

## Chapter 2

# A Multiagent-Based Domain Transportation Approach for Optimal Resource Allocation in Emergency Management

Jihang Zhang, Minjie Zhang, Fenghui Ren and Jiakun Liu

**Abstract** In metropolitan regions, emergency events request urgent response within a short time limit in order to minimise the damage and the number of fatality. Most of these events require different resources that are usually distributed over a large area. How to efficiently allocate the distributed resources to an event is a challenging research issue. Traditional centralised resource allocation approaches have difficulties to find out the best resource assignment within the event's time limits by considering the dynamics of the metropolitan environment and the event itself. In this paper, a multiagent-based decentralised resource allocation approach using domain transportation theory is proposed to handle an emergency event with multiple tasks. Experimental results indicates that the proposed approach can effectively generate the optimal resource allocation plans by considering multiple factors of an emergency event.

**Keywords** Domain transportation · Resource allocation · Emergency management

---

J. Zhang (✉) · M. Zhang · F. Ren  
School of Computing and Information Technology,  
University of Wollongong, Wollongong, Australia  
e-mail: jz718@uowmail.edu.au

M. Zhang  
e-mail: minjie@uow.edu.au

F. Ren  
e-mail: fren@uow.edu.au

J. Liu  
School of Mathematics and Applied Statistics,  
University of Wollongong, Wollongong, Australia  
e-mail: jiakunl@uow.edu.au

## 2.1 Introduction

In recent decades, with the rapid increase of population in metropolitan regions, public emergency departments suffer significant pressure about how to efficiently and effectively allocate rescue resources for emergency events (i.e. vehicle accident, fire, terrorist attacks, etc.). Generally, these emergency events have the three common characteristics, including that (1) they are hard to be predicted in dynamic environments such as metropolitan regions; (2) they usually require multiple different resources with different usage costs, mobilities, availabilities, ownerships and functionalities (i.e. a vehicle accident might require ambulances, polices and fire fighters); and (3) they have strict time limits for emergency departments to response and allocate rescue resources. 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 with different functionalities and availabilities are distributed over an extensive area. Emergency event operators usually have difficulties to efficiently find out the optimal resource allocation for an emergency event due to a large number of possibilities.

In recent years, agent and multi-agent systems have been becoming promising technologies for resource allocation in emergency situations and many useful approaches have been proposed to assist human operators to make decisions [1–3, 5, 6]. Aygul et al. [3] proposed an agent-based solution for finding the optimal allocation of emergency medical resources, which was implemented based on service-oriented architecture. Since their approach requires the global information of all resources, so it might not available in most real life situations. Beatriz et al. [5] proposed a multi-agent system to allocate ambulances for emergency medical events by using the contract net protocol and a winner determination algorithm to find out the optimal ambulances allocation. However, their approach is restricted to handle ambulances, so it might not suitable for emergency events with the requirements of different resources. Fiedrich [1] proposed an multi-agent system for optimal resource allocation for large natural disasters based on High Level Architecture (HLA), which was capable of sequentially allocating appropriate resources to tasks. However, sequential resource allocation might increase the number of fatality of tasks in the later position of the task queue.

In order to overcome the above limitations, this paper proposes an agent-based decentralised resource allocation approach for an emergency event in metropolitan regions. The proposed approach first converts the resource allocation problem of a single emergency event into different resource allocation tasks. Then, different agents propose their resource allocation proposals to these tasks simultaneously. Finally, for each task, the domain transportation theory is used to combines these proposals and find out the optimal resource allocation that minimises the total allocation cost. The major contributions of the proposed approach are that (1) the proposed approach is designed to efficiently allocate distributed resources in decentralised manner, which is more applicable and practical comparing with centralised approaches; (2) the proposed approach is designed to effectively handle different types of emergence

events that require resources with different functionalities; (3) the proposed resource allocation framework can simultaneously allocate different resources to different tasks of an emergency event; and (4) the proposed resource allocation algorithm is highly adaptable and flexible, which is capable of generating optimal allocation solutions for emergency situations in different domains with different cost functions or attributes.

The rest of this paper is organised as follow. Section 2.2 gives problem description and definitions. Section 2.3 introduces the theoretical foundation of the optimal resource allocation. Section 2.4 describes the agent-based solution for the implementation of the proposed resource allocation approach. Section 2.5 demonstrates the experimental results and provides analysis. Section 2.6 gives the conclusion and outlines the future work.

## 2.2 Problem Description and Definitions

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

### 2.2.1 Definitions of Domain Knowledge

**Definition 1 (Environment)** An *environment* is represented by a city map, which is defined as an undirected graph,  $G = [\mathbb{V}, \mathbb{E}]$ , where  $\mathbb{V} = \{v_1, v_2, \dots, v_i\}$  is a set of nodes, which represent important locations in a metropolitan region, and  $\mathbb{E} = \{e_1, e_2, \dots, e_j\}$  is a set of edges, which represent the paths between the nodes.  $e_j$  is further defined as a two-tuple,  $e_j = (v_o, v_p)$ , where  $v_o, v_p \in \mathbb{V}$  are the nodes that be connected by  $e_j$ .

**Definition 2 (Resource)** A *resource* is defined as a seven-tuple,  $res = (rty, ser, fun, rlo, ava, vel, exp)$ , where  $rty \in \{facility, mobile\}$  represents the type of resources, where facility refers to unmovable 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;  $fun \in \{fire \text{ fighter}, fire \text{ truck}, police \text{ officer}, police \text{ car}, ambulance, medical \text{ personnel}\}$  represents  $res$ 's functionality;  $rlo \in \mathbb{V}$  represents  $res$ 's current location;  $ava = \{0, 1\}$  represents  $res$ 's availability, where 0 indicates unavailable and 1 indicates available;  $vel \in (0, +\infty)$  represents  $res$ 's average velocity in kilometre per hour (km/h) and  $exp$  represents  $res$ 's money expenditure in per hour when  $rty = facility$ , while  $exp$  represents  $res$ 's money expenditure in per hour, per kilometre when  $rty = mobile$ .

In the proposed approach, set  $\mathbb{REE}$  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,  $tas = (dea, ser, \mathbb{TR})$ , where  $dea$  represents the deadline for resources to be allocated to  $tas$ ;  $ser$  represents the type of emergency service that  $tas$  requires to make response and  $\mathbb{TR} = \{tr_1, tr_2, \dots, tr_e\}$  represents a set of required resources for completing  $tas$ .  $tr_e$  is further defined as a two-tuple,  $tr_e = \{rty, fun\}$ .

In the proposed approach, it is assumed that local emergency departments have the knowledge to estimated a task's deadline  $dea$  based on an event's severity.

**Definition 4 (Event)** An event is defined as a five-tuple,  $eve = (con, \mathbb{SER}, elo, sev, \mathbb{TAS})$ , where  $con \in \{\text{fire, rescue, loss of life, damage to health, security of person, security of property}\}$  represents  $eve$ 's content;  $\mathbb{SER} \subseteq \{\text{fire \& rescue, medical, police}\}$  represents a set of emergency services required by  $eve$ ;  $elo \in \mathbb{V}$  represents  $eve$ 's location;  $sev \in [1, 10]$  represents  $eve$ 's severity and  $\mathbb{TAS} = \{tas_1, tas_2, \dots, tas_k\}$  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,  $rap = (eve, \mathbb{RES})$ , where  $\mathbb{RES} \subseteq \mathbb{REE}$  represents a set of resources that be proposed for completing tasks in  $eve$ . Besides, a resource allocation proposal for a single task in the event is defined as a two-tuple,  $rap_k = (tas_k, \mathbb{RES})$ , where  $rap_k.\mathbb{RES} \subseteq rap.\mathbb{RES}$ .

In the proposed approach, the cost of resource allocation is calculated by cost functions. Usually, different emergency events might need to use different cost functions, which might involve different cost attributes. At here, a cost function is defined, by considering two significant factors in emergency resource allocation, i.e., money expenditure and time. 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's allocation is defined by Eq. 2.1:

$$COR(eve.task_k, res) = \begin{cases} res.exp, & \text{if } res.rty = \text{facility} \\ res.exp \times DIS(res.rlo, eve.elo) \times \frac{DIS(res.rlo, eve.elo)}{res.vel} \times \\ & DLINE(eve.task_k, res), & \text{if } res.rty = \text{mobile}, \end{cases} \quad (2.1)$$

where  $DIS(res.rlo, eve.elo)$  is a function that return the distance of a passable road between resource location  $res.rlo$  and event location  $eve.elo$ , which could be implemented by various path searching algorithms, such as Dijkstra's and A\* algorithms [7, 9].  $DLINE(eve.task_k, res)$  is a function that is used to determine

whether  $res$  can be allocated within  $eve.task_k$ 's deadline, which is further defined by Eq. 2.2:

$$DLINE(eve.task_k, res) = \begin{cases} 1, & \text{if } \frac{DIS(res.rlo, eve.elo)}{res.vel} \leq eve.task_k.dea \\ +\infty, & \text{if } \frac{DIS(res.rlo, eve.elo)}{res.vel} > eve.task_k.dea \end{cases} \quad (2.2)$$

Based on above terms, the cost function for allocating all required resources to a single task is defined by Eq. 2.3:

$$COT(eve.task_k, rap_k) = \sum_{res_v \in rap_k.RES} COR(eve.task_k, res_v) \quad (2.3)$$

Furthermore, the cost function for allocating all required resources to all tasks in a single event is defined by Eq. 2.4:

$$COE(eve, rap) = \sum_{eve.task_k \in eve.TAS} COT(eve.task_k, rap_k) \quad (2.4)$$

### 2.2.2 Problem Description

For an emergency event  $eve$ , there could be different resource allocation proposals. In the proposed approach, the all possible resource allocation proposals for  $eve$  are represented as set  $RAP$ . The main objective of the proposed approach is to search an optimal resource allocation proposal  $rap^* \in RAP$  for  $eve$ . The objective function for an event's resource allocation is formally defined by Eq. 2.5:

$$OBJE = \arg \min_{rap^* \in RAP} COE(eve, rap^*) \text{ subject to } rap^* \in REE \quad (2.5)$$

where  $OBJE$  represents the name of objective function for an event's resource allocation and  $rap^* \in REE$  means that the proposed resources in  $rap^*.RES$  must belong to the available resources in the environment  $G$ . The event objective function indicates that the optimal resource allocation proposal  $rap^*$  must have the minimal allocation cost in  $RAP$ . Besides, the proposed approach assumes there is always enough resources in  $REE$  to be allocated for  $eve$ .

However, due to the fact that an emergency event usually require resources with different types and functionalities, searching a complete  $rap^*$  could be a complicated and time-consuming process. In order to efficiently solve this searching problem, the proposed approach creates a set of resource allocation tasks  $eve.TAS$  for  $eve$ . For each task  $task_k$  in  $eve.TAS$ , it only requires resources that provide the same type of emergency service (i.e.  $res.ser$ ). By doing so, the searching of  $rap^*$  for  $eve$  is converted to the concurrent searching of  $rap_k^* \in RAP_k$  for each task  $task_k$  in  $eve.TAS$ ,

where  $\mathbb{RAP}_k$  represents the all possible or available resource allocation proposals for  $tas_k$ . The objective function of resource allocation for a task  $tas_k$  is formally defined by Eq. 2.6:

$$OBJT = \arg \min_{rap_k^* \in \mathbb{RAP}_k} COT(eve.tas_k, rap_k^*) \text{ subject to } rap_k^*.RES \in \mathbb{REE} \quad (2.6)$$

where  $OBJT$  represents the name of the objective function for an task's resource allocation.

### 2.3 Theoretical Foundation of the Domain Transportation for the Optimal Resource Allocation

Domain transportation theory is a linear programming method to minimise the cost of relocating resources [4]. In domain transportation, the resource allocation problem of a task can be described as a resource mapping problem from the available resources in an environment to the required resources of the task, which is formally represented by Eq. 2.7:

$$rap_k : \mathbb{REE} \rightarrow tas_k.TR \quad (2.7)$$

Apparently, there are many different mapping proposals (i.e.  $\mathbb{RAP}_k$ ) from domain  $\mathbb{REE}$  to domain  $tas_k.TR$ . In this paper, domain transportation theory is used to find out  $rap_k^*$  for  $tas_k$ , which can fulfil Eq. 2.6.

More precisely, let  $\mathbb{REE}_e(y_i)$  denote the amount of functionality  $e$  resource at location  $y_i \in \mathbb{V}$ . Let  $x_k = eve.tas_k$  and  $tr_e(x_k)$  represent the required amount of functionality  $e$  resource at task  $x_k$ . The cost of transferring the resource at  $y_j$  to the task  $x_k$  is given by Eq. 2.1, i.e.  $COR(x_k, y_j)$ . An admissible allocation proposal  $rap_k$  is a mapping from  $\mathbb{V}$  to  $\mathbb{TAS}$  satisfying the balance condition that for any subset  $E \subset \mathbb{TAS}$ ,

$$\sum_{x_k \in E} tr_e(x_k) = \sum_{y_j \in rap_k^{-1}(E)} \mathbb{REE}_e(y_j), \quad (2.8)$$

where  $rap_k^{-1}$  is the inverse mapping of  $rap_k$ . As before, let  $\mathbb{RAP}_k$  denote the set of all admissible allocation proposals. The purpose of Eq. 2.6 is to find an optimal proposal  $rap_k^* \in \mathbb{RAP}_k$  such that

$$COT(x_k, rap_k^*) = \min_{rap_k \in \mathbb{RAP}_k} COT(x_k, rap_k). \quad (2.9)$$

From optimal transport theory, Eq. 2.9 can be transferred to the following linear programming

$$\max \left\{ \sum_{x_k \in \text{TAS}, y_j \in \mathbb{V}} u(x_k) \text{tr}_e(x_k) + v(y_j) \text{REE}_e(y_j) : u(x_k) + v(y_j) \leq \text{COR}(x_k, y_j) \right\}. \quad (2.10)$$

Moreover, there exists a maximiser  $(u^*, v^*)$  (unique up to a constant) achieving the above maximum.  $u^*$  is called a potential, and  $v^*$  is its dual potential. The pair  $(u^*, v^*)$  also satisfies a generalised Legendre duality associated with the cost function  $\text{COR}$ . Hence, for each  $x_k$ , there exists a unique  $y_i$  such that the equality holds in the constraint, namely  $u^*(x_k) + v^*(y_i) = \text{COR}(x_k, y_i)$ , which means the task  $x_k$  requires the resource from location  $y_i$ . Thus, one can construct a mapping  $\text{inv}_k^* : x_k \in \text{TAS} \rightarrow y_j \in \mathbb{V}$ . From optimal transport theory, by differentiating the above equation one can see that

$$\begin{aligned} \nabla u^*(x_k) &= \nabla_x \text{COR}(x_k, \text{inv}_k^*(x_k)), \\ y_i &= \text{inv}_k^*(x_k) = \nabla_x^{-1} \text{COR}(x_k, \nabla u^*(x_k)). \end{aligned} \quad (2.11)$$

In fact, one 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  $\{\text{tr}_e, \text{TAS}, \text{REE}_k, \mathbb{V}\}$ , formulate the linear programming Eq. 2.10;
2. Solve Eq. 2.10 to find out a potential  $u^*$ ;
3. Using Eq. 2.11 to construct the mapping  $\text{inv}_k^* : x_k \in \text{TAS} \rightarrow \mathbb{V}$ , which implies that task  $x_k$  requires the resource from the area  $\text{REQ} := \text{inv}_k^*(x_k) \subset \mathbb{V}$ ;
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.

## 2.4 Agent-Based Decentralised Resource Allocation

The proposed resource allocation approach is implemented based on agent and multi-agent technologies, due to agents ability of autonomous reasoning, intelligent decision making, group coordination and collaboration [8]. This section gives detail description of agents' definitions, resource allocation framework and process.

### 2.4.1 Definitions of Agents

Generally, there are four types of agents in the proposed resource allocation approach, which are *response agent*, *mobile agent*, *facility agent* and *deployment agent*. Each agent's definition is described 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, which are (1) identifying event content  $eve.con$  for a new event  $eve$ ; (2) identifying the emergency services  $eve.SER$  that is required by  $eve$  based on  $eve.con$ ; (3) identifying a set of tasks  $eve.TAS$  for  $eve$  based on  $eve.ser$ ; and (4) sending  $eve.TAS$  to a deployment agent.

**Definition 8 (Mobile Agent)** A *mobile agent* is represented by  $ma$ , which has the information of a specific mobile resource  $ma.res$ . A mobile agent has two major functionalities, which are (1) managing a mobile resource  $ma.res$ ; and (2) implementing resource allocation paths after receiving resources allocation commands.

**Definition 9 (Facility Agent)** A *facility agent* is represented by  $fa$ , which has the information of a specific facility resource  $fa.res$  and a set of mobile agents. More precisely,  $fa.MA = \{ma_1, ma_2, \dots, ma_j\}$  represents a set of mobile agents that belong to  $fa$  and  $REF = \{fa.res\} \cup \{ma_1.res, ma_2.res, \dots, ma_j.res \mid ma_j \in fa.MA\}$  represents all resources under  $fa$ 's management. A facility agent has three major functionalities, which are (1) managing a facility resource  $fa.res$ ; (2) generating resource allocation proposals for tasks based on  $fa.REF$ ; and (3) informing its mobile agents to execute resources allocation commands after receiving the commands from a deployment agent.

**Definition 10 (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, which are (1) informing an event's tasks information (i.e.  $eve.TAS$ ) to facility agents that are located in its circle communication area, represented by  $da.com$ ; (2) combining and generating the optimal resource allocation proposal  $rap_k^*$  for a task based on a set of proposals (i.e.  $RAP_k$ ) submitted by facility agents; and (3) informing relevant facility agents to execute  $rap_k^*$ .

### 2.4.2 Resource Allocation Framework and Process

The proposed resource allocation approach is implemented by a multi-agent system (MAS), which includes a task identification module, resource identification module, proposal generation module, optimal allocation module and proposal execution module. The framework of the MAS is depicted in Fig. 2.1.

As depicted in Fig. 2.1, an event's resource allocation process involves one response agent, one deployment agent, multiple mobile agents and facility agents. 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  $\mathbb{FA}$  in  $da.com$ 
8: for all  $tas_k \in eve.TAS$  do
9:    $da$  creates  $\mathbb{RAP}_k$  and  $\mathbb{FA}_k$ 
10:  for all  $fa_i \in \mathbb{FA}$  do
11:    if  $fa_i.res.ser = tas_k.ser$  then
12:       $da$  updates  $\mathbb{FA}_k = \{fa_i\} \cup \mathbb{FA}_k$ 
13:       $da$  sends  $tas_k$  to  $fa_i$ 
14:       $fa_i$  finds  $rap_k^i : fa_i.REF \rightarrow tas_k.TR$ 
15:      if  $tcu \leq PDLINE(tas_k.dea, eve.sev)$  then
16:         $fa_i$  submits  $rap_k^i$  to  $da$ 
17:         $da$  updates  $\mathbb{RED}_k = \{rap_i\} \cup \mathbb{RAP}_k$ 
18:      end if
19:    end if
20:  end for
21: end for
22: while  $|eve.TAS| > 0$  do
23:  for all  $tas_k \in eve.TAS$  do
24:    if  $tcu \geq PDLINE(tas_k.dea, eve.sev) \vee \forall fa_g \in \mathbb{FA}_k : fa_g \text{ submit } rap_k^g$  then
25:      if  $\mathbb{RAP}_k$  contains enough resources for  $tas_k$  then
26:         $da$  expend  $da.com$  by double
27:        process goes back to Line 7
28:      else
29:         $da$  finds  $rap_k^* : \mathbb{RED}_k \rightarrow tas_k.TR$ 
30:         $da$  updates  $eve.TAS = eve.TAS \setminus \{tas_k\}$ 
31:         $da$  informs agents to execute  $rap_k^*$ 
32:      end if
33:    end if
34:  end for
35: 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  $eve$  happened, 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 emergency service identification,  $ra$  needs to acquire the information of the resources required by each emergency service, which may 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  $da$ .

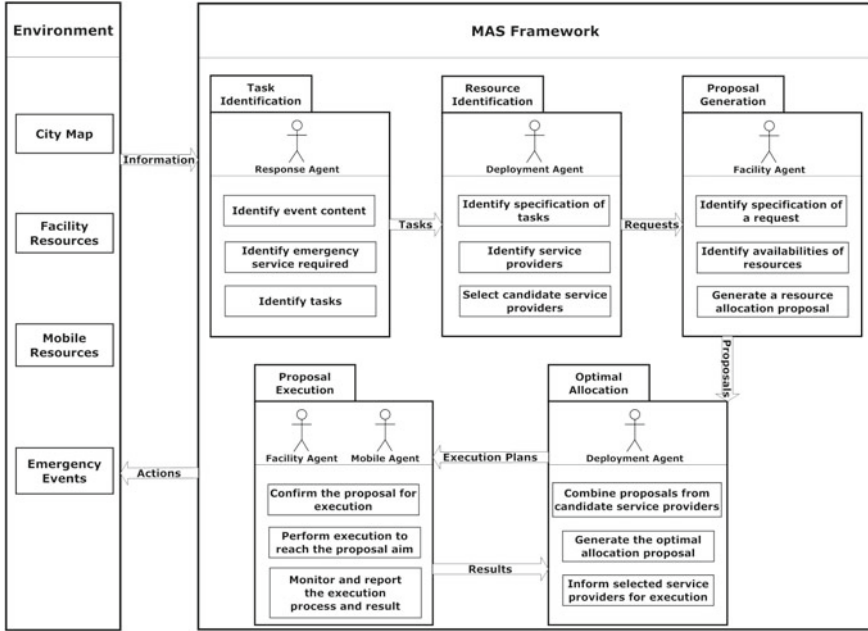


Fig. 2.1 The framework of the proposed resource allocation system

**Step 2: (Lines 6–13)** After receiving  $eve.TAS$ ,  $da$  first needs to calculate a communication area  $da.com$ , which is a circle centred at the event's location  $eve.elo$  and measured by square kilometres.  $da.com$  is calculated by Eq. 2.12:

$$da.com = \pi \times (avv \times (\frac{\sum_{task \in eve.TAS} task.dea}{|eve.TAS|} - tcu))^2 \quad (2.12)$$

where  $avv$  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 velocity of different functionalities' resources.

After the calculation of  $da.com$ ,  $da$  needs to locate all facility agents inside  $da.com$ , which is represented by set  $\mathbb{FA} = \{fa_1, fa_2, \dots, fa_i\}$  (Line 9). Then,  $da$  sends each  $task$  to relevant facility agents in  $\mathbb{FA}$  based on  $task$ 's emergency service requirement  $task.ser$ . At the same time,  $da$  also creates a facility agent contact list  $\mathbb{FA}_k = \{fa_1, fa_2, \dots, fa_g\}$  ( $\mathbb{FA}_k \subseteq \mathbb{FA}$ ) and resource allocation proposal list  $\mathbb{RAP}_k = \{rap_k^1, rap_k^2, \dots, rap_k^g\}$  for each  $task$  in  $eve.TAS$ .

**Step 3: (Lines 14–17)** After a facility agent  $fa_i \in \mathbb{FA}$  receives  $task$ ,  $fa_i$  uses the domain transportation theory (see Sect. 2.3) to calculate an optimal resource allocation proposal  $rap_k^i$  based on all available resources that under  $fa_i$ 's management (Line 14). After the calculation of  $rap_k^i$ ,  $fa_i$  submits  $rap_k^i$  to  $da$  if current time  $tcu$  has not exceeded task proposal deadline. The task proposal deadline is calculated by

function  $PDLINE(tas_k.dea, eve.sev)$ , which can be defined by local emergency departments based on the detail of  $tas_k$  and  $eve$ . After the submission of  $rap_k^i$ ,  $da$  adds  $rap_k^i$  to  $\mathbb{RAP}_k$ .

**Step 4: (Lines 23–25)** After  $da$  receives  $tas_k$ 's resource allocation proposals from all facility agents in  $\mathbb{FA}_k$  or  $tas_k$ 's proposal deadline has been reached,  $da$  checks whether  $\mathbb{RAP}_k$  has enough resources for  $da$  to generate an 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  $\mathbb{RAP}_k$  does not have enough resources to complete  $tas_k$ ,  $da$  expands its original communication area  $da.com$  by double to contact more facility agents, and then the process goes back to Step 2.

**Step 6: (Lines 29–31)** If  $\mathbb{RAP}_k$  has enough resources to complete  $tas_k$ ,  $da$  uses domain transportation theory (see Sect. 2.3) to generate an optimal resource allocation proposal  $rap_k^*$  for  $tas_k$  based on all resources in  $\mathbb{RAP}_k$ , which is represented by  $\mathbb{RED}_k$  (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.

## 2.5 Experiment

In this section, experimental results are presented and the performance of the proposed resource allocation approach is analysed. The experiments focus primarily on testing the resource allocation time, money expenditure and cost of an event when employing the proposed optimal resource mapping algorithm. The benchmark of the experiments is the decentralised ambulances allocation approach proposed by Beatriz et al. [5], which uses an auction mechanism based on trust and time to assign ambulances to emergency events.

### 2.5.1 Experimental Setting

In the experiments, the proposed resource allocation was tested in two different scenarios, which are: (1) resource allocation for an event with the proposed optimal resource allocation approach and (2) resource allocation for an event with Beatriz's ambulances allocation approach.

Although Beatriz's approach was specifically designed to allocate ambulances, but it is suitable to be used to allocate different kinds of mobile resources. Therefore, only mobile resources were required by the two scenarios' experiments. Furthermore, both two scenarios were conducted over an  $1000 \times 1000$  grid. The experiments in each scenarios were repeated for 1000 times and the average resource allocation cost, time

**Table 2.1** Parameters for resource’s setting

<i>rty</i>	<i>rlo</i>	<i>ava</i>	<i>vel</i>	<i>exp</i>	<i>fun</i>
Mobile	[0, 0]–[1000, 1000]	{0,1}	[20–200]	[1–100]	Ambulance, fire truck, police car

**Table 2.2** Parameters for event’s setting

<i>con</i>	SER	<i>elo</i>	<i>sev</i>	TAS
Fire	Fire & rescue, medical, police	[500, 500]	5	3

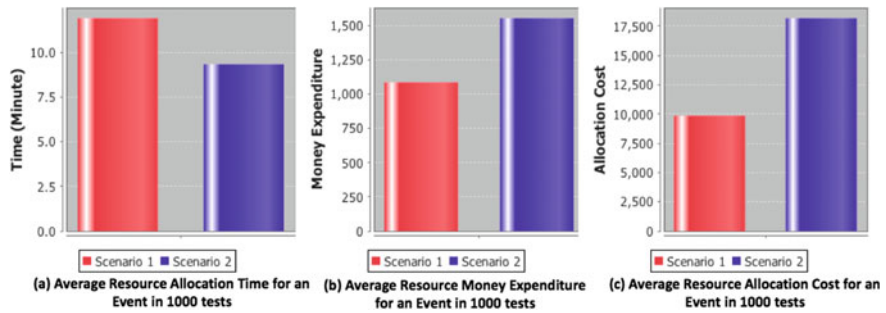
**Table 2.3** Parameters for task’s setting

No.	<i>ser</i>	<i>dea</i> (min)	TR	<i>tr.rty</i>	<i>tr.fun</i>
Task 1	Fire & rescue	30	15	Mobile	Fire truck
Task 2	Medical	40	10	Mobile	Ambulance
Task 3	Police	50	10	Mobile	Police car

and money expenditure of an event was recorded. The resource allocation cost was calculated by Eq. 2.1. For each time of the experiment, the resources’ parameters were randomly regenerated based on Table 2.1, but the parameters of the event (Table 2.2) and its tasks (Table 2.3) remain same.

2.5.2 Experimental Result and Analysis

The experimental results are shown in Fig. 2.2. As we can see from Fig. 2.2a, Beatriz’s approach (Scenario 2) requires slight less resource allocation time for an event comparing with our approach (Scenario 1). Nevertheless, from Fig. 2.2b, c, we can see



**Fig. 2.2** Experimental results

that our approach outperforms Beatriz's approach significantly in terms of money expenditure (1086 vs. 1329) and allocation cost (9850 vs. 18100). This is mainly because in Beatriz's approach, resource allocation time is used as the primary criterion to determine the optimal allocation of resources, but our approach takes both resource allocation time and resource money expenditure into consideration. Another potential reason behind the results is that Beatriz's resource allocation approach was based on auction mechanism and the relationships between contract agents are competitive. 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. By doing so, our approach can effectively minimise the required resource money expenditure and allocation cost by not compromising too much in terms of resource allocation time.

## 2.6 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 resource allocation proposal for an event that requires different resources, 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 search the optimal resource allocation proposal for each task. The proposed approach was designed to handle multiple resource allocation tasks simultaneously and it can be used for emergency events in different domains. The experimental results indicate the good performance of the proposed approach in terms of resource allocation cost. Our future work will focus on handling concurrent emergency events and incorporating a resource coordination mechanism for resource contention problems between events.

## References

1. Fiedrich, F.: An hla-based multiagent system for optimized resource allocation after strong earthquakes. In: *Proceedings of the 38th Conference on Winter Simulation*, pp. 486–492 (2006)
2. Fiedrich, F., Burghardt, P.: Agent-based systems for disaster management. *Commun. ACM* **50**(3), 41–42 (2007)
3. Gabdulkhakova, A., Konig-Ries, B., Rizvanov, D.A.: An agent-based solution to the resource allocation problem in emergency situations. In: *2011 Ninth IEEE European Conference on Web Services (ECOWS)*, pp. 151–157 (2011)
4. Liu, J.: Regularity of monge-ampère equations in optimal transportation. *Bull. Austr. Math. Soc.* **83**(01), 173–176 (2011)

5. López, B., Innocenti, B., Busquets, D.: A multiagent system for coordinating ambulances for emergency medical services. *IEEE Intell. Syst.* **23**(5), 50–57 (2008)
6. Martí, R., Robles, S., Martín-Campillo, A., Cucurull, J.: Providing early resource allocation during emergencies: The mobile triage tag. *J. Netw. Comput. Appl.* **32**(6), 1167–1182 (2009)
7. Skiena, S.: Dijkstra's algorithm. In: *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, pp. 225–227 (1990)
8. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. *Knowl. Eng. Rev.* **10**(02), 115–152 (1995)
9. Zeng, W., Church, R.L.: Finding shortest paths on real road networks: the case for a\*. *Int. J. Geogr. Inf. Sci.* **23**(4), 531–543 (2009)

Multi-agent and Complex Systems

Bai, Q.; Ren, F.; Fujita, K.; Zhang, M.; Ito, T. (Eds.)

2017, VIII, 210 p. 73 illus., 43 illus. in color., Hardcover

ISBN: 978-981-10-2563-1