An Innovation Approach for Optimal Resource Allocation in Emergency Management

Jihang Zhang, Minjie Zhang, Senior Member, IEEE, Fenghui Ren, and Jiakun Liu

Abstract—In metropolitan regions, emergency events with different severity levels usually require multiple resources that have appropriate functionalities, money expenditure, moving velocities, etc. These resources could distribute over an extensive area with different ownerships. Solving the resource allocation problem for such an event involves complicated collaboration of multiple emergency departments under strict time constraints. Traditional resource allocation approaches usually have difficulties to efficiently find out the best resource assignment within the time limits by considering the large number of possibilities, which result in a considerable increase in fatalities. In this paper, a multiagent-based decentralised resource allocation approach using the domain transportation theory is proposed to handle a multi-task emergency event. The proposed approach is designed to effectively select appropriate resources without the global information and to concurrently generate the resource deployment plans for multiple tasks by considering the severity level of an emergency event. In the experiments, the proposed approach is tested along with other related approaches, and the experimental results indicate that the proposed approach can efficiently generate the optimal solution in terms of resource allocation time and money expenditure.

Index Terms—Resource allocation, emergency management, multi-agent system, domain transportation

1 INTRODUCTION

As the continuous rise of the population densities around urban areas, the concern about the damage and casualty caused by emergency events becomes increasingly severe. For example, according to the Australian Bureau of Statistics [1], there are about 9,123 Australian people lost their lives due to emergency accidents in 2011 along. Generally, emergency events that occur in metropolitan regions refer to incidents, caused by the interaction of people with the environment or human systems, such as urban fire, industrial accidents, communication failures, acts of terrorism, etc. [11]. These events have the four common characteristics, including that (1) they are hard to be predicted, which makes resources per-allocation become almost impossible; (2) they usually require multiple resources that might be distributed over an extensive area with different usage costs, mobilities, availabilities, ownerships and functionalities, which complicates the resource searching progress; (3) they have strict time limits for emergency departments to response and allocate rescue resources, which increases the difficulty of finding the optimal resource allocation plan; and (4) the environments around these events and the events themselves are dynamic, which introduces many uncertain factors during the emergency resource allocation.

Currently, most resource allocation processes for the emergency events in metropolitan regions are still operated manually, which is highly inefficient. This is because in a metropolitan region, large numbers of rescue resources from different emergency departments or service providers are distributed at different locations. Operators of emergency events usually have difficulties to efficiently find out the optimal resource allocation for an emergency event due to a large number of possibilities. For example, in Sydney region, there are 1,500 vehicles, which provide ambulance services with more than 4,000 ambulance operators; about 338 fire stations with over 6,500 fire fighters; and over 6,500 police offices working in about 300 police stations [2]. Selecting resources from such a large resource pool, it could easily make possible human mistakes due to the evaluation of many factors in a very short time.

It is no doubt that public emergency departments suffer significantly on the pressure about how to efficiently and effectively allocate rescue resources for emergency events to reduce casualties. One potential solution of solving such problem is agent and multi-agent technologies. Generally, an agent refers to a software system, which is capable of perceiving, reasoning, acting and communicating with others to achieve its goals. A multi-agent system (MAS) refers to a computerised system that consists of multiple loosely-coupled autonomous agents, which usually need to communicate with each other to accomplish common objectives in a shared environment [30]. In metropolitan regions, emergency events usually require multiple resources that belong to different emergency departments and are distributed over a large area. Therefore, multiple emergency departments need to cooperate together and each of them might be responsible for different resource allocation tasks. Agents and multi-agent technologies can offer promising solutions for solving such a complex resource allocation problem.
problem due to their abilities of autonomous reasoning, intelligent modelling, management, dynamic reaction, collective decision making and group collaboration [4].

In order to effectively address resource allocation problems that can occur in different phases of emergency management, many promising agent-based resource allocation strategies have been proposed in recent years, such as game theory-based resource allocation [13], [27], predication-based resource allocation [6], [8], [15], [22], auction-based resource allocation [17], [23], [24], [25] and centralised-based resource allocation [7], [10], [31]. However, all of these approaches might have different emphasis and may have difficulty to be applied to emergency events’ resource allocation in metropolitan regions. More specifically, game theory-based approaches are mainly used to allocate security resources to reduce the risk of terrorist attacks and predication-based approaches are mainly used to pre-allocate rescue resources for large-scale disasters. These two types of approaches are not suitable for resource allocation of emergency events since these events usually happen randomly in urban areas, which cannot be predicated or prevented in advanced. Auction-based approaches are primarily used to coordinate distributed agents to generate resource allocation plans in decentralised manners. In auction-based approaches, the relationships between agents are usually competitive, which are not suitable for modelling the cooperative relationships between different emergency departments in the real world. Centralised-based approaches are good to be used to generate the optimal resource allocation plan based on the global information of all resources in an environment. However, acquiring global the accurate information of all available resources could be extremely difficult in most real life situations by considering the time constraint.

To overcome the above limitations of existing approaches, this paper proposes an agent-based decentralised resource allocation approach for an emergency event in metropolitan regions. The overall motivation of the proposed approach is to use decentralised autonomous agents to intelligently generate an optimal resource allocation proposal for an emergency event based on the event’s attributes, such as event severity, content, required emergency services and so on. By using this optimal resource proposal to deploy rescue resources, local emergency departments in a metropolitan region can not only reduce the average emergency response time, but also rationalise the resources money expenditure on a particular event. More precisely, the proposed approach first converts the resource allocation problem of a single emergency event to different resource allocation tasks. Then, multiple agents simultaneously propose their resource allocation proposals to these tasks. Finally, for each task, the domain transportation theory is used to combine these proposals and generate the optimal solution that minimises the total allocation cost.

The major contributions of this paper include that (1) the proposed approach is designed to effectively allocate distributed resources with different attributes in a decentralised manner. Comparing with centralised manner, the proposed approach is more applicable and practical to be used to handle emergency responses that involve the collaboration of multiple emergency departments and private resource companies; (2) the proposed MAS framework is designed to simultaneously allocate different types of resources to multiple interrelated tasks, which is more efficient comparing with traditional sequential task allocation approaches and (3) the proposed resource allocation algorithm is capable of selecting resources with appropriate allocation time and money expenditures and generating an optimal resource allocation proposal according to an emergency event’s severity, which can not only minimise the resource deployment time, but also improve the resource utilisation in terms of money expenditure.

The rest of this paper is organised as follow. Section 2 gives the problem description and definitions. Section 3 introduces the theoretical foundation of the optimal resource allocation. Section 4 describes the agent-based solution and the implementation of the proposed resource allocation approach. Section 5 shows the experimental results and provides analysis. Section 6 demonstrates the proposed approach in a case study. Section 7 gives related work of agent-based resource allocation for emergency events. Section 8 concludes the paper and outlines the future work.

2 PROBLEM DESCRIPTION AND DEFINITIONS

This section introduces the important definitions used in the proposed resource allocation approach and the description of the fundamental problem that the proposed approach is trying to address.

2.1 Definitions of Domain Knowledge

Definition 1 (Environment). An environment is represented by a city map, which is defined as an undirected graph, \( G = (V, E) \).

- \( V = \{v_1, v_2, \ldots, v_i\} \) is a set of nodes, which represent important locations in a metropolitan region.
- \( E = \{e_1, e_2, \ldots, e_j\} \) is a set of edges, which represent the paths between the nodes. \( e_j \) is further defined as a two-tuple, \( e_j = (v_o, v_p) \), and \( v_o, v_p \in V \) are the nodes that be connected by \( e_j \).

Definition 2 (Resource). A resource is defined as a seven-tuple, \( res = (nam, rty, ser, fun, rlo, ava, vel, exp) \), where

- \( nam \) represents the name of the resource.
- \( rty \in \{facility, mobile\} \) represents the type of resources. facility refers to immovable rescue resources, such as fire stations, hospitals, etc, and mobile refers to rescue vehicles and personnel.
- \( ser \in \{fire & rescue, medical, police\} \) represents the type of emergency services that \( res \) can provide.
- \( fun \in \{hospitalisation, ambulance transport, fire fighting, police support\} \) represents \( res \)’s functionality.
- \( rlo \in V \) represents \( res \)’s current location.
- \( vel \in (0, +\infty) \) represents \( res \)’s average velocity in kilometre per hour (km/h).
- \( exp \) represents \( res \)’s money unit.

In the proposed approach, set \( REE \) indicates all resources in \( G \). Besides, it is assumed that the money expenditure
exp of res is known by local emergency departments. Furthermore, the set of resource services and functionalities defined above could be extended in real-world applications.

Definition 3 (Task). A task is defined as a three-tuple, \( \text{tas} = (\text{dea}, \text{ser}, \text{TR}) \), where

- \( \text{dea} \) represents the deadline of the resources required by tas to arrive at the event position.
- \( \text{ser} \) represents the type of emergency service that tas requires to make response.
- \( \text{TR} = \{t_1, t_2, \ldots, t_e\} \) represents a set of required resources for completing \( \text{tas} \) and \( t_e = \{\text{rty, fun}\} \).

In the proposed approach, it is assumed that local emergency departments have the knowledge to estimate \( \text{dea} \) based on an event’s severity and content.

Definition 4 (Event). An event is defined as a five-tuple, \( \text{eve} = (\text{con}, \text{SER}, \text{elo}, \text{sev}, \text{TAS}) \), where

- \( \text{con} \in \{\text{fire, rescue, loss of life, damage to health, security of person, security of property}\} \) represents eve’s content.
- \( \text{SER} \subseteq \{\text{fire & rescue, medical, police}\} \) represents a set of emergency services required by eve.
- \( \text{elo} \in \mathbb{V} \) represents eve’s location.
- \( \text{sev} \in \{1, 2, 3, 4, 5\} \) represents eve’s severity, where 1 indicates the lowest severity and 5 indicates the highest severity.
- \( \text{TAS} = \{\text{tas}_1, \text{tas}_2, \ldots, \text{tas}_k\} \) represents a sequence of tasks that need to be completed for eve.

Definition 5 (Resource Allocation Proposal). A resource allocation proposal for an event is defined as a two-tuple, \( \text{rap} = (\text{eve}, \text{RES}) \), where

- \( \text{eve} \) represents an emergency event.
- \( \text{RES} \subseteq \text{REE} \) represents a set of resources that are proposed for completing tasks.

Besides, a resource allocation proposal for a single task in the event is defined as a two-tuple, \( \text{rap}_k = (\text{tas}_k, \text{RES}) \) and \( \text{rap}_k.\text{RES} \subseteq \text{rap}.\text{RES} \).

In the proposed approach, the cost of resource allocation is calculated by a cost function, which is used to determine the optimal resource allocation proposal (i.e., the proposal with the minimum cost). Usually, different emergency events might need to use different cost functions, which might involve different cost attributes. Here, a cost function is defined, by the consideration of two significant factors of resource allocation in emergency management, i.e., money expenditure and time [28]. More specifically, there is no doubt that the arrival time of emergency services is crucial for the mitigation of the damage caused by an emergency event. Apart from the resource allocation time, money expenditure also plays a non-ignorable role in emergency management, since the budget for an emergency department is limited in terms of managing large amount of rescue resources to address unlimited emergency events. Therefore, improving the reasonability and effectiveness of money usage in emergency management means that more emergency events can be processed and more lives can be saved. Apparently, money expenditure and time should have different significance for different events. In the proposed resource allocation approach, the importance of money expenditure and time is determined by the severity of an event (i.e., \( \text{eve}.\text{ser} \)). For an high severity event, the proposed approach will consider resource allocation time as a more important attribute compared with money expenditure in the cost function. On the contrary, for an event with a low severity, resource money expenditure will play a more important role in the cost function.

In the following, the cost for allocating a single resource to a single task is firstly defined.

Definition 6 (Cost Function). A cost function for a single resource allocation is defined as follows:

\[
\text{COR}(\text{eve}.\text{tas}_k, \text{res}) = \begin{cases} 
E^m(\text{res.exp}), & \text{if res.rty = facility} \\
\left( w^f E^r \left( \frac{\text{DIS}(\text{res}.\text{rlo}, \text{elo})}{\text{res}.\text{rlo}} \right) + w^m E^m(\text{res.exp}) \right) \times \text{DLINE}(\text{eve}.\text{tas}_k, \text{res}), & \text{if res.rty = mobile,}
\end{cases}
\]

where \( \text{DIS}(\text{res}.\text{rlo}, \text{elo}) \) is a function that returns the distance of a passable road between resource location \( \text{res}.\text{rlo} \) and event location \( \text{eve}.\text{elo} \), which could be implemented by various path searching algorithms, such as A* algorithms [32]. \( E^m(\text{res.exp}) \) and \( E^r \left( \frac{\text{DIS}(\text{res}.\text{rlo}, \text{elo})}{\text{res}.\text{rlo}} \right) \) are two evaluation functions that convert the value of a resource’s money expenditure (i.e., \( \text{res.exp} \)) and allocation time (i.e., \( \text{DIS}(\text{res}.\text{rlo}, \text{elo}) \)) to a normalised value in between 0 and 1, respectively. These evaluation functions can be implemented based on various normalisation approaches, such as feature scaling [3]. \( w^f \) and \( w^m \) represent the weighting (importance) of the resource’s money expenditure and allocation time, respectively, which are calculated by following equations:

\[
\begin{align*}
w^f &= \frac{\text{COR}_{\text{mobile}}}{{\text{COR}_{\text{mobile}}} + \text{COR}_{\text{facility}}} \\
w^m &= 1 - w^f
\end{align*}
\]

Besides, \( \text{DLINE}(\text{eve}.\text{tas}_k, \text{res}) \) is a function that is used to determine whether \( \text{res} \) can be allocated within \( \text{eve}.\text{tas}_k \)'s deadline, which is further defined as follows:

\[
\text{DLINE}(\text{eve}.\text{tas}_k, \text{res}) = \begin{cases} 
1, & \text{if} \quad \frac{\text{DIS}(\text{res}.\text{rlo}, \text{elo})}{\text{res}.\text{rlo}} \leq \text{eve}.\text{tas}_k.\text{dea} \\
+\infty, & \text{if} \quad \frac{\text{DIS}(\text{res}.\text{rlo}, \text{elo})}{\text{res}.\text{rlo}} > \text{eve}.\text{tas}_k.\text{dea}.
\end{cases}
\]

Based on above terms, the cost function for allocating all required resources to a single task is defined as follows:

\[
\text{COT}(\text{eve}.\text{tas}_k, \text{rap}_k) = \sum_{\text{res} \in \text{rap}_k.\text{RES}} \text{COR}(\text{eve}.\text{tas}_k, \text{res}).
\]

Furthermore, the cost function for allocating all required resources to all tasks in an event is defined as follows:

\[
\text{COE}(\text{eve}, \text{rap}) = \sum_{\text{eve}.\text{tas}_k \in \text{eve}.\text{TAS}} \text{COT}(\text{eve}.\text{tas}_k, \text{rap}_k) \text{ subject to } \text{rap}_k.\text{RES} \subseteq \text{rap}.\text{RES}.
\]
2.2 Problem Description

For an emergency event \( \text{eve} \), there could be different resource allocation proposals. In the proposed approach, the all possible proposals for \( \text{eve} \) are represented as a set \( \text{RAP} \). The main objective of the proposed approach is to search an optimal resource allocation proposal \( \text{rap}^* \in \text{RAP} \) for \( \text{eve} \), which is formally defined as follows:

\[
\text{OBJE} = \arg \min_{\text{rap}^* \in \text{RAP}} \text{COE}(\text{eve}, \text{rap}^*) \quad \text{subject to} \quad \text{rap}^* \in \text{REE},
\]

where \( \text{rap}^* \in \text{REE} \) means that the proposed resources in \( \text{rap}^*.\text{RES} \) must belong to the available resources in the environment \( G \). The event objective function indicates that the optimal resource allocation proposal \( \text{rap}^* \) must have the minimum allocation cost in \( \text{RAP} \). Besides, the proposed approach assumes there is always enough resources in \( \text{REE} \) to be allocated for \( \text{eve} \).

However, due to the fact that an emergency event usually requires resources with different types and functionalities, searching a complete \( \text{rap}^* \) could be a complicated process. In order to efficiently solve this searching problem, the proposed approach creates a set of tasks \( \text{eve}.\text{TAS} \) for \( \text{eve} \). For each task \( \text{tas}_k \) in \( \text{eve}.\text{TAS} \), it only requires resources that provide the same type of emergency service (i.e., \text{res.ser}). For example, for a vehicle fire accident with two tasks, \( \text{tas}_1 \) only require resources that provide medical service (i.e., ambulances and hospitals), and \( \text{tas}_2 \) only require resources that provide fire & rescue services (i.e., fire engines and fire fighters). Further more, each task \( \text{tas}_k \) in \( \text{eve}.\text{TAS} \) has a different or same resource deployment deadline (i.e., \text{tas}_k.\text{dea}) as described in Definition 3.

Concerning functionality \( \text{res} \), for each \( \text{tas}_k \) we denote the required resource at task \( \text{tas}_k \) by \( \text{res}_k \in \text{RES} \), which can be regarded as a vector-valued function on the set of nodes \( V \). Let \( \text{RES}_e(y_i) \) denote the amount of resource with functionality \( e \) at location \( y_i \in V \). Naturally, \( \sum_{y_i \in V} \text{RES}_e(y_i) \) denotes the total amount of type \( k \) resource in \( G \).

To solve Equation (6), it suffices to consider a single task, namely Equation (7). For each \( \text{tas}_k \), it requires a set of resources \( \text{TR} = \{ \text{tr}_1, \text{tr}_2, \ldots, \text{tr}_r \} \), which might have different functionalities, but are provided by a same emergency service. Let \( x_k = \text{eve}.\text{tas}_k \) and \( \text{tr}_r(x_k) \) represent the required amount of functionality \( e \) resource at task \( x_k \). Then \( \sum_{\text{tas}_k \in \text{TAS}} \text{tr}_r(x_k) \) denotes the total amount of functionally \( e \) resource in such an event. As we only consider a single event, usually the resource is sufficient, namely

\[
\sum_{x_k \in \text{TAS}} \text{tr}_r(x_k) \leq \sum_{y_i \in V} \text{RES}_e(y_i).
\]

Concerning functionality \( e \) resource, for each \( k \) and \( j \), the cost of transferring the resource at \( y_j \) to the task \( x_k \) is given by Equation (1), i.e., \( \text{COR}(x_k, y_j) \). An admissible allocation proposal \( \text{rap}_{k} \) is a mapping from \( V \) to \( \text{tas}_k.\text{TR} \) satisfying the balance condition that for any subset \( E \subset \text{TAS} \)

\[
\sum_{x_k \in E} \text{tr}_r(x_k) = \sum_{y_j \in \text{rap}_{k}^{-1}(E)} \text{RES}_e(y_j),
\]

where \( \text{rap}_{k}^{-1} \) is the inverse mapping of \( \text{rap}_{k} \). This balance condition means during the allocation process of the
optimal proposal, the resources in the proposal are neither increase nor decrease. The cost of implementing the proposal \( \text{rap}_k \) is given by Equation (4), which is equivalent to
\[
\text{COT}(\text{rap}_k) = \sum_{x_i \in \text{TAS}} \text{COR}(x_i, \text{rap}_k^{-1}(x_i)).
\]

The purpose of the objective function of an task, i.e., Equation (7) is to find an optimal proposal \( \text{rap}_k \in \text{RAP}_k \) such that
\[
\text{COT}(x_k, \text{rap}_k^*) = \min_{\text{rap}_k \in \text{RAP}_k} \text{COT}(x_k, \text{rap}_k). \quad (10)
\]

From optimal transport theory, Equation (10) can be transferred to the following linear programming
\[
\max \left\{ \sum_{x_k \in \text{TAS}, y_i \in \mathcal{V}} u(x_k)tr_e(x_k) + v(y_i)\text{REE}_e(y_j) : u(x_k) + v(y_j) \leq \text{COR}(x_k, y_j) \right\}.
\]

Moreover, there exists a maximiser \( (u^*(x_k), v^*(y_j)) \) (unique up to a constant) achieving the above maximum, which represents the optimal allocation for task \( x_k \) by using the resource at location \( y_j \). \( u^*(x_k) \) is called a potential, and \( v^*(y_j) \) is its dual potential. The pair \( (u^*(x_k), v^*(y_j)) \) also satisfies a generalised Legendre duality associated with the cost function \( \text{COR} \). Hence, for each \( x_k \) there exists an unique \( y_j \) such that the equality holds in the constraint, namely \( u^*(x_k) + v^*(y_j) = \text{COR}(x_k, y_j) \), which means the task \( x_k \) requires the resource from location \( y_j \). Thus, we can construct a mapping \( \text{inv}^*_k : x_k \in \text{TAS} \rightarrow y_j \in \mathcal{V} \). From optimal transport theory [16], by differentiating the above equation we can see that
\[
\nabla u^*(x_k) = \nabla_y \text{COR}(x_k, \text{inv}^*_k(x_k)), \quad y_j = \text{inv}^*_k(x_k)
\]
\[
= \nabla_x^{-1} \text{COR}(x_k, \nabla u^*(x_k)). \quad (12)
\]

In fact, we can further show that \( \text{inv}^*_k \) is exactly the inverse of the optimal allocation proposal \( \text{rap}_k^* \).

Therefore, to construct \( \text{rap}_k^* \), it suffices to follow the following steps:

1. From the given data \( \{tr_e, \text{TAS}, \text{REE}_e, \mathcal{V}\} \), formulate the linear programming Equation (11);
2. Solve Equation (11) to find out a potential \( u^* \);
3. Use Equation (12) to construct the mapping \( \text{inv}^*_k : x_k \in \text{TAS} \rightarrow y_j \), which implies that task \( x_k \) requires the resource from the area \( \text{REQ} := \text{inv}^*_k(x_k) \subset \mathcal{V} \); and
4. Take the inverse, we obtain the optimal allocation proposal \( \text{rap}_k^* : \text{REQ} \rightarrow x_k \), which can inform the agent how to distribute the resource in the optimal way.

As a remark, we point out that when the cost function \( \text{COR} \) satisfies the condition that \( \nabla_x \text{COR}(x_k, \cdot) \) is injective at each \( x_k \) from the uniqueness of potential \( u^* \) (up to a constant) and relation (12), the optimal allocation proposal \( \text{rap}_k^* \) is also unique.

4 Agent-Based Decentralised Resource Allocation

The proposed resource allocation approach is implemented based on agent and multi-agent technologies. This section gives detailed description of agents’ definitions, implementation, resource allocation framework and process.

4.1 Definitions of Agents and Implementation

There are four types of agents in the proposed MAS, which are response agents, mobile agents, facility agents and deployment agents. The four types of agents’ are defined as follows.

Definition 7 (Response Agent). A response agent is represented by \( ra \), which has the information of a specific emergency event. A response agent has four major functionalities, including:

1. Identifying event content \( \text{eve}.\text{con} \) for a new event \( \text{eve} \);
2. Identifying the emergency services \( \text{eve}.\text{SER} \) that is required by \( \text{eve} \) based on \( \text{eve}.\text{con} \);
3. Identifying a set of tasks \( \text{eve}.\text{TAS} \) for \( \text{eve} \) based on \( \text{eve}.\text{ser} \); and
4. Sending \( \text{eve}.\text{TAS} \) to a deployment agent.

Definition 8 (Deployment Agent). A deployment agent is represented by \( da \), which has the information of a specific emergency event. A deployment agent has three major functionalities, including:

1. Informing an event’s tasks information (i.e., \( \text{eve}.\text{TAS} \)) to facility agents that are located in its circle communication area, represented by \( da.\text{com} \);
2. Combining and generating the optimal resource allocation proposal \( \text{rap}_k^* \) for a task based on a set of proposals (i.e., \( \text{RAP}_k \)) submitted by facility agents; and
3. Informing relevant facility agents to execute \( \text{rap}_k^* \).

Definition 9 (Facility Agent). A facility agent is represented by \( fa \), which has the information of a specific facility resource \( \text{fa}.\text{res} \) and a set of mobile agents. More precisely, \( \text{fa}.\text{MA} = \{ma_1, ma_2, \ldots, ma_t\} \) represents a set of mobile agents that belong to \( fa \) and \( \text{REE} = \{\text{fa}.\text{res}\} \cup \{ma_1.\text{res}, ma_2.\text{res}, \ldots, ma_t.\text{res}\} \) represents all resources under \( fa \)’s management. A facility agent has three major functionalities, including:

1. Managing a facility resource \( \text{fa}.\text{res} \);
2. Generating resource allocation proposals for tasks based on \( \text{fa}.\text{REE} \); and
3. Informing its mobile agents to execute resources allocation commands after receiving the confirmation from a deployment agent.

Definition 10 (Mobile Agent). A mobile agent is represented by \( ma \), which has the information of a specific mobile resource \( ma.\text{res} \). A mobile agent has two major functionalities, including:

1. Managing a mobile resource \( ma.\text{res} \); and
2. Implementing resource deployment plans after receiving resources allocation commands.

In the proposed resource allocation approach, Java Agent Development Framework (JADE) is used for the
implementation of the above agents [5]. The major reason of choosing JADE as the agent platform is that it provides runtime environment agent registration and operation, which is extremely important for allocating resources to unexpected emergency events in open and dynamic environments. Besides, the Belief-Desire-Intention (BDI) architecture is used as the agents’ implementation architecture [19]. In BDI architecture, an agent pursues its predefined goals (desire) by adopting appropriate plan (intentions) according to the current state of the agent’s environment (beliefs). One of the major advantages of using BDI architecture in emergency resource allocation is that it provides mechanisms for separating the agents’ activities of selecting a resource allocation plan from the execution of currently active plan. In addition to agent’s architecture, Knowledge Query and Manipulation Language (KQML) is used for agents to exchange information, which is an agent communication language and allows platform independent agent communication [9].

4.2 Resource Allocation Framework and Process
The proposed resource allocation approach is implemented by a MAS, which includes a task identification module, a resource identification module, a proposal generation module, an optimal allocation module and a proposal execution module. The framework of the MAS is depicted in Fig. 2.

As depicted in Fig. 2, in the task identification module, a single response agent is used to identify resource allocation tasks for an emergency event according to the emergency service required by this event. Then, the task identification module passes the tasks to the resource identification module, in which a deployment agent generates multiple task processing threads. Each of the task processing thread is used to identify the resource requirements specification in a task and pass this task to relevant candidate service providers (facility agents). After that, in the proposal generation module, each facility agent that has received a resource allocation task will use domain transportation theory to generate a resource allocation proposal based on all resources under its management. All the generated resource allocation proposals will be passed to the optimal allocation module, in which the deployment agent will combine these proposals and use domain transportation theory to generate an optimal resource allocation plan for a task. Finally, the optimal allocation plan is passed to the proposal execution module, in which relevant facility agents will confirm the detail of the resource allocation plan and inform relevant mobile agents to execute the plan accordingly.

The allocation process is formally described by Algorithm 1.
Algorithm 1: Resource Allocation Process

1: assign ra to eve
2: ra identifies eve.con
3: ra identifies eve.SER based on eve.con
4: ra identifies eve.TAS based on eve.SER
5: ra sends eve.TAS to da
6: da calculates circle communication area da.com
7: da locates FA in da.com
8: for all tas_k ∈ eve.TAS do
9: da creates RAP_k and FA_k
10: for all fa_i ∈ FA do
11: if fa_i.res.ser = tas_k.ser then
12: da updates FA_k = {fa_i} ∪ FA_k
13: da sends tas_k to fa_i
14: fa_i finds rap_k : fa_i.REF → tas_k.TR
15: if tcu ≤ PDLINE(tas_k.dea, eve.ser) then
16: fa_i submits rap_k to da
17: da updates RED_k = {rap_k} ∪ RAP_k
18: end if
19: end if
20: end for
21: end for
22: while |eve.TAS| > 0 do
23: for all tas_k ∈ eve.TAS do
24: if tcu ≥ PDLINE(tas_k.dea, eve.ser) ∨ ∀ fa_i ∈ FA_k :
25: fa_i submit rap_k then
26: if RAP_k does not contains enough resources for
27: tas_k then
28: da expand da.com = da.com × 2
29: the process goes back to Line 7
30: else
31: da finds rap_k : RED_k → tas_k.TR
32: da updates eve.TAS = eve.TAS \ {tas_k}
33: da informs agents to execute rap_k
34: end if
35: end if
36: end for
37: end while

The resource allocation process shown in Algorithm 1 includes six steps, which are explained as follows.

Step 1: (Lines 1–5). When an emergency event eve happens, a new response agent ra is assigned to eve to identify the emergency content eve.con. Then, ra needs to identify the emergency services eve.SER required by eve according to eve.con. For example, when eve.con = fire, the required emergency services could be eve.SER = {fire & rescue, medical, police}. After the emergency service identification, ra needs to acquire the event severity (i.e., eve.ser) and the resources required by each emergency service, which may be provided by human operators or other external agents. Then, ra converts each of emergency service to a task. Finally, ra sends eve.TAS to a deployment agent da.

Step 2: (Lines 6–13). After receiving eve.TAS, da first needs to calculate a communication area da.com, which is a circle centered at the event’s location eve.elo and measured by square kilometres. da.com is calculated as follows:

\[ da.com = \pi \times \left( \frac{avw \times \left( \sum_{tas_k \in eve.TAS} tas_k.dea - tcu \right)^2}{|eve.TAS|} \right) \]

where avw represents the average moving velocity (km/h) of all required mobile resources in eve.TAS and tcu represents the current time. In the proposed approach, it is assumed that local emergency departments have the knowledge of the average velocities of resources.

After the calculation of da.com, da needs to locate all facility agents inside da.com, represented by set FA = {fa_1, fa_2, ..., fa_n} (Line 9). Then, da sends each tas_k to relevant facility agents in FA based on tas_k’s emergency service requirement tas_k.ser. At the same time, da also creates a contact list for the facility agent FA_k = {fa_1, fa_2, ..., fa_n} (FA_k ∈ FA) and resource allocation proposal list RAP_k = {rap_k, rap_k', ..., rap_k''} for each tas_k in eve.TAS.

Step 3: (Lines 14–17). After a facility agent fa_i ∈ FA receives tas_k, fa_i uses the domain transportation theory (see Section 3) to calculate an optimal resource allocation proposal rap_k based on all available resources that are under fa_i’s management (Line 14). After the calculation of rap_k, fa_i submits rap_k to da if current time tcu has not exceeded the task proposal deadline. The task proposal deadline is calculated by function PDLINE(tas_k.dea, eve.ser), which can be defined by local emergency departments based on the detail of tas_k and eve. After the submission of rap_k, da adds rap_k to RAP_k.

Step 4: (Lines 23–25). After da receives tas_k’s resource allocation proposals from all facility agents in FA_k or tas_k’s proposal deadline has been reached, da checks whether RAP_k has enough resources for da to generate a final resource allocation plan to complete tas_k. If the resources are enough, the process goes to Step 6. Otherwise, the process goes to Step 5.

Step 5: (Lines 26–27). If RAP_k does not have enough resources to complete tas_k, da expands its original communication area da.com by doubling its size to be able to contact more facility agents, and then the process goes back to Step 2.

Step 6: (Lines 29–31). If RAP_k has enough resources to complete tas_k, da uses domain transportation theory (see Section 3) to generate an optimal resource allocation proposal rap_k for tas_k based on all resources in RAP_k, which is represented by RED_k (Line 24). Finally, da informs relevant facility agents to execute rap_k and remove tas_k from eve.TAS. If there are more tasks in eve.TAS, the process repeats Step 4, otherwise the process ends.

5 EXPERIMENT

In this section, experimental results are presented and the performance of the proposed resource allocation approach is analysed. The experiments focus primarily on the test of the resource allocation time, money expenditure and cost of an event when employing the proposed optimal resource allocation approach. The experimental results were compared with other related emergency resource allocation approaches. The rest of section is divided into two sections. Section 5.1 describes the experimental settings and Section 5.2 demonstrates the experimental results and gives performance analysis.
5.1 Experimental Setting

In order to effectively analyse the advantages and disadvantages of the proposed approach, the proposed approach is tested along with the other four related agent-based emergency resources allocation approaches for a single event allocation [7], [10], [17], [18]. Briefly, Chou et al.’s [7] approach was designed to allocate rescue resources to a large-scale emergency situations by using a genetic algorithm; Gabdulkhakova et al.’s [10] approach was designed to generate solutions for emergency medical allocation by using the service-oriented architecture; López et al.’s [17] approach was designed to deploy ambulances for emergency medical events by using a contract-net protocol and Ponda et al.’s [18] approach was designed to allocate tasks to heterogeneous agents by using a distributed auction mechanism. Generally, these four approaches can be classified into centralised approaches [7], [10] and decentralised approaches [17], [18]. In centralised approaches, a centralised agent can not only access the information of all distributed resources, but also directly control other agents on the map. Contrarily, in decentralised approaches, because resource information is limited to individual agents, an agent needs to communicate with other agents to exchange information during a resource allocation process. The summary of the four approaches is listed in Table 1.

In order to evaluate our approach in real world environments, Google Map is employed in the resource allocation tests. More precisely, an agent-based emergency resource allocation simulator was designed and implemented to conduct the experiments (Fig. 3). The resource allocation simulator has integrated Google Map technology, which allows us to search for the real world information of local facility resources in a particular city, such as the location of fire stations, hospitals and police stations. Since Google Map does not provide the information of mobile resources of a facility resource, the experimental parameters of mobile resources were randomly generated based on Table 2. The experimental parameters of emergency events and tasks were randomly generated based on the range values in Tables 3 and 4, respectively.

The resource allocation experiment for each approach was repeated for 1,000 times and the average resource allocation cost, time, money expenditure and solution calculation speed of an event were recorded. The resource allocation cost was calculated by Equation (1) and the results of the average resource allocation cost, time and money expenditure were normalised between 0 and 1 by their minimal and maximal values. A high value of the normalised result indicates high cost, while a low normalised value indicates low cost. In each time of test, the parameters of the emergency event and its tasks were regenerated, while the parameters of the resources remained the same.

### Table 1

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Approach</th>
<th>Resource Control</th>
<th>Allocation Method</th>
<th>Global Information</th>
<th>Optimisation Objective</th>
<th>Task Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gabdulkhakova et al. [10]</td>
<td>Centralised</td>
<td>Rational Allocation</td>
<td>Require</td>
<td>Money &amp; Distance</td>
<td>Sequential</td>
</tr>
<tr>
<td>2</td>
<td>Chou et al. [7]</td>
<td>Centralised</td>
<td>Genetic Algorithm</td>
<td>Require</td>
<td>Money &amp; Time</td>
<td>Sequential</td>
</tr>
<tr>
<td>3</td>
<td>López et al. [17]</td>
<td>Decentralised</td>
<td>Contract-net Protocol</td>
<td>Not Require</td>
<td>Time</td>
<td>Sequential</td>
</tr>
<tr>
<td>4</td>
<td>Ponda et al. [18]</td>
<td>Decentralised</td>
<td>Distributed Auction Protocol</td>
<td>Not Require</td>
<td>Money &amp; Time</td>
<td>Sequential</td>
</tr>
<tr>
<td>5</td>
<td>Our Approach</td>
<td>Decentralised</td>
<td>Domain Transportation</td>
<td>Not Require</td>
<td>Money &amp; Time</td>
<td>Concurrent</td>
</tr>
</tbody>
</table>
5.2 Experimental Result and Analysis

The experimental results are shown in Fig. 4, which are resource allocation time test shown in Fig. 4a, resource money expenditure test shown in Fig. 4b, resource allocation cost shown in Fig. 4c and the calculation time of the optimal solution shown in Fig. 4d.

As we can see from Fig. 4a, in the three decentralised approaches (i.e., López et al., Ponda et al. and our approach), the approach of López et al. required the shortest resource allocation time (0.26) for an emergency event in 1,000 tests. Our approach performed slightly better than Ponda et al.’s approach (0.36 versus 0.41). Nevertheless, Fig. 4b shows that López et al.’s approach cost significant more resource money expenditure than our and Ponda et al.’s approaches. This is mainly because in López et al.’s approach, resource allocation time is used as the only criterion to determine the optimal allocation of resources, but our and Ponda et al.’s approaches take both resource allocation time and resource money expenditure into consideration. Due to this reason, López et al.’s approach eventually required the most resource allocation cost in the three decentralised approaches. Furthermore, in Fig. 4c, we can find that the resource allocation cost of our approach is less than the approach of Ponda et al. (0.66 versus 0.76). The reason of such experimental results is that Ponda et al.’s resource allocation approach is based on auction mechanism and the relationships between contract agents are competitive. Therefore, the optimal resource allocation plan in Ponda et al.’s approach is the single solution provided by the winner agent of the auction. However, for most emergency situations, it is more reasonable for agents to act cooperatively rather than competitively. In our approach, a deployment agent uses the domain transportation theory to generate the optimal solution based on the resource allocation proposals from multiple facility agents, so the optimal solution is a combined solution by integrating the advantages of each facility agent’s proposal.

When it comes to the centralised approaches (i.e., Gabdulkhakova et al. and Chou et al.), the approach of Chou et al. spent less resource allocation cost than Gabdulkhakova et al.’s approach (0.58 versus 0.75). The major reason of such results is that the optimisation objectives of Gabdulkhakova et al.’s approach are resource distance and money expenditure, which result in some disadvantages of the optimisation of resource allocation time comparing with the approach of Chou et al. Comparing these two centralised approaches with our approach, Chou et al.’s approach performed slightly better than our approach in both resource allocation time and money expenditure. This is mainly because due to a central node used in their system to acquire all available resources information, but our system was implemented in a decentralised manner.

Finally, in Fig. 4d, we can find that our approach outperformed the other four approaches in term of the calculation time of the optimal solution (756 milliseconds), which is not...
only due to that the domain transportation theory used in our optimal allocation algorithm is based on linear programming method, but also because the resource allocation tasks are processed concurrently in our MAS. In contrast, Chou et al.’s approach required the longest time to generate the optimal solution for an event (2,832 milliseconds), which mainly due to the high computation cost of calculating the fitness value function for comparing the advantages and disadvantages of different chromosomes.

In summary, Chou et al.’s approach can efficiently generate the optimal allocation plans by using the centralised resource control system. However, it is almost impossible to access every available resources that come from different emergency departments or companies and control them by using one central node in most real-life situations. In contrast, our approach was designed to allocate resource in a decentralised manner. With the optimal resource mapping plan generated by the domain transportation theory, the resource allocation cost in our approach for an emergency department or company is very close to Chou et al.’s centralised approach.

6 Case Study

In the previous section, we demonstrated that the proposed resource allocation approach could effectively generate a resource allocation plan by considering both resource money expenditure and allocation time. In order to better understand how agents perform their actions during the process of resource allocation and how the proposed approach adjusts its resource allocation strategies in terms of emergency events with different severities, a case study is presented in this section. This case study has three major purposes, which are: (1) demonstrating agents’ activities and information flow during the resource allocation process; (2) observing the changes of resources selection of the proposed approach when the severity of an event is increased from 1 to 5; and (3) analysing how the proposed approach balances the importance of resource money expenditure and resource allocation time for an event with different severities.

6.1 Case Study Setting

The case study was also conducted on the Google Map. In detail, the emergency event used for the case study is a fire incident. The parameters of the emergency event are listed in Table 5. There are four resource allocation tasks in this event, in which Task 1 requires three fire engines, Task 2 requires two police cars, Task 3 requires two ambulances and Task 4 requires one hospital. The detail of tasks’ parameters are listed in Table 6. The facility resources used in the case study are selected from Google Map, which are two fire stations, two police stations and two hospitals (see Fig. 5). Each of the facility resource has three mobile resources under its management, with different moving velocity and money expenditure (see Table 7 for detail). In the case study, the proposed approach was used to allocate resources to the same emergency event for five times with different event severities (from 1 to 5). During each time of allocation, resources that had been selected by the proposed approach as the final resource allocation plan were recorded. Besides, the normalised values (between 0 and 1) of the actual resources’ money expenditure and allocation time (i.e., the values that have not been scaled by the weighting of money expenditure, time) were also recorded. Furthermore, an activity diagram by using Unified Modelling Language (UML) was provided to demonstrate agent’s activities and information outputs during each step of the resource allocation process [20]. Finally, the Pareto optimality test was conducted to verify whether the resource allocation solutions generated from our approach were belong to Pareto frontier [21], which is a widely used method for multi-objectives optimisation problems.

6.2 Case Study Results and Analysis

The results of case study are shown in Figs. 6, 7, and 8 and Table 8. Fig. 6 depicts how agents performed actions during the resource allocation for the Severity 1 emergency event. Fig. 7 demonstrates the normalised results of unweighted resource allocation time and money expenditure of the same emergency event with five different severity levels. Fig. 8 shows the result of the Pareto optimality test. Table 8 shows the resources that had been selected by the proposed approach as the optimal proposal for each level’s emergency event.

From Fig. 6, we can see that when the emergency event happened, a response agent ra first identified that the event content was fire and required three emergency services (fire & rescue, police and medical). Based on these services information, ra identified three tasks (t1, t2 and t3) and their required resources, then these tasks were sent to a deployment agent da. da calculated its initial communication range according to the resource requirements inside the tasks, which was 3,490 m², then da contacted relevant facility agents and informed them the tasks information. More specifically, t1 was sent to facility Fire Brigades-NSW and
Alexandria Fire Station, $t_2$ was sent to Sydney Mounted Police Stables and Waverley Police Station, and $t_3$ was sent to Eastern Suburbs Private Hospital and the Sydney Clinic. After receiving the information of these tasks, each facility agent used domain transportation method to select required resources from its controlled resource pool. Since the severity level of this emergency event is 1, the resources that has been selected by the facility agents were the resources with the lowest money expenditure. Finally, each facility agents submitted their proposals to $da$ and $do$ used domain transportation to combine these proposals to generate a final plan for each task.

Fig. 7 further demonstrates that when the severity of the emergency event equals to 1, the percentage of the cost generated by resource allocation time is much higher than the that of generated by the resource money expenditure (74 versus 26 percent). These results are reasonable since when an emergency event has a low severity level,
the resource money expenditure is much more important than that of resource allocation time in the cost function (refer to Equations (1) and (3)). Under such a situation, the proposed approach will preferentially consider to choose resources with low money expenditures to decrease the final resource allocation cost. In Fig. 7, with the increment of the event’s severity, it is apparently that the cost generated by the resource money expenditure has increased and the resource allocation time has started to generate less and less cost. Besides, we can also find that the combined allocation cost of the event rises moderately from severity 1 to 5. When it comes to the severity 5 event, the resource allocation time only occupations 22 percent of the combined cost, while the resource money expenditure occupies 78 percent of the combined cost. This is mainly because for high severity events, the resource allocation time becomes more important in the cost function, thus the proposed approach will try to select resources with high moving velocity and closer distance to the location of the emergency event. Although high velocity resources usually come with higher money expenditures and will increase the total cost slightly, however, in high severity events, it is no doubt that time is the top propriety we need to consider about. For example, in Table 8, we can see that the fire engines chosen to handle the emergency event with the severity level 5 are all from the facility resource “Fire Brigades-NSW”, which provides fire engines with the fast speed but high money expenditure to achieve efficient resource deployment.

Regarding to the optimality of the solutions, Fig. 8 indicates that the proposed approach can achieve the Pareto efficiency for the event in all five severity levels. More precisely, in Fig. 8, each point represents the normalised allocation time and money expenditure of a feasible resource allocation solution for the case study event. A circle point represents a normal solution or dominated solution in the design objective space, for which there always exists a point in the solution space that is better than this circle point in both resource allocation time and money expenditure. The star points represent a set of Pareto optimal solutions for the case study, which are also called Pareto frontier. Generally, the Pareto optimal solutions are those solutions that any improvement in one objective will result in the worsening of at least one other objective. The five triangle points (i.e., S1, S2, S3, S4, and S5) represent our solutions for the event in the five severity levels. Apparently, the all five solutions generated by the proposed approach are overlap with the Pareto frontier, which indicates the proposed approach can achieve the optimal solution for the event in each severity level in terms of allocation time and money expenditure.

### 7 RELATED WORK

Generally, most state-of-the-art optimal resource allocation systems or algorithms can be classified into centralised and decentralised approaches. In this section, some related work of resource allocation for emergency response is given and the difference between our approach and the related work is also analysed.

In majority of centralised approaches or systems, a master node is used to interact with a set of peripheral nodes, and it is assumed that the master node has global information to take appropriate decisions. As described in the experimental section, Gabdulkhakova et al. [10] proposed an agent-based solution for emergency medical allocation by using service-oriented architecture. In their system, a centralised control simulator was used to coordinate all resources (vehicle agents) from different emergency services and optimise the allocation result based on time. In [12], Hawe et al. introduced an agent-based emergency simulation system for a two-site major incident, in which a centralised control simulator was used to coordinate all resources (vehicle agents) from different emergency services and optimise the allocation result based on time. In [29], Widener et al. proposed a hybrid agent-based...
optimisation approach of positioning relief teams for large-scale disaster to assist local residents, in which $p$-median optimisation model is used to select the $p$ best relief team locations based on the given positions of all trapped residents and candidate shelters.

Apart from agent-based approaches, there are other promising centralised approaches have been proposed to optimise emergency resource allocation plans. In [26], Sung and Lee introduced an optimal allocation approach for emergency medical resources in mass casualty incidents. In their approach, providing the greatest good to the maximum number of victims is the primarily goal, which is modelled as an ambulance routing problem and formulated as a set partitioning problem. They applied a column generation approach to generate the optimal allocation plan for all available ambulances based on victims’ transportation times. Lam et al. [14] introduced another approach to optimise the response time of ambulance resources, in which geographical information system-based analysis was used to determine ambulances allocation plans based on ambulance call records and a centralised dispatching unit was used to deploy ambulances to emergency events. In [7], Chou et al. proposed a biological-based genetic algorithm to allocate rescue resources to large-scale disasters. In their approach, the traditional genetic algorithm was modified by including elite reserve areas, non-linear value conversion and migration mechanisms to effectively generate the optimal solution.

In the above approaches, it is clear that they possess some advantages by using centralised system control, such as resource deployment optimisation with global information, simplified maintenance and controlling all peripheral nodes with one master node. Nevertheless, we should not neglect the shortcomings of using centralised approaches in emergency management. For example, the approaches proposed by Lam et al. [7], Widener et al., Hawe et al. [12] and [29] were designed to optimise the resource allocation for large-scale emergency events with global information. However, in real-world situations, such large-scale emergency situations usually required different types of resources from different emergency departments or companies. These departments and companies might use their customised systems to manage their own resources, which significantly increase the difficulty of using a single master node to control all available resources. Contrarily, in our MAS, resources from different department and companies can be controlled by facility agents with heterogeneous resource management systems. Another critical problem of using centralised approaches in emergency response is system failure, especially in medical related emergency services. In [10], [14], [26], all medical resources such as ambulances are distributed through a centralised node, which can leads to a single point of failure both in decision making and in communication. Nevertheless, in our approach, resources in medical facilities are managed by independent facility agents, and the system failure of one facility agent will not affect the system operation of other agents.

When it comes to decentralised approaches of emergency resource allocation, although they have certain difficulties of achieving the optimal results as centralised approaches, but they are more practical to be implemented in real-world environments. In decentralised MASs, auction-based approaches are widely used to organise distributed agents. In [17], López et al. introduced a decentralised MAS to allocate ambulances for emergency medical events by using the inverse auction mechanism (also called contract-net protocol), in which an auctioneer (ambulance coordinator agent) proposes some tasks to bidders (ambulance team agents) to bid and a winner determine algorithm is used to generate the optimal allocation plan based on time. In [23], Su et al. also used a contract-net protocol to coordinate agents in disaster environments. In their approach, tasks and agents are divided into groups by considering spatial and communication constraints, and a two-layer coordination mechanism is used to dynamically re-group agents to maximise the number of accomplished tasks. Suárez et al. [24] proposed a reverse combinatorial auction system for task and resource allocation in the emergency rescue scenarios. In their system, rescue agents gather local rescue information from events’ areas and then pass the rescue tasks to a coordinator agent to generate tasks. Finally the coordinator agent returns a task list to each rescue agent to bid and the distance factor is used as the winning criterion to determine the winner bid in their winner determine algorithm. In [25], Suárez et al. further improved the working efficiency of their resource allocation system by incorporating a bid-tree formulation algorithm, which is capable of generating a binary tree that supports content-based lookup of bids. Once the bid-tree is generated, a modified A* algorithm is used to search for the winner bid through the tree.

Although auction mechanisms can provide effective solutions for resource allocation in decentralised MASs, one major concern is that the competitive relationships between agents in the above auction-based MASs might not suitable to modelling the cooperative nature of emergency management. In our MAS, multiple facility agents and their mobile agents are act cooperatively to propose solutions for resource allocation tasks and a deployment agent is used to optimise the final allocation proposal by considering the solutions of the entire set of participating agents. Another potential problem of the above auction-based approaches is that the resource allocation tasks are bid sequentially, which is not efficient. Even in [25], Suárez et al. incorporated a task re-schedule module in their system, some tasks that be placed at the later positions of the task queue can still be compromised, which is less effective comparing with our concurrent tasks processing approach.

8 Conclusion and Future Work

In this paper, an agent-based decentralised resource allocation approach was proposed to handle an emergency event in metropolitan regions. In order to efficiently search an optimal solution for an event that requires multiple resources with different characteristics, the proposed approach first creates a set of tasks based on the emergency services required by the event. Then, domain transport theory is used to find out the optimal resource mapping from available resources in an environment to the required resources of a task. The proposed approach is capable of handling multiple resource allocation tasks simultaneously and can be used for emergency events in different domains by simply adjusting the cost attributes in the cost function.
In order to increase the practicability of the proposed resource allocation approach, agent and multi-agent technologies was used for the system design and implementation due to their ability of autonomous decision making, reasoning, intelligent modelling and group cooperation. The multi-agent system was designed in a decentralised manner, in which resources were managed by multiple facility agents and mobile agents that could be distributed in different locations. A deployment agents was used to generate the final resource allocation plan for a task based on the proposals provided by facility agents. The experimental results indicate the good performance of the proposed approach in terms of resource allocation cost.

In the future, we will primarily focus on handling resource allocation for concurrent emergency events, in which the happening time of different events could overlap with each other. Although the proposed decentralised MAS in this paper can be applied to address the resource allocation of each event separately, one commonly occurred problem in decentralised MASs is resource contention requested by multiple events concurrently, which is not addressed in this paper. In our future work, we will incorporate coordination agents into our MAS, which could be dynamically generated to collect the information of resource contention events from deployment agents and conduct the optimal resource reassignment or dynamical reschedule resource assignment plans to improve the overall resource allocation results for multiple events.

Summary of Important Abbreviations:

- (1) res—Resources;
- (2) tas—Task;
- (3) eve—Event;
- (4) rap—Resource Allocation Proposal;
- (5) COR—Resource Cost Function;
- (6) COT—Task Cost Function;
- (7) OBJE—Event Objective Function;
- (8) OBJT—Task Objective Function.

REFERENCES

Jihang Zhang received the BSc and Honours degrees from the University of Wollongong, Wollongong, NSW, Australia, in 2014 and 2015, respectively. He is currently working toward the PhD degree in the School of Computing and Information Technology, University of Wollongong. His current researches mainly focus on adaptive resource allocation strategies for emergency management.

Minjie Zhang (SM’13) received the BSc and MSc degrees from Fudan University, Shanghai, China, and the PhD degree in computer science from the University of New England, Armidale, NSW, Australia. She is currently a professor in the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia. She is an active researcher and has published more than 100 papers in the past 10 years. Her current research interests include multiagent systems and agent-based modelling in complex domains. She is a senior member of the IEEE.

Fenghui Ren received the BSc degree from Xidian University, Xi’an, China, in 2003, and the MSc and PhD degrees from the University of Wollongong, Wollongong, NSW, Australia, in 2006 and 2010, respectively. He is currently an Australia Research Council Discovery Early Career Researcher Award in Australia fellow and a lecturer in the School of Computing and Information Technology, University of Wollongong. He is an active researcher and has published more than 50 research papers. His current research interests include agent-based concept modelling of complex systems, data mining and pattern discovery in complex domains, agent-based learning, smart grid systems, and self-organisations in distributed and complex systems.

Jiakun Liu received the BSc degree in mathematics from Chu KoChen Honours College, Zhejiang University, China, in 2006, and the PhD degree from Australian National University, in 2010. In 2010, he was awarded a Simons Postdoctoral Fellowship for a three-year term at Princeton University, Princeton, New Jersey. In 2013, he moved to the University of Wollongong to commence a continuing position. His main research interests include nonlinear elliptic and parabolic partial differential equations and applications in geometry, and optimal transportation.
Queries to the Author

Q1. Please provide the publication year in Refs. [1] and [2].
Q2. Please provide the page range for Ref. [23].