this section, first, the definitions of coalition formation for task allocation are given and explained. Then, multiple objectives for task allocation of the proposed models are introduced in detail.

2.1 Definitions of Coalition Formation for Task Allocation

In a disaster environment, there are M number of tasks, which are described as $TASK = \{t_1, t_2, ..., t_m\}$, where t_i represents the i^{th} task. At the same time, there are n number of agents, which are described as $AGENT = \{a_1, a_2, ..., a_n\}$, where a_j represents the j^{th} agent. Here, $m \gg n$. The definitions of a task and an agent are given as follows.

Definition 1 A *task* (t_i) can be defined as a three-tuple.

$$t_i = < id, dline, cap_i >, \tag{1}$$

where *id* is the unique identification of t_i; $dline_i$ is the deadline of t_i and cap_i is the type of t_i, which is described by a vector $cap_i = (c_1^i, c_2^i, ..., c_R^i)$, where c_r^i is to indicate whether t_i requires the r^{th} capability. If t_i requires the r^{th} capability to be completed, c_r^i =1, otherwise, c_r^i =0.

Definition 2 An *agent* (a_j) can be defined as a three-tuple.

$$a_j = < id,\ cap_j >, \tag{2}$$

where *id* is the unique identification of a_j; and cap_j is the capabilities of a_j, which is described by a vector $cap_j = (c_1^j, c_2^j, ..., c_R^j)$, where c_r^j is to indicate whether a_j has the r^{th} capability. If a_j has the r^{th} capability, c_r^j =1, otherwise, c_r^j =0.

For each task (t_i), each agent (a_j) needs to evaluate the completion time ($time_{j,i}$) and effect ($effect_{j,i}$). $time_{j,i}$ is the time that a_j needs to complete t_i. While $effect_{j,i}$ is the effect of t_i completed by a_j, which can be calculated from the cosine value between the type of t_i (cap_i, see, Definition 1) and the capabilities of a_j (cap_j, see, Definition 2), which is described as follows.

$$effect_{j,i} = \frac{\sum_{r=1}^{R} c_r^i \cdot c_r^j}{\sqrt{\sum_{r=1}^{R} {c_r^i}^2} \cdot \sqrt{\sum_{r=1}^{R} {c_r^j}^2}}, \tag{3}$$

In disaster rescue, complex tasks with huge workloads or tight deadlines cannot be completed by single agent. In such circumstances, multiple agents have to form coalitions to cooperatively perform and complete them. In the coalition formation for task allocation, all agents are divide into different coalitions, which can be defined as follows.

Definition 3 A *division of agents* (d_p) can be defined as a set of coalitions.

$$d_p = \{g_1, g_2, ...\}, \tag{4}$$

where g_k is the k^{th} coalition of agents in d_p.

Definition 4 A *coalition of agents* (g_k) can be defined as a set of agents.

$$g_k = \{a_1, a_2, ...\}, \tag{5}$$

where a_j is the j^{th} coalition of agents in g_k.

To simplified the description of task allocation, the proposed models only allocate coalitions to tasks, where a coalition could include 1～n number of agents. The allocation of a coalition (g_k) to a task (t_i) is indicated by $x_{k,i}$, where if g_k is allocated to t_i, $x_{k,i}$ = 1, otherwise, $x_{k,i}$ = 0.

The time ($UT_{k,i}$) and effect ($UE_{k,i}$) of t_i completed by a coalition g_k can be evaluated by uniting the time ($time_{j,i}$) and effect ($effect_{j,i}$) of t_i completed by each agent in g_k, where $UT_{k,i}$ and $UE_{k,i}$ are calculated as follows.

$$UT_{k,i} = x_{k,i} \cdot \frac{1}{\sum_{a_j \in g_k} \frac{1}{time_{j,i}}}, \tag{6}$$

$$UE_{k,i} = x_{k,i} \cdot \frac{1}{s} \sum_{a_j \in g_k} effect_{j,i}, \tag{7}$$

where s is the number of agents in g_k (see, Definition 3).

2.2 Multiple Objectives for Task Allocation in Disaster Rescue

In disaster rescue, heterogeneous tasks have different task allocation objectives, which include the effectiveness objective, the efficiency objective, etc. To achieve efficient and effective task allocation in disaster rescue, the proposed models consider three main task allocation objectives, which are the effectiveness objective, the efficiency objective and the execution time objective.

The effectiveness objective trends to maximize the effect of completed tasks. Based on above definitions and descriptions, the effectiveness objective can be calculated as follows.

$$\max \sum_{k=1}^{q} \sum_{i=1}^{m} x_{k,i} \cdot UE_{k,i}, \tag{8}$$

where q is the number of allocated coalitions and m is the number of tasks.

The efficiency objective trends to maximize the number of completed tasks. Based on above definitions and descriptions, the efficiency objective can be calculated as follows.

$$\max \sum_{k=1}^{q} \sum_{i=1}^{m} x_{k,i} \ . \tag{9}$$

The execution time objective trends to minimize the execution time of allocated tasks. Based on above definitions and descriptions, the execution time objective can be calculated as follows.

$$\min \sum_{k=1}^{q} \sum_{i=1}^{m} x_{k,i} \cdot UT_{k,i} \tag{10}$$

3. The Basic Principle of the Proposed Models

In disaster rescue, most of time, the number of tasks (m) is much more than the number of agents (n) (i.e., $m \gg n$). In such circumstances, through a task allocation, agents cannot complete all tasks in the disaster environment. To complete as many tasks as possible and suit the dynamics of disaster environments, the multi-stage task allocation mechanism of the dynamic programming is borrowed by our heuristic-based model. Based on the above ideas, in each stage of task allocation, the proposed models perform the following two steps.

1. According to tasks and agents, find the most suitable task allocation plan for the current stage;
2. According to the task allocation plan, allocate coalitions of agents to tasks until all allocated agents completed their tasks.

The above two steps are iteratively performed in each task allocation stage until all tasks in a disaster environment are completed or exceed their deadlines. The task allocation process of the proposed models are described by Algorithm 1.

Algorithm 1 The task allocation process of the proposed models

1 Input: *TASK, AGENT*
2 Output: *Alloc*
3 $curTime = 0$, $Alloc = Ø$
4 **while** $TASK \neq Ø$ **do**
5    $A_{best} = CFTAM\ (TASK, AGENT)$
6    $curTime = curTime + MAX(UT_{k,i})$
7   **for *each*** $x_{k,I} \in A_{best}$ **do**
8       $TASK = TASK \setminus t_{i,k}$
9       $Alloc = Alloc\{a_j \in g_k \rightarrow t_i\}$
10   **end**
11   **for *each*** $x_{k,i} \in A_{best}$ **do**
12   **if** $curTime > dlinei$ **then**
13          $TASK = TASK \setminus t_i$
14   **end**
15   **end**
16   *Update*(*TASK, AGENT*)
17 **end**

Algorithm 1 can be explained as follows. The inputs of the proposed models include all tasks *TASK* and all agents *AGENT* in a disaster environment (see, Definitions 1 and 2) (Line 1). The output of the proposed models is all task allocation plan *Alloc* for *TASK* and *AGENT* (Line 2). Before task allocation, the current time *curTime* and *Alloc* are initialized to 0 and *Ø*, respectively (Line 3). The task allocation in the proposed models is a multi-stage cycling process until there is no task (i.e., all tasks are completed or exceed their deadlines) (Line 4). In each stage, the proposed models employ the coalition formation based task allocation method (i.e., CFTAM) to find the most suitable task allocation plan A_{best} for tasks and agents of the current stage and allocate coalitions of agents to perform tasks according to A_{best} (Line 5). During task performing, the models push *curTime* to the latest completion time of allocated tasks (Line 6), eliminate the allocated tasks from *TASK* (Line 7) and store A_{best} into *Alloc* (Line 10). After that, the models eliminate uncompleted tasks from *TASK*, whose deadlines are exceeded (Lines 11 to 13). Finally, due to the dynamics of the disaster environment, the models update *TASK* and *AGENT* through adding new tasks and agents (Line 17).

The general process of the two proposed models are similar, while they employ different methods to generate the most suitable task allocation plan (i.e., CFTAM, Line 5 in Algorithm 1), which will be introduced in the following two sections.

4. The Heuristic Method for Task Allocation Plan Finding

The heuristic-based model employs a heuristic method to find the most suitable task allocation plan (i.e., A_{best}), which can be described by Algorithm 2.

Algorithm 2 The heuristic method

```
1  Input: TASK, AGENT
2  Output: A_best
3  DIV = Division(AGENT)
4  for each d_p ∈ DIV do
5      (obj_p, A_p) = MOLP(d_p, TASK)
6  end
7  A_best = A_p with MAX(obj_p)
```

Algorithm 2 can be explained as follows. The inputs of the heuristic method are all tasks *TASK* and all agents *AGENT* in the current task allocation stage (see, Definitions 1 and 2) (Line 1). The output of the heuristic method is the most suitable task allocation plan A_{best} for the current stage (Line 2). The heuristic method first finds all divisions of agents and stores in *DIV* (Line 3). For each division of agents d_p in *DIV* (see, Definition 3), the method employs the MOLP to find the task allocation plan A_p of d_p and calculates the objective value obj_p of A_p. After found A_p and calculated obj_p for all d_p in *DIV*, the method chooses the A_p with the maximum obj_p as the most suitable task allocation plan A_{best}.

4.1 Finding All Divisions of Agents

Before employing the MOLP, the heuristic method needs to find all divisions of agents *DIV*. To achieve this, an incremental process is employed by the method to generate *DIV*, where all a_j in *AGENT* are sequentially added to all subsets in *DIV*. For an example of *AGENT* = $\{a_1, a_2, a_3, a_4\}$, the incremental process to find *DIV* is shown in Table 1.

Table 1 The incremental process

a_j	*DIV*	Increament of *DIV*
a_1	$\{\}$	$\{a_1\}$
a_2	$\{a_1\}$	$\{a_1, a_2\}$ $\{a_1\}, \{a_2\}$
a_3	$\{a_1, a_2\}$ $\{a_1\}, \{a_2\}$	$\{a_1, a_2, a_3\}$ $\{a_1, a_2\}, \{a_3\}$ $\{a_1, a_3\}, \{a_2\}$ $\{a_1\}, \{a_2, a_3\}$ $\{a_1\}, \{a_2\}, \{a_3\}$
a_4	$\{a_1, a_2, a_3\}$	$\{a_1, a_2, a_3, a_4\}$

$\{a_1, a_2\}, \{a_3\}$	$\{a_1, a_2, a_3\}, \{a_4\}$
$\{a_1, a_3\}, \{a_2\}$	$\{a_1, a_2, a_4\}, \{a_3\}$
$\{a_1\}, \{a_2, a_3\}$	$\{a_1, a_2\}, \{a_3, a_4\}$
$\{a_1\}, \{a_2\}, \{a_3\}$	$\{a_1, a_2\}, \{a_3\}, \{a_4\}$
	$\{a_1, a_3, a_4\}, \{a_2\}$
	$\{a_1, a_3\}, \{a_2, a_4\}$
	$\{a_1, a_3\}, \{a_2\}, \{a_4\}$
	$\{a_1, a_4\}, \{a_2, a_3\}$
	$\{a_1\}, \{a_2, a_3, a_4\}$
	$\{a_1\}, \{a_2, a_3\}, \{a_4\}$
	$\{a_1, a_4\}, \{a_2\}, \{a_3\}$
	$\{a_1\}, \{a_2, a_4\}, \{a_3\}$
	$\{a_1\}, \{a_2\}, \{a_3, a_4\}$
	$\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}$

4.2 Finding the Task Allocation Plan

After finding all divisions of agents DIV, the heuristic method employs the MOLP to find task allocation plan A_p for each division of agents d_p in DIV and calculates the objective value obj_p of A_p. The formulation and constraints of the MOLP in the heuristic method are described as follows.

$$\max\ \alpha\sum_{k=1}^{q}\sum_{i=1}^{m}x_{k,i}UE_{k,i}+\beta\sum_{k=1}^{q}\sum_{i=1}^{m}x_{k,i}\eta_1-\gamma\sum_{k=1}^{q}\sum_{i=1}^{m}x_{k,i}UT_{k,i}\eta_2 \quad (11)$$

$$\text{s.t.}\quad \sum_{k=1}^{q}x_{k,i}\le 1, \quad (12)$$

$$\sum_{i=1}^{m}x_{k,i}\le 1, \quad (13)$$

$$\forall t_i : curTime+UT_{k,i}\le dline_i, \quad (14)$$

The objective function of the MOLP (equation (11)) includes the effectiveness objective, the efficiency objective and the execution time objective (see, equations (8) to (10)) and their weights α (the effectiveness objective), β (the efficiency objective) and γ (the execution time objective) ($\alpha, \beta, \gamma \in [0,1]$ and $\alpha + \beta + \gamma = 1$). η_1 and η_2 are two balance coefficients (they will be discussed in the next subsection). The constraint of equation (12) limits that one coalition can only be allocated to one task; While the constraint of equation (13) limits that one task can only be allocated to one coalition; The constraint of equation (14) limits that the execution time of tasks cannot exceed their deadlines, (i.e., allocated tasks can be completed before their deadlines).

The objective value obj_p of the task allocation plan A_p is calculated based on the objective function of the MOLP (see, equation (11)).

4.3 The Balance Coefficients

To enable our models to satisfy different task allocation objectives in disaster rescue, α, β and γ are introduced into the objective function of the MOLP (see, equation (11)) so as to adjust the weights of three task allocation objectives. However, since the ranges of three task allocation objectives are different, the task allocation plans generated by the MOLP are not consistent with the values of α, β and γ. To enable α, β and γ to correctly reflect relationships between three task allocation objectives, the balance coefficients are introduced.

In equation (11), it can be seen that two balance coefficients (i.e., η_1 and η_2) are introduced to the efficiency objective and the execution time objectives of MOLP so as to balance their ranges to the effectiveness objective.

Since the difference between effectiveness objective and the efficiency objective is only an effect value ($UE_{k,i}$), the balance coefficient of the efficiency objective η_1 is the average effect of allocated tasks, which is calculated as follows.

$$\eta_1 = \frac{\sum_{k=1}^{q}\sum_{i=1}^{m} x_{k,i} UE_{k,i}}{\sum_{k=1}^{q}\sum_{i=1}^{m} x_{k,i}} . \quad (15)$$

The balance coefficient of the execution time objective η_2 can be explained as the ratio of the total effects to the total execution time of the allocated tasks, which is calculated as follows.

$$\eta_2 = \frac{\sum_{k=1}^{q}\sum_{i=1}^{m} x_{k,i} UE_{k,i}}{\sum_{k=1}^{q}\sum_{i=1}^{m} x_{k,i} UT_{k,i}} . \quad (16)$$

5. The MOLP Method for Task Allocation Plan Finding

In the MOLP method, the process of finding the most suitable task allocation plan (i.e., A_{best}) can be described by Algorithm 3.

Algorithm 3 The MOLP method

```
1  Input: TASK, AGENT
2  Output: A_best
3  G = Coalition(AGENT)
4  A_best = MOLP(G, AGENT)
```

Algorithm 3 can be explained as follows. The inputs of the method are all tasks *TASK* and all agents *AGENT* in the current task allocation stage (see, Definitions 1 and 2) (Line 1). The output of the MOLP method is the most suitable task allocation plan A_{best} for the current stage (Line 2). The MOLP method first finds all coalitions of agents and stores in G (see, Definition 4) (Line 3). Then, the MOLP method directly finds the most suitable task allocation plan (i.e., A_{best}) through the MOLP formulation of G and *TASK* (Line 4).

5.1 Finding All Coalitions of Agents

Rather than finding all divisions, the *MOLP* method first finds all possible coalitions of all available agents G, which is an enumerative process. The same example of $AGENT = \{a_1, a_2, a_3, a_4\}$ is used to demonstrate the enumerative process to find G, which is shown in Table 2.

Table 2 The enumerative process

n	C_4^n	G
1	C_4^1=4	$g_1=\{a_1\}$ $g_2=\{a_2\}$ $g_3=\{a_3\}$ $g_4=\{a_4\}$
2	C_4^2=6	$g_5=\{a_1, a_2\}$ $g_6=\{a_1, a_3\}$ $g_7=\{a_1, a_4\}$ $g_8=\{a_2, a_3\}$ $g_9=\{a_2, a_4\}$ $g_{10}=\{a_3, a_4\}$
3	C_4^3=4	$g_{11}=\{a_1, a_2, a_3\}$ $g_{12}=\{a_1, a_2, a_4\}$ $g_{13}=\{a_1, a_3, a_4\}$ $g_{14}=\{a_2, a_3, a_4\}$
4	C_4^4=1	$g_{15}=\{a_1, a_2, a_3, a_4\}$

5.2 Finding the Task Allocation Plan

After finding all coalitions of agents G, the method builds the MOLP based on G and *TASK*. The formulation and constraints of the MOLP of the MOLP method are described as follows.

$$\max \ \alpha\sum_{k=1}^{q}\sum_{i=1}^{m} x_{k,i}UE_{k,i} + \beta\sum_{k=1}^{q}\sum_{i=1}^{m} x_{k,i}\eta_1 - \gamma\sum_{k=1}^{q}\sum_{i=1}^{m} x_{k,i}UT_{k,i}\eta_2 \quad (17)$$

$$\text{s.t.} \quad \sum_{k=1}^{q} x_{k,i} \le 1, \quad (18)$$

$$\sum_{i=1}^{m} x_{k,i} \le 1, \quad (19)$$

$$\forall t_i : curTime + UT_{k,i} \le dline_i, \quad (20)$$

$$\forall a_j \in AGENT : \sum_{a_j \in g_k} x_{k,i} \le 1 \quad (21)$$

The equations (17) to (20) of the MOLP method are the same as the equations (11) to (14) of the heuristic method. Be different with the heuristic method, equation (21) limits that the coalitions that have the same agent can only be allocated once. The MOLP method should include a constraint for each agent in *AGENT*. For the example of $AGENT = \{a_1, a_2, a_3, a_4\}$ in Table 2, the constraints of equation (21) are shown as follows.

$$a_1 : x_{1,i} + x_{5,i} + x_{6,i} + x_{7,i} + x_{11,i} + x_{12,i} + x_{13,i} + x_{15,i} \le 1 \quad (22)$$

$$a_2 : x_{2,i} + x_{5,i} + x_{8,i} + x_{9,i} + x_{11,i} + x_{12,i} + x_{14,i} + x_{15,i} \le 1 \quad (23)$$

$$a_3 : x_{3,i} + x_{6,i} + x_{8,i} + x_{10,i} + x_{11,i} + x_{13,i} + x_{14,i} + x_{15,i} \le 1 \quad (24)$$

$$a_4 : x_{4,i} + x_{7,i} + x_{9,i} + x_{10,i} + x_{12,i} + x_{13,i} + x_{14,i} + x_{15,i} \le 1 \quad (25)$$

where equation (22) can be explained as that since coalitions g_1, g_5, g_6, g_7, g_{11}, g_{12}, g_{13} and g_{15} have the agent a_1, their allocation indicators $x_{1,i}$, $x_{5,i}$, $x_{6,i}$, $x_{7,i}$, $x_{11,i}$, $x_{12,i}$, $x_{13,i}$, and $x_{15,i}$ cannot be allocated more than once.

6. Experiments and Analysis

Two experiments are conducted to evaluate the performance of the proposed models. In the first experiment (Experiment 1), task allocation plans generated by the proposed models are evaluated under different weights of task allocation objectives. In the second experiment (Experiment 2), the time consumed for task allocation (i.e., computational complexity) of the proposed models are compared. All experiments are implemented in Matlab 2014a.

6.1 Experimental Settings

In experiments, there are 50 tasks that need to be completed, whose deadlines are range from 10 to 40 units of time. There are 5 agents to perform tasks. The time consumptions for an agent to complete a task are range form 5 to 20 units of time. According to the cosine values between two vectors (i.e., types of tasks and capabilities of agents), the completion effects of tasks are range from 0 to 1, where 0 and 1 represent the worst and the best completion effects of tasks, respectively. The settings of experiments are described in Table 3.

Table 3 The settings of experiments

Variables	Values
The number of tasks	50
The deadlines of tasks	10 ~ 40 units of time
The number of agents	5
The time of an agent to perform a task	5 ~ 20 units of time
The completion effect of tasks	0 (worst) ~ 1 (best)

To evaluate the performance of the proposed models on the multi-objective task allocation, we use different weights of the three task allocation objectives (the effectiveness objective α, the efficiency objective β and the execution time objective γ). In experiments, five kinds of values of α, β and γ are shown in Table 4.

Table 4 The weights of three task allocation objectives

α effectiveness	β efficiency	γ execution time
1	0	0
0	1	0
0.5	0	0.5
0	0.5	0.5
1/3	1/3	1/3

In Experiment 1, two proposed models separately perform task allocation under five different weights of task allocation objectives. The three indicators of task allocation are compared to analyze the performance of the proposed models, which are 1) the average effect of completed tasks (i.e., $avg(UE_{k,i})$), 2) the average number of completed tasks in each stage (i.e., $avg(\sum x_{k,i})$) and 3) the average execution time of completed tasks (i.e., $avg(UT_{k,i})$).

In Experiment 2, the time consumed for task allocation of the two proposed models are compared, when the values of α, β and γ are 1/3, 1/3 and 1/3, respectively.

6.2 The Results of Experiment 1

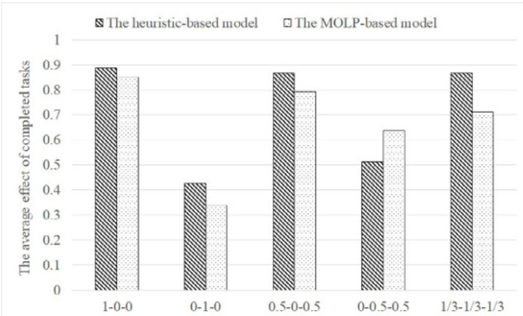

Figure 1 The average effects of completed tasks

The average effects of completed tasks under five different weights of task allocation objectives are shown in Figure 1. In Figure 1, the X-axis is the weights of task allocation objectives. The Y-axis is the average effect of tasks that are completed by agents. From Figure 1, it can be seen that two proposed models have different task allocation performance under the same weights of task allocation objectives. This is because that in a task allocation stage, there are several most suitable task allocation plans, without other constraints, two models may choose different task allocation plans, which affects the sequential task allocation. The tasks allocated by considering the effectiveness objective (i.e., 1-0-0, 0.5-0-0.5 and 1/3-1/3-1/3) have higher completion effects than that allocated without considering the effectiveness objective (i.e., 0-1-0 and 0-0.5-0.5). We can say that the effectiveness objective has great impacts on the effects of completed tasks. In three task allocations by considering the effectiveness objective (i.e., 1-0-0, 0.5-0-0.5 and 1/3-1/3-1/3), the tasks allocated according to 1-0-0 weights have the highest completion effects. The tasks allocated according to 0.5-0-0.5 and 1/3-1/3-1/3 have similar completion effects, which corresponds to their weights of task allocation objectives.

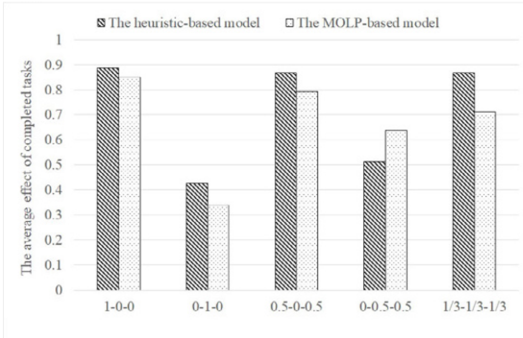

Figure 2 The average number tasks completed in each stage

The average numbers of completed tasks in each stage under five different weights of task allocation objectives are shown in Figure 2. In Figure 2, the X-axis is the weights of task allocation objectives. The Y-axis is the average number of tasks that are completed in a stage of task allocation. From Figure 2, it can be seen that based on 0-1-0, 0.5-0-0.5 and 1/3-1/3-1/3, the proposed model can allocate more tasks in each task allocation stage than based on 1-0-0 and 0.5-0-0.5. This is because that the efficiency objective aims to complete as many tasks as possible. Based on 0-1-0, the models can achieve the highest task allocation efficiency, where in each task allocation stage, five tasks are completed (i.e., five agents perform five different tasks). However, since the multi-stage of task allocation mechanism employed by the proposed models is myopic, where models maximize completed tasks in the whole task allocation process through maximizing completed tasks in each task allocation stage and leads to the impact of the efficiency objective is not obvious. In addition, the execution time objective aims to minimize the execution time of tasks, which is opposite to the efficiency objective. Therefore, the 0.5-0-0.5 and 1/3-1/3-1/3 task allocations can complete less tasks than the 0-1-0 task allocation in each stage of task allocation.

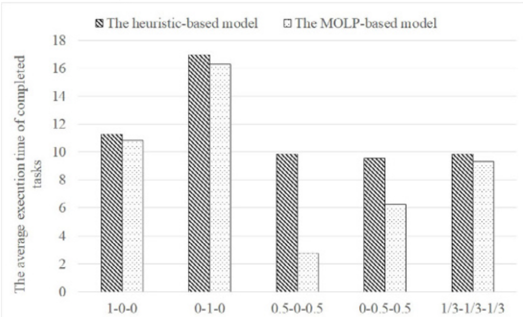

Figure 3 The average execution time of completed tasks

The average execution time of completed tasks under five different weights of task allocation objectives are shown in Figure 3. In Figure 3, the X-axis is the weights of task allocation objectives. The Y-axis is the average execution time of tasks that are completed by agents. From Figure 3, it can be seen that tasks allocated based on 0-1-0 has the longest execution time. This is because based on 0-1-0, the proposed models do not consider the time execution objective and the efficiency objective is opposite to the execution time objective, where agents do not cooperate to perform tasks. Although based on 1-0-0, the proposed models do not consider the execution time objective neither, the effectiveness objective does not affect the execution time of tasks, tasks have a little shorter execution time than 0-1-0. The task allocations based on 0.5-0-0.5, 0.5-0-0.5 and 1/3-1/3-1/3 consider the execution time objective so their allocated tasks have shorter execution time. Since the task allocation based on 0.5-0-0.5 does not consider the efficiency objective (i.e., the opposite objective of the execution time objective), its tasks has the shortest execution time among five different weights of task allocation objectives.

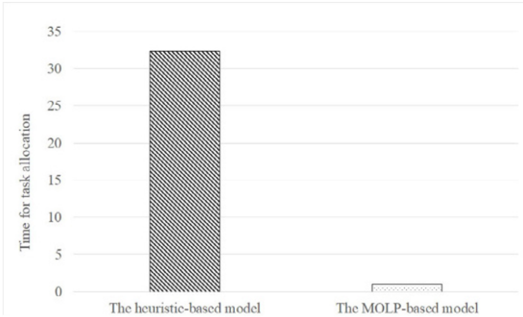

Figure 4 The time consumed for task allocation of the proposed models

6.3 The Results of Experiment 2

The time consumed for task allocation of the two proposed models under the 1/3-1/3-1/3 weights of task allocation objectives are shown in Figure 4. In Figure 4, the X-axis is the two proposed models. The Y-axis is time consumed for task allocation. From Figure 4, it can be seen that the heuristic-based model consumes much more time to generate task allocation plan than the MOLP-based model. This is because that the heuristic-based model uses an incremental mechanism to find all divisions for all agents. Table 1 indicates that the number of divisions increases exponentially with the increase number of agents. In addition, the heuristic-based model also needs to employ the MOLP to generate task allocation plan for each division. Be different with the heuristic-based model, although the MOLP-based model also uses an enumerative process to find all coalitions of agents, the number of coalitions increases linearly with the increase number of agents. In addition, the MOLP-based model only use the MOLP for once to generate the most suitable task allocation plan by adding some constraints in the MOLP, which greatly reduces the computational complexity (time) of the model.

7. Related Work

In the recent years, many approaches have been proposed to handle task allocation problem from different perspectives (Akter 2017, Tanzi 2017, Su 2014, Su 2016, Tsarouchi 2017, Hooshangi 2017).

In (Gerkey 2004), Gerkey et al. proposed a taxonomy to category task allocation approaches based on three criteria: 1) a task can be completed by a single robot (i.e., agent) (SR) or multiple robots (MR); 2) a robot can work on a single task (ST) or multiple tasks (MT); 3) the task allocation process is instantaneous (IA) or time-extended (TA). Based on the taxonomy of Gerkey et al., the proposed models can be categorized as MR-ST-TA task allocation approach.

Guerriero et al. proposed a task allocation approach for drones (i.e., agents) (Guerriero 2014). In their approach, Guerriero et al. employ the graph theory to model the task allocation problem and use the MOLP to achieve efficient task allocation of drones. In addition, Guerriero et al. use the time window to ensure that tasks can be completed on time. According to their experiments, the approach proposed by Guerriero et al. has good performance on the single to single task allocation problem. However, since Guerriero et al. do not consider the cooperation of drones, their approach cannot handle complex tasks in disaster rescues. Be different with Guerriero et al.'s approach, the proposed models enable agents to form coalitions and employ the MOLP to generate task allocation plan for tasks and coalitions. By doing so, agents in the same coalition can cooperatively perform a task so as to enhance the capabilities of agents to handle complex tasks in disaster environments.

Su et al. proposed a dynamic task allocation approach for heterogenous agents in disaster rescue (Su 2016). In their approach, the disaster environment is first divided into different areas. Then, according to types of tasks in the areas, agents are allocated to the corresponding area based on their capabilities. Finally, to handle the dynamics of tasks and agents in the disaster environment, Su et al. propose a dynamic coordination mechanism to balance agents among areas. Su et al.'s approach considers different types of tasks and capabilities of agents. At the same time, their coordination mechanism has a good adaptability to dynamic disaster environments. However, their approach only aims to complete as many tasks as possible and do not other task allocation objectives in disaster rescues such as the completion effectiveness of tasks, the execution time of tasks, etc. Be different with Su et al.'s approach, the proposed models employ the MOLP to generate task allocation plan by considering effectiveness, efficiency and execution time of tasks so as to satisfy different objectives of task allocation in disaster environments.

Ramchurn et al. proposed a max-sum-based approach, which achieves task allocation in disaster environment in a decentralised manner (Ramchurn 2010). In Ramchurn et al.'s approach, agents exchange their utility function for task allocation and make their decisions for task allocation through adjusting relevant variables in the function. Based on the max-sum algorithm, agents can achieve a consistent task allocation plan through several turns of function exchange and adjustment. The Ramchurn et al.'s approach considers the time constraint in disaster environments. Since the computational complexity of their approach is less than many task allocation approaches, it is more suitable for task allocation in dynamic environments. However, Ramchurn et al.'s approach dose not consider the suitability between types of tasks and capabilities of agents. The decisions of task allocation is made only based on whether the workload of a task can be completed by agents. Be different with Ramchurn et al.'s approach, the proposed models evaluate the completion effectiveness of tasks based on the suitability between types of tasks and capabilities of agents.

Ramchurn et al. also proposed the mixed integer linear programming (MILP) based approach to achieve efficient task allocation by considering the temporal and spital constraints in disaster environments (Ramchurn 2010). In their approach, Ramchurn et al. enable agents to form coalitions so as to cooperatively perform complex tasks in disaster environments. However, since the Ramchurn et al.'s approach considers many constraints of task allocation in disaster environments, the computation complexity of their approach is high, which limits their applications in high dynamic disaster environments. By employing the multi-stage task allocation mechanism, the computational complexity of the proposed models is low, which makes them work well in dynamic environments.

8. Conclusion and Future Work

In this paper, two innovative models are proposed for task allocation in disaster environments. Based on the proposed models, agents can form coalitions to cooperatively perform complex tasks in disaster environments. During task allocation, the proposed models consider the effects of agents' capabilities on their task completion. In addition, the proposed models can achieve efficient and effective task allocation for heterogenous tasks by evaluating multiple task allocation objectives. Finally, the proposed models can suit the dynamic task allocation environments through employing multi-stage task allocation mechanism. The experiments indicate the flexibility and adaptability of the proposed model on dynamic task allocation in disaster environments. In the future, we will improve the adaptability of the proposed models by adding new task allocation objectives.

9. Acknowledgments

The work is supported by the National Natural Science Foundation of China (Grants No. 61402449, 91546111, 91646201, 61703013), and the Key Project of Beijing Municipal Education Commission (Grants No. KZ201610005009).

References

[1] Akter, S.& Wamba, S. F. (2017). Big data and disaster management: a systematic review and agenda for future research. Annals of Operations Research (9), 1-21.

[2] Athukorala, P.& Resosudarmo, B. P. (2005).

The indian ocean tsunami: economic impact, disaster management and lessons. Departmental Working Papers (1).

[3] Banipal, K. S. (2005). Framework for strategic disaster management: lessons learned from hurricane katrina. Social Science Electronic Publishing.

[4] Chapman, A. C., Micillo, R. A., Kota, R., & Jennings, N. R. (2009). Decentralised dynamic task allocation: a practical game: theoretic approach. International Conference on Autonomous Agents and Multiagent Systems (Vol.2, pp.915-922). International Foundation for Autonomous Agents and Multiagent Systems.

[5] Davare, A., Chong, J., Zhu, Q.& Densmore, D. M. (2006). Classification, customization, and characterization: using milp for task allocation and scheduling.

[6] Farinelli, A., Rogers, A.& Jennings, N. R. (2014) Agent-based decentralised coordination for sensor networks using the max-sum algorithm. Autonomous Agents and Multi-Agent Systems, 28(3), 337-380.

[7] Farinelli, A., Rogers, A., Petcu, A.& Jennings, N. R. (2008). Decentralised coordination of low-power embedded devices using the max-sum algorithm. International Joint Conference on Autonomous Agents and Multiagent Systems (Vol.2, pp.639-646). IFAAMAS

[8] Gerkey, B. P.& Mataric, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research, 23(9), 939-954.

[9] Guerriero, F., Surace, R., Loscrí, V.& Natalizio, E. (2014). A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows constraints. Applied Mathematical Modelling, 38(3), 839-852.

[10] Hooshangi, N.& Alesheikh, A. A. (2017). Agent-based task allocation under uncertainties in disaster environments: an approach to interval uncertainty. International Journal of Disaster Risk Reduction, 24, 160-171.

[11] Jalil, A. M.& Hamouda, W. (2016). Optimal solution for a joint power allocation and relay assignment problem. IEEE Transactions on Vehicular Technology, 65(8), 6497-6507.

[12] Koes, M., Nourbakhsh, I.& Sycara, K. (2005). Heterogeneous multirobot coordination with spatial and temporal constraints. National Conference on Artificial Intelligence (pp.1292-1297). AAAI Press.

[13] Li, J. (2009). National disaster management system and its function in emergency response and disaster relief of "5.12" sichuan earthquake. Environment & disaster Mitigation, 8-10.

[14] Mulcaire, C. & Overington, B. (2013). Federal disaster recovery funding: minimizing roadblocks to maximize resources. International Economic Development council.

[15] Nie, Y.& Zhang, H. M. (2010). Solving the dynamic user optimal assignment problem considering queue spillback. Networks & Spatial Economics, 10(1), 49-71.

[16] Predrag T. Tošić, & Agha, G. A. (2004). Maximal Clique Based Distributed Coalition Formation for Task Allocation in

Large-Scale Multi-agent Systems. International Conference on Massively Multi-Agent Systems (Vol.3446, pp.104-120). Springer-Verlag.

[17]Ramamritham, K., Stankovic, J. A.& Zhao, W. (1989). Distributed scheduling of tasks with deadlines and resource requirements. IEEE Transactions on Computers, 38(8), 1110-1123.

[18]Ramchurn, S. D., Farinelli, A., Macarthur, K. S.& Jennings, N. R. (2010). Decentralized coordination in robocup rescue. Computer Journal, 53(9), 1447-1461.

[19]Ramchurn, S. D., Polukarov, M., Farinelli, A., Truong, C., & Jennings, N. R. (2010). Coalition Formation with Spatial and Temporal Constraints. International Conference on Autonomous Agents and Multiagent Systems (Vol.3, pp.1181-1188). DBLP.

[20]Rogers, A., Farinelli, A., Stranders, R.& Jennings, N. R. (2011). Bounded approximate decentralised coordination via the max-sum algorithm. Artificial Intelligence, 175(2), 730-759.

[21]Rueda-Medina, A. C., Franco, J. F., Rider, M. J., Padilha-Feltrin, A.& Romero, R. (2013). A mixed-integer linear programming approach for optimal type, size and allocation of distributed generation in radial distribution systems. Electric Power Systems Research, 97(1), 133-143.

[22]Shehory, O.& Kraus, S. (1998). Methods for task allocation via agent coalition formation. Artificial Intelligence, 101(1–2), 165-200.

[23]Smith, S. F., Gallagher, A.& Zimmerman, T. (2007). Distributed management of flexible times schedules. International Joint Conference on Autonomous Agents and Multiagent Systems (pp.74). ACM.

[24]Su, X., Zhang, M.& Bai, Q. (2014). Decentralised Task Allocation Under Space, Time and Communication Constraints in Disaster Domains. Smart Modeling and Simulation for Complex Systems. Springer Japan.

[25]Su, X., Zhang, M.& Bai, Q. (2015). Dynamic task allocation for heterogeneous agents in disaster environments under time, space and communication constraints. Computer Journal, 58(8).

[26]Su, X., Zhang, M.& Bai, Q. (2016). Coordination for dynamic weighted task allocation in disaster environments with time, space and communication constraints. Journal of Parallel & Distributed Computing, 97, 47-56.

[27]Su, X., Zhang, M., Ye, D. & Bai, Q. (2014). A dynamic coordination approach for task allocation in disaster environments under spatial and communicational constraints. In AAAI2014 workshop.

[28]Szabo, C.& Kroeger, T. (2012). Evolving multi-objective strategies for task allocation of scientific workflows on public clouds. Evolutionary Computation (Vol.425, pp.1-8). IEEE.

[29]Tanzi, T. J., Sebastien, O.& Harivelo, F. (2017). Towards a collaborative approach for disaster management using radio science technologies. Ursi Radio Science Bulletin, 87(1), 25-36.

[30]Tsarouchi, P., Michalos, G., Makris, S., Athanasatos, T., Dimoulas, K.& Chryssolouris, G. (2017). On a human–robot workplace design and task allocation system.

International Journal of Computer Integrated Manufacturing, 1-8.

[31]Yin, P. Y., Yu, S. S., Wang, P. P., & Wang, Y. T. (2007). Multi-objective task allocation in distributed computing systems by hybrid particl`e swarm optimization. Applied Mathematics & Computation, 184(2), 407-420.

[32]Zhang, B.& Peng, J. (2013). Uncertain programming model for uncertain optimal assignment problem. Applied Mathematical Modelling, 37(9), 6458-6468.

Xing Su received his BSc in software engineering from Beijing University of Technology, China in 2007. He received his MSc and PhD in computer science from University of Wollongong, Australia in 2012 and 2015, respectively. From 2016, he works a lecturer in the Faculty of Information at the Beijing University of Technology in Beijing, China. His research interests include distributed artificial intelligence, multi-agent systems, disaster management and service-oriented computing.

Yuechen Wang received her BSc in computer science and technology from Hebei Normal University, China in 2016. From 2016, Yuechen studys as a postgraduate in College of Computer Science and Technology in Beijing University of Technology, China. Her research interests include multi-agent systems, disaster management, computer vision and affective computing.

Xibin Jia, born in 1969, received Ph.D. degree in computer science and technology from Beijing University of Technology in 2007, M.S. degree in intelligent instrument from North China Institute of Technology in 1996 and B.S. degree in wireless technology from Chongqing University in 1991. She is a Professor in the Faculty of Information at the Beijing University of Technology in Beijing, China. Her areas of interest include visual information cognition, multi-source information fusion and intelligent artification.

Limin Guo is a lecturer at the Beijing University of Technology. Her research interests include database research and implementation, spatial-temporal data mining, etc. She received her bachelor's degree from Huazhong University of Science and Technology in 2005 and PhD degree in the Institute of Software, Chinese Academy of Sciences in 2012.

Zhiming Ding is a professor at the Beijing University of Technology, China. His main research interests include database systems, mobile & spatialtemporal data management, Intelligent Transportation Systems, sensor data management, and information retrieval. He received his bachelor's degree, masters degree, and Ph.D. degree from Wuhan University (1989), Beijing University of Technology (1996), and Institute of Computing Technology, Chinese Academy of Sciences (2002), respectively. Before joining Beijing University of Technology in August 2014, he worked with the Institute of Scientific & Technical Information, Ministry of Communications of China (1989–1993), SINOCHEM (1996–1999), FernUniversitt in Hagen, Germany (2002–2004), and Institute of Software, Chinese Academy of Sciences (2004–2014). He owns 5 invention patents, and has published 3 books and about 110 papers in academic journals and conferences.