PAPER

Mining Emergency Event Logs to Support Resource Allocation

Huiling LI[†], *Nonmember*, Cong LIU[†a)], *Member*, Qingtian ZENG[††b)], Hua HE[††], Chongguang REN[†], Lei WANG[†], *and* Feng CHENG[†††], *Nonmembers*

SUMMARY Effective emergency resource allocation is essential to guarantee a successful emergency disposal, and it has become a research focus in the area of emergency management. Emergency event logs are accumulated in modern emergency management systems and can be analyzed to support effective resource allocation. This paper proposes a novel approach for efficient emergency resource allocation by mining emergency event logs. More specifically, an emergency event log with various attributes, e.g., emergency task name, emergency resource type (reusable and consumable ones), required resource amount, and timestamps, is first formalized. Then, a novel algorithm is presented to discover emergency response process models, represented as an extension of Petri net with resource and time elements, from emergency event logs. Next, based on the discovered emergency response process models, the minimum resource requirements for both reusable and consumable resources are obtained, and two resource allocation strategies, i.e., the Shortest Execution Time (SET) strategy and the Least Resource Consumption (LRC) strategy, are proposed to support efficient emergency resource allocation decision-making. Finally, a chlorine tank explosion emergency case study is used to demonstrate the applicability and effectiveness of the proposed resource allocation approach.

key words: *emergency resource allocation, emergency event logs, process mining, Petri nets*

1. Introduction

Emergency resource allocation is of vital importance for emergency management systems and can greatly improve the supply efficiency of emergency resources. More and more emergency event logs are collected and stored by modern emergency management systems, and these data provide valuable insights for emergency process management, e.g., response process model discovery [1]. The availability of emergency event logs enables the so-called data-driven resource allocation techniques, i.e., supporting emergency resource allocation by analyzing emergency event logs.

Process mining aims to discover a process model from event logs automatically [2]. However, existing process mining techniques cannot be applied directly to analyze emergency event logs because of the following reasons: (1) Emergency tasks heavily rely on resources. On the one hand, large quantities of emergency resources, e.g., food, and medicine are required, and the whole emergency response process may be delayed due to lack of resources. On the other hand, if too many resources are allocated, this may cause low resource utility or even a severe resource waste; and (2) An emergency response process is a real-time service, requiring timely disposal for mission success. Therefore, time performance optimization is needed.

In view of the above features, optimizing emergency resource allocation will benefit both an emergency mission success with high time performance and a high resource utility rate. In this paper, the emergency resource allocation problem is optimized from the following two perspectives: (1) reducing emergency resource demands as many as possible; and (2) shortening emergency disposal time as much as possible. To this end, we first discover ER-net that is an extension of Petri net for emergency response processes by extending the state-of-the-art process discovery algorithm, i.e., Inductive Miner [3]. Then, the minimum reusable resource requirements and consumable resource requirements of an emergency response process are calculated using the obtained ER-nets. Finally, the Shortest Execution Time (SET) strategy and the Least Resource Consumption (LRC) strategy are presented for emergency resource decision-making.

The remainder of this paper is organized as follows. Section 2 discusses related work. In Sect. 3, emergency event logs are introduced. In Sect. 4, an emergency response process model discovery approach is introduced. Section 5 discusses the minimum resource requirement of an emergency response process. In Sect. 6, two allocation strategies are proposed to support the emergency resource decision-making. Section 7 introduces our tool support. Section 8 shows the applicability of the proposed approach by an emergency case study. Section 9 draws concluding remarks.

2. Related Work

In this section, we discuss related work on emergency response process modeling and analysis, and process mining and emergency response process mining, and finally, summarize limitations of existing work.

Manuscript received February 7, 2021.
Manuscript revised May 19, 2021.
Manuscript publicized June 28, 2021.

[†]The authors are with School of Computer Science and Technology, Shandong University of Technology, Zibo, 255000, China.
[††]The authors are with College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590, China.
[†††]The author is with School of Mathematics, Southwest Jiaotong University, Sichuan, 611756, China.
a) E-mail: liucongchina@sdust.edu.cn
b) E-mail: qtzeng@163.com

DOI: 10.1587/transinf.2021EDP7029

2.1 Emergency Response Process Modeling and Analysis

Li et al. proposed a Petri net-based approach for subway station fire emergency response process analysis in [4]. A hierarchical Petri net that includes a logic net, a semantic net, and a set of case models, is introduced for modeling and verification of emergency response processes in [5]. The applicability of this approach is validated by an emergency treatment process of highways under snow/ice weather conditions. More recently, we proposed a top-down approach for model construction and correctness verification of cross-organization emergency response processes in [6]. To support efficient emergency resource management, Zeng et al. [7] proposed an approach to support emergency resource management including both intra-organization private resource management and cross-organization public resource management. In [8], Duan et al. introduced TRM_WF_nets to model cross-organization emergency response processes and analyzed the efficiency and privacy protection by a set of proposed reduction rules.

In [9], by considering multiple disaster places and multiple resource suppliers, an optimized algorithm for obtaining optimal emergency allocation scheme was established in order to solve resource collision problem for large-scale public emergency response. In [10], the problem of allocating multiple emergency service resources to protect critical transportation infrastructures was studied. Different modeling approaches, including deterministic, stochastic programming, and robust optimization, are used to model various risk preferences in decision making under uncertain service availability and accessibility. To find an optimal solution for resource deployment and dispatching, Kondaveti and Ganz introduced a decision support framework built on rapid information collection and resource tracking functionalities to find an optimal solution for resource deployment and dispatching in [11]. The equipment control structure presented in [12] enables decentralized and collective decision-making for equipment prioritization and distribution in response to disasters.

More recently, by considering time performance, Wang et al. [13] discussed a negotiation strategy and a compromised resources allocation model for emergency response. The negotiation strategy can help requesters find the resources demanded fast and the mathematical model is presented to obtain the earliest start time of emergency response on condition of continuous requirement for resources. Zhou and Reniers [14] provided an approach to detect emergency action conflicts resulting from resource-use. For conflicts caused by limited resources sharing, the queuing system modeled by a Petri-net and integrated into the model of emergency actions, is adopted to avoid them. Gao et al. [15] proposed a two-layered framework to facilitate the allocation of limited emergency resources to meet its time constraints with high efficiency to support the cross-organization emergency resource allocation issue. In [16], Zhou and Reniers established two types of RO-TCHPN models to model emergency response to an oil fire and the queuing method is introduced to deal with possible conflicts when multiple actions use the same limited resources.

2.2 Process Mining and Emergency Response Process Mining

Process mining aims at extracting useful information from event logs, which provides new means for process discovery, monitoring, and improvement in various application scenarios [17]. For example, Alpha-miner first defines four kinds of ordering relations. Then, a Petri net is derived from these task dependency relations in [2]. To support less-structured business processes, Christian and Wil proposed the fuzzy miner in [18]. To guarantee the correctness of the discovered model, Leemans et al. proposed Inductive Miner in [3].

Liu [19] presented a data mining approach to address the resource allocation problem and improve the productivity of workflow resource management. However, this approach only focuses on who executes what task in which process and do not consider the execution time and the process nature of workflows. In the emergency response process mining area, He et al. applied process mining techniques to uncover emergency rescue process of coal mine gas explosion accidents from the historical event logs in [20]. However, this approach does not support emergency processes involving multiple organizations. To this end, we propose to discover a cross-organization emergency response process model automatically from emergency event logs by considering various collaboration patterns in [1]. However, no resource allocation analysis is performed.

2.3 Summary of Existing Work

Based on the above literature review, we can see that researches into emergency resource optimization have drawn much public attention and enjoyed an accelerated flush. Most of these works suffer from the following limitations: (1) Emergency resource quantity is not investigated during the allocation. This will result in inaccurate resource allocation, which means allocating insufficient or excessive resources to certain activities; (2) Lack of time description for emergency activities, with which we can analyze the time performance of a whole emergency response process on the basis of resource allocation; and (3) Existing resource management techniques are based on manual modelling, and data-driven techniques are missing. To deal with these problems, we propose two effective resource allocation strategies by analyzing emergency event logs as well as considering quantity and classification of emergency resources.

3. Emergency Event Logs

Emergency event logs captured during the execution of emergency response processes in emergency management systems are essentially a collection of events such that each event belongs to an emergence response process instances

Table 1 A fragment of emergency event log (t_1: investigate injuries; t_2: investigate leakage situation; t_3: medical personnel team to scene; t_4: dispose leaked chlorine; t_5: treat slightly injured people; t_6: treat severely injured people; t_7: evaluate chlorine disposal; and t_8: process post-treatment; r_1: investigators; r_2: investigation equipment; r_3: sodium bicarbonate; r_4: medical oxygen; and r_5: emergency personnel).

$\#_{case}$	*Event*	$\#_{act}$	$\#_{stime}$	$\#_{etime}$	$\#_{num}(r_1,e)/\#_{type}(r_1)$	$\#_{num}(r_2,e)/\#_{type}(r_2)$	$\#_{num}(r_3,e)/\#_{type}(r_3)$	$\#_{num}(r_4,e)/\#_{type}(r_4)$	$\#_{num}(r_5,e)/\#_{type}(r_5)$
1	e_1	t_1	7:04 Feb 01 2018	7:10 Feb 01 2018	8 / 0	4 / 0	0 / 1	0 / 1	0 / 0
1	e_2	t_2	7:11 Feb 01 2018	7:15 Feb 01 2018	10 / 0	6 / 0	0 / 1	0 / 1	2 / 0
1	e_3	t_3	7:16 Feb 01 2018	7:25 Feb 01 2018	0 / 0	0 / 0	29 / 1	10 / 1	0 / 0
1	e_4	t_5	7:26 Feb 01 2018	7:30 Feb 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
1	e_5	t_6	7:31 Feb 01 2018	7:40 Feb 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
1	e_6	t_4	7:41 Feb 01 2018	7:50 Feb 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
1	e_7	t_8	7:51 Feb 01 2018	7:55 Feb 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
1	e_8	t_7	7:56 Feb 01 2018	8:05 Feb 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
2	e_9	t_2	8:08 Nov 01 2018	8:16 Nov 01 2018	12 / 0	7 / 0	0 / 1	0 / 1	3 / 0
2	e_{10}	t_4	8:17 Nov 01 2018	8:27 Nov 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
2	e_{11}	t_1	8:28 Nov 01 2018	8:34 Nov 01 2018	6 / 0	3 / 0	0 / 1	0 / 1	0 / 0
2	e_{12}	t_3	8:35 Nov 01 2018	8:47 Nov 01 2018	0 / 0	0 / 0	22 / 1	11 / 1	0 / 0
2	e_{13}	t_5	8:48 Nov 01 2018	8:53 Nov 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
2	e_{14}	t_6	8:54 Nov 01 2018	9:02 Nov 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
2	e_{15}	t_8	9:03 Nov 01 2018	9:09 Nov 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0
2	e_{16}	t_7	9:10 Nov 01 2018	9:18 Nov 01 2018	0 / 0	0 / 0	0 / 1	0 / 1	0 / 0

and refers to an emergency activity that involves a group of attributes, such as name, timestamp (the start time and the end time), resources, etc. Formal definition of emergency events and attributes are given as follows.

Definition 1: (**Events and Attributes**) Let ξ be the event universe, i.e., the set of all possible event identifiers, $\mathcal{R}$ be the set of all possible emergency resources, and $\mathcal{A}$ be the attribute universes, i.e., the set of all possible attributes.

- For any emergency resource $r \in \mathcal{R}$, we have $\#_{type}(r)$ represents the property of emergency resource r. Two types of emergency resources, i.e., reusable and consumable ones, are involved.
- For any event $e \in \xi$, we have $\#_{case}(e)$ represents the case to which e belongs to; $\#_{act}(e)$ represents the emergency activity name of e; $\#_{stime}(e)$ represents the start time of e; $\#_{etime}(e)$ represents the end time of e; $\#_{res}(e)$ represents the set of emergency resources required by e; for any $r \in \#_{res}(e)$, $\#_{num}(r, e)$ represents the number of emergency resource r used by e.

Definition 2: (**Case and Event Log**) A case over ξ is a finite sequence of events $\sigma \in \xi^*$ such that each event appears only once and all events have the same case id, i.e., $1 \leq i < j \leq |\sigma|: \sigma(i) \neq \sigma(j) \wedge \#_{case}(\sigma(i)) = \#_{case}(\sigma(j))$. An event log is a finite set of cases, i.e., $L \subseteq \xi^*$.

Considering a chlorine tank leakage incident and its disposal process, emergency event logs are recorded and stored by emergency management systems. In general, the collected simplified chlorine tank explosion disposal event log[†], contains 264 cases, 4752 events, and 8 activities in total. This log will be used in the next sections to explain our concepts and techniques. A fragment of log is shown in Table 1 where 16 events are involved.

For example, for e_1 we have: $\#_{act}(e_1) = t_1$ means that the activity is denoted as t_1, $\#_{case}(e_1) = 1$ means that the event belongs to case with ID 1, $\#_{stime}(e_1)$ =*7:04 Feb 01 2018* is the start time of e_1, $\#_{etime}(e_1)$ = *7:10 Feb 01 2018* is the end time of e_1, $\#_{num}(r_1, e_1) = 8$ means that 8 units of emergency resources r_1 is required during the execution of e_1, $\#_{type}(r_1) = 0$ means that emergency resource r_1 is a kind of reusable resource, and $\#_{type}(r_3) = 1$ means that emergency resource r_3 is a kind of consumable resource. Note that actual resource consumption information and execution time differ case by case since the real-life disposal of the same kind of emergency varies a lot. In addition, reusable resources can be reused by other emergency activities when released while the consumable ones can be used only once and cannot be reused any more.

4. Emergency Resource Process Model Discovery

This section details the emergency response process model discovery approach from emergency event logs.

4.1 ER-Net

This section formalizes emergency response processes with resource and time attributes. Petri nets are widely used to model service-oriented processes [21], [22], software processes [23]–[25], [35], business processes [26], [27], [33], [34], [36], emergency response processes [6], [7], [28]. Basic concepts and notations on Petri nets are reviewed following [29].

Definition 3: (**Petri nets**) A Petri net is defined as a 4-tuple $\Sigma = (P, T, F, M_0)$ such that:

- $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$ where P is a finite set of places and T is a finite set of transitions;

[†]https://pan.baidu.com/s/1AKgDf17FhTYupLyVpSBBDQ, Access Number: mt8m

- $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs; and
- $M_0 : P \to \{0, 1, 2, 3 \ldots, n\}$ is the initial marking.

For any $x \in P \cup T$, the set ${}^\bullet x = \{y | (y, x) \in F\}$ is the pre-set of x, and $x^\bullet = \{y | (x, y) \in F\}$ is the post-set of x. M_0 denotes the initial marking and $R(M_0)$ is the set of reachable markings of Σ. For any $t \in T$, t is *enabled* under M, denoted as $(\Sigma, M)[t >$, if $\forall p \in^\bullet t : M(p) \geq 1$. If $(\Sigma, M)[t >$ holds, t may fire, resulting in a new marking M', denoted as $(\Sigma, M)[t > (\Sigma, M')$ such that $M'(p) = M(p) - 1$ if $p \in^\bullet t \backslash t^\bullet$, $M'(p) = M(p) + 1$ if $p \in t^\bullet \backslash^\bullet t$, and otherwise $M'(p) = M(p)$.

To model emergency response process with resource and time factors, E-net is introduced in [28]. The definition of E-net is extended to describe the uncertainty of emergency resources during execution of emergency response processes. Such extension of E-net is called ER-net.

Definition 4: (**E-net**) $\Sigma = (P, T, F, M_0, \chi, \gamma, \alpha, \beta)$ is an E-net if the following conditions are satisfied:

- (P, T, F, M_0) is a Petri net;
- $P = P_A \cup \{p_s, p_e\}$ where P_A is an activity place set, and p_s and p_e are the start and end places respectively;
- $\chi : P_A \to \mathbb{Z}^m$ is the resource function of activities. $\forall p \in P_A, \chi(p) = (q_1, q_2, \ldots, q_m)$, where q_i is the quantity of r_i required by activity p and $r_i \in Resource$. $Resource$ is the available resource set of an E-net;
- $\gamma : Resource \to \{0, 1\}, \gamma(r_i) = 0$ means that r_i is reusable, and $\gamma(r_i) = 1$ means that r_i is consumable;
- $\alpha : P_A \cup Resource \to \mathbb{R}$. $\forall p \in P_A \cup Resource, \alpha(p) \geq 0$ is the minimum duration to execute/prepare activity/resource p;
- $\beta : P_A \cup Resource \to \mathbb{R}$. $\forall p \in P_A \cup Resource, \beta(p) \geq 0$ is the maximum duration to execute/prepare activity/resource p, satisfying $\alpha(p) \leq \beta(p)$; and
- $\forall p \in P, M_O(p) = 1$ if ${}^\bullet p = \emptyset$, and otherwise $M_0(p) = 0$.

Let $\mathbb{Z}$ be a non-negative integer set, and $\mathbb{R}$ be a non-negative real number set. $\mathbb{Z}^n =< z_1, z_2, z_3, \ldots z_n >$ is an n-dimensional integer vector.

Definition 5: (**ER-net**) $\Sigma = (P, T, F, M_0, \gamma, \chi, \Gamma, \alpha, \beta)$ is an ER-net if the following conditions are satisfied:

- (P, T, F, M_0) is a Petri net such that $\forall p \in P, M_0(p) = 1$, if ${}^\bullet p = \emptyset$, otherwise $M_0(p) = 0$;
- $\gamma : Res \to \{0, 1\}, \gamma(r_i) = 0$ means that r_i is reusable, and $\gamma(r_i) = 1$ means that r_i is consumable, where $Res \subseteq \mathcal{R}$ is the available resource set;
- $\chi : T \to \mathbb{Z}^m$ is a function that maps a transition to its minimum resource vector such that $\forall t \in T, \chi(t) = (q_1, q_2, \ldots, q_m)$, where q_i is the minimum quantity of r_i required by t and $r_i \in Res$;
- $\Gamma : T \to \mathbb{Z}^m$ is a function that maps a transition to its maximal resource vector such that $\forall t \in T, \Gamma(t) = (q_1, q_2, \ldots, q_m)$, where q_i is the maximum quantity of r_i required by t and $r_i \in Res$;
- $\alpha : T \to \mathbb{R}$ is a time function that maps a transition to its minimal execution time such that $\forall t \in T, \alpha(t) \geq 0$;
- $\beta : T \to \mathbb{R}$ is a time function that maps a transition to its maximal execution time such that $\forall t \in T, \beta(t) \geq 0$, satisfying $\alpha(t) \leq \beta(t)$.

The firing rule of an ER-net is same as that of an E-net. Given a marking $M, \forall t \in T, t$ is enabled under M, if $\forall p \in {}^\bullet t, M(p) \geq 1$. Firing an enabed t removes a token from each of places in ${}^\bullet t$ and deposit one to each of places in $t^\bullet$. All properties, such as reachability and boundedness, can be defined similarly to those in an E-net. The main differences between ER-net and E-net include: (1) to keep the atomic property of transitions, we use transitions to represent emergency activities, and their time and resource is labeled on transitions to represent execution time and resource requirements; and (2) each emergency activity has two resource functions indicating the minimum and maximal resource requirements, respectively.

Algorithm 1. ER-net Discovery

INPUT: Emergency response process event logs (L).
OUTPUT:$\Sigma = (P, T, F, M_0, \gamma, \chi, \Gamma, \alpha, \beta)$.
1. $P \leftarrow \emptyset, F \leftarrow \emptyset, T \leftarrow \emptyset, M_0 \leftarrow \emptyset, \gamma \leftarrow \emptyset, \chi \leftarrow \emptyset$
   $\Gamma \leftarrow \emptyset, \alpha \leftarrow \emptyset, \beta \leftarrow \emptyset$; //Initialization
2. $(P_L, T, F_L) \leftarrow$Inductive Miner(L); //Mining the control flow
3. $E_{set} \leftarrow \emptyset$ //A set of events with the same task name
4. For each $t \in T$ //Mining the resource and time information
5. For each $\sigma \in L$
6. For each $e \in \sigma$
7. If $\#_{act}(e) == t$
8. $E_{set} \leftarrow E_{set} \cup \{e\}$
9. End If
10. End For
11. For $\forall e_i, e_j \in E_{set}(e_i \neq e_j)$ //Mining time information
12. $\alpha(t) \leftarrow min\{|\#_{etime}(e_i) - \#_{stime}(e_i)|, |\#_{etime}(e_j) - \#_{stime}(e_j)|\}$;
13. $\beta(t) \leftarrow max\{|\#_{etime}(e_i) - \#_{stime}(e_i)|, |\#_{etime}(e_j) - \#_{stime}(e_j)|\}$;
14. For each $r_i \in Res$
15. $\chi(t).r_i \leftarrow min\{\#_{num}(r_i, e_i), \#_{num}(r_i, e_j)\}$;
16. $\Gamma(t).r_i \leftarrow max\{\#_{num}(r_i, e_i), \#_{num}(r_i, e_j)\}$;
17. $\gamma(r_i) \leftarrow \#_{type}(r_i)$;
18. End For
19. End For
20. End For
21. End For
22. For each $p \in P$
23. If ${}^\bullet p == \emptyset$ //add the initial transition and initial places
24. $P \leftarrow P \cup \{p_s\}; T \leftarrow T \cup \{t_s\}; F \leftarrow F \cup \{(p_s, t_s), (t_s, p)\}$;
25. End If
26. If $p^\bullet == \emptyset$ //add the end transition and end places
27. $P \leftarrow P \cup \{p_e\}; P \leftarrow P \cup \{p_e\}; P \leftarrow P \cup \{(p_e, t_e), (t_e, p)\}$;
28. End If
29. End for
30. $M_0 \leftarrow \{p_s\}$
31. Return $\Sigma = (P, T, F, M_0, \gamma, \chi, \Gamma, \alpha, \beta)$

4.2 ER-Net Discovery

In this section, we propose an algorithm to discover ER-net models by taking as input emergency event logs.

Algorithm 1 extends the state-of-the-art process discovery algorithm Inductive Miner [3] with resources accessing and time information. By taking the emergency event logs of the simplified chlorine tank explosion response process as input, the discovered ER-net is shown in Fig. 1.

Different from E-nets, an emergency activity is repre-

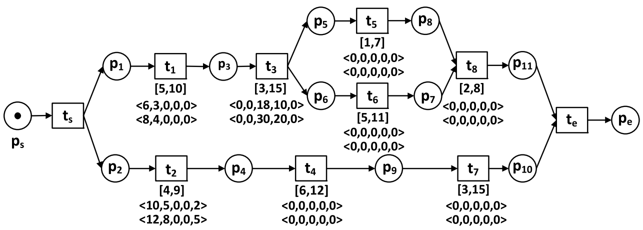

Fig. 1 ER-net of the chlorine tank explosion response process

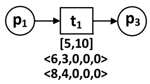

Fig. 2 ER-net of a single emergency activity

sented by a transition and two places, as shown in Fig. 2 in an ER-net. Transition t_1 represents activity t_1, and places p_1 and p_3 represent the start and end of t_1. If p_1 contains a token, it indicates that t_1 is enabled. $\alpha(t_1)$=5 and $\beta(t_1)$=10 represent its minimum and maximum execution time, and $\chi(t_1) =< 6,4,0,0,0 >$, and $\Gamma(t_1) =< 8,4,0,0,0 >$ represent its minimum and maximum resource vector.

5. Resource Requirement Analysis

In this section, we first detect potential resource dependency among activities. Then, the minimum resource requirements for both consumable and reusable resources of an emergency response process are obtained.

In the following, we first redefine the reusable and consumable resource vectors as follows.

Definition 6: Let $\Sigma = (P, T, F, M_0, \gamma, \chi, \Gamma, \alpha, \beta)$ be an ER-net, $\forall t \in T, \chi(t) =< q(r_1), q(r_2), \ldots, q(r_i) >$ represents the minimum resource vector and $\Gamma(t) =< q(r_1), q(r_2), \ldots, q(r_i) >$ represents the maximum resource vector of t. Then, we have

- $\chi^r(t) =< q(r_1), q(r_2), \ldots, q(r_m) >$ denotes the minimum reusable resource vector of t such that all involved resources are reusable resources;
- $\chi^c(t) =< q(r_1), q(r_2), \ldots, q(r_n) >$ is the minimum consumable resource vector of t such that all involved resources are consumable resources;
- $\Gamma^r(t) =< q(r_1), q(r_2), \ldots, q(r_m) >$ is the maximum reusable resource vector of t such that all involved resources are reusable resources;
- $\Gamma^c(t) =< q(r_1), q(r_2), \ldots, q(r_n) >$ is the maximum consumable resource vector of t such that all involved resources are consumable resources.

We redefine two vector operators, "<" and "≥", used in the following discussion. Let $X =< x_1, x_2, \ldots, x_n >$ and $Y =< y_1, y_2, \ldots, y_n >$ be two n-dimension vectors. If $\forall X.x_i \geq Y.y_i$, for $1 \leq i \leq n$, we denote $X \geq Y$. If $\exists X.x_i < Y.y_i$, for $1 \leq i \leq n$, we denote $X < Y$. In the following discussions, we use X_{AC} and X_{AR} to represent the available consumable resource vector and available reusable resource vector, respectively.

5.1 Resource Dependency Detection

This section presents an approach to detect resource dependency among emergency activities.

Definition 7: (**Resource Dependency**) Let $\Sigma = (P, T, F, M_0, \gamma, \chi, \Gamma, \alpha, \beta)$ be an ER-net. For any two activities $t_i, t_j \in T(t_i \neq t_j)$, t_i and t_j have resource dependency, denoted as $t_i \otimes t_j$, if $\chi^r(t_i) \bullet \chi^r(t_j) \neq \mathbf{0}$.

In Definition 7, "•" represents vector multiplication, "**0**" represents zero vector, and $\chi^r(t_i)$ and $\chi^r(t_j)$ represent the minimum reusable resource vector of t_i and t_j respectively.

According to Definition 7, Algorithm 2 is given to detect resource dependency in an ER-net.

Algorithm 2. Resource Dependency Detecting (DS)

INPUT: $\Sigma = (P, T, F, M_0, \gamma, \chi, \Gamma, \alpha, \beta)$
OUTPUT: $DS = \{DSub | \forall t_i, t_j \in Dsub \mid t_i \otimes t_j\}$
1. $DS \leftarrow \emptyset$; //Initialization
2. For $\forall t_i \in \{T \backslash DS\}$
   // to detect resource dependency among activities
3. $DSub \leftarrow \{t_i\}$ // A set of activities that use same resource
4. For $\forall t_j \in \{T \backslash DS \backslash DSub\}$
5. If $\chi^r(t_i) \bullet \chi^r(t_j) \neq \mathbf{0}$
6. $DSub \leftarrow DSub \cup \{t_j\}$
7. End If
8 End For
9. $DS \leftarrow DS \cup DSub$
10. End For
11.Output DS

5.2 Resource Requirement Analysis

In this section, we analyze the minimum consumable resource requirement, the minimum reusable resource requirement, and the reliable reusable resource requirement.

- **Minimum Consumable Resource Requirement**: The minimum consumable resource vector for an ER-net is denoted as $MX_{MC} =< q(r_1), q(r_2), \ldots, q(r_n) >$. If $X_{AC} < MX_{MC}$, then the whole process will break down because of shortage of consumable resources.
- **Minimum Reusable Resource Requirement**: The minimum reusable resource vector for an ER-net is denoted as $MX_{MR} =< q(r_1), q(r_2), \ldots, q(r_m) >$. If $X_{AR} < MX_{MR}$, then the whole process will break down because of shortage of reusable resources.
- **Reliable Reusable Resource Requirement**: Even though $X_{AR} \geq MX_{MR}$, resource conflicts may still exist because resource dependency and limited available resources. This kind of conflicts may delay the execution time of an emergency response process. Therefore, the reliable reusable resource vector $X_{RR} =< q(r_1), q(r_2), \ldots, q(r_m) >$ is introduced. If $X_{AR} \geq X_{RR}$, then there will no resource conflict during process execution, i.e., potential resource conflicts will be avoided

Algorithm 3. Calculate the resource vector

```
INPUT: Σ = (P, T, F, M_0, γ, χ, Γ, α, β), DS
OUTPUT: MX_MC, MX_MR and X_RR
sum ← 0, MX_MC ← 0; MX_MR ← 0; SUM ← 0
For ∀MX_MC.r_j ∈ MX_MC, ∀MX_MR.r_i ∈ MX_MR
      // obtain the minimum consumable and reusable resource vector
   For ∀t_i ∈ T
      sum ← sum + q(Γ^c(t_i).r_j)
      If q(Γ^r(t_i).r_i) > q(MX_MR.r_i)
         q(MX_MR.r_i) ← q(Γ^r(t_i).r_i)
      End If
   End For
      q(MX_MC.r_j) ← sum; sum ← 0;
End For
X_RR ← MX_MR
For ∀X_RR.r_n ∈ X_RR
      // obtain the reliable reusable resource vector
   For ∀DSub ∈ DS
      For ∀t_i ∈ DSub
         SUM ← Γ^r(t_i).r_i + SUM
         If SUM > q(MX_MR.r_i)
            q(X_RR.r_i) ← SUM
         End If
         SUM ← 0
      End For
   End For
End For
Output MX_MC, MX_MR and X_RR
```

because sufficient resources are provided to support parallel activities with resource dependency.

Algorithm 3 is proposed to calculate the MX_{MR}, MX_{MC} and X_{RR}. Taking the emergency event logs of the simplified chlorine tank explosion response process as an example, we can obtain $MX_{MC} =< 30, 20 >, MX_{MR} =< 12, 8, 5 >$ and $X_{RR} =< 20, 12, 5 >$ by executing Algorithm 3.

6. Two Effective Resource Allocation Strategies

In this section, we propose two optimized resource allocation strategies, i.e., the Shortest Execution Time (SET) strategy and the Least Resource Consumption (LRC) strategy. The former optimizes time performance of an emergency response process while the latter focuses on achieving a high resource utilization.

6.1 Shortest Execution Time Strategy

Without considering resource conflicts, shortest execution time of an emergency response process is analyzed. We denote the earliest time to start activity t as $T_{e1}(t)$ when activities are completed in minimum execution time, and we denote the earliest time to start activity t as $T_{e2}(t)$ when activities are completed in maximum execution time.

$$T_{e1}(t) = \begin{cases} 0 & t = t_s \\ max\{T_{e1}(t') + \alpha(t') \mid t' \in {}^\bullet({}^\bullet t) & otherwise \end{cases}$$

$$T_{e2}(t) = \begin{cases} 0 & t = t_s \\ max\{T_{e2}(t') + \beta(t') \mid t' \in {}^\bullet({}^\bullet t) & otherwise \end{cases}$$

$T_{E1} = T_{e1}(t_e)$ is the shortest execution time of an emergency response process where t_e is the sink transition when activities are executed in their minimum execution time. $T_{E2} = T_{e2}(t_e)$ is the shortest execution time of an emergency response process when activities are completed in their maximum execution time. Each event has a minimum execution time (α) and a maximum execution time (β). $T_{e1}(t)$ and $T_{e2}(t)$ are obtained by using the minimum execution time (α) and the maximum execution time(β).

Table 2 $T_{e1}(t)$ and $T_{e2}(t)$ for each activity of Fig. 1.

Activity	t_s	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_e
$T_{e1}(t)$	0	0	0	5	4	8	8	10	13	15
$T_{e2}(t)$	0	0	0	10	9	25	25	21	36	44

To maintain an emergency response process finish in the minimum execution time interval, the optimized consumable and reusable resource vectors, X_{AC} and X_{AR}, are assigned as: $X_{AC} = MX_{MC}$ and $X_{AR} = X_{RR}$. Under this resource allocation condition, consumable resources can meet the minimum requirement and reusable resources are sufficient to avoid potential resource conflicts. Therefore, the whole process can be finished in the minimum execution time interval. Take the chlorine tank explosion response process as an example, we have $MX_{MC} =< 30, 20 >$ and $X_{RR} =< 20, 12, 5 >$. Then, based on the SET strategy, we set $X_{AC} =< 30, 20 >$ and $X_{AR} =< 20, 12, 5 >$ to achieve the best time performance. The execution time interval for this emergency response process is [15, 44] as shown in Table 2.

6.2 Least Resource Consumable Strategy

To finish an emergency response process with the smallest resource consumption, the optimized consumable and reusable resource vectors, X_{AC} and X_{AR}, are assigned as: $X_{AC} = MX_{MC}$ and $X_{AR} = MX_{MR}$. Under this resource allocation condition, both consumable and reusable resources can meet the minimum requirement. Therefore, the whole process can be finished with the smallest resource consumption. However, resource conflicts may exist during the process execution. Thus, some activities in conflict may be postponed because of waiting for reusable resources which are exclusively occupied by others. As a result, time performance of the emergency response process is affected. Next, Algorithm 4 is proposed to calculate the maximum execution time interval for processes with potential resource conflicts.

Similarly, for the simplified chlorine tank explosion response process case, we can obtain $MX_{MR} =< 12, 8, 5 >$. Then, we set $X_{AC} =< 30, 20 >$ and $X_{AR} =< 12, 8, 5 >$ to pursue the smallest resource consumption. By executing Algorithm 2, we find t_1 and t_2 are in resource dependency. According to the ER-net model obtained in Fig. 1, and the meaning of resource as shown in Table 1, we see that t_1 requires 8 investigators and t_2 requires 12 investigators, and the total number of available investigators is 12. Thus, t_1 and t_2 may in conflicts if t_1 and t_2 execute simultaneously and corresponding execution time may be extended. By running Algorithm 4, the extended execution time interval is [19, 53] by using a minimum execution time (α) and a maximum execution time (β), respectively.

Algorithm 4. Calculate the maximum execution time

INPUT: $\Sigma = (P, T, F, M_0, \gamma, \chi, \Gamma, \alpha, \beta)$, DS and T_{E1}, T_{E2}
OUTPUT: $MT[MT\_\alpha, MT\_\beta]$
1. $MT\_\alpha \leftarrow T_{E1}; MT\_\beta \leftarrow T_{E2}$ //Initialization
2. For $\forall DSub \in DS$
3. For $\forall t_i \in DSub$
4. $account\_\alpha \leftarrow \alpha(t_i); account\_\beta \leftarrow \beta(t_i);$
5. For$\forall t_j \in \{DSub \backslash \{t_i\}\}$
6. $account\_\alpha \leftarrow account\_\alpha + \alpha(t_j); account\_\beta \leftarrow account\_\beta + \beta(t_j)$
7. $T \leftarrow T \backslash \{t_j\}$
8. End For
9. $\alpha(t_i) \leftarrow account\_\alpha; \beta(t_i) \leftarrow account\_\beta$
10. End For
11. End for
12. $MT\_\alpha \leftarrow T_{e1}\{t_e\}; MT\_\beta \leftarrow T_{e2}\{t_e\};$
13. Output $MT[MT\_\alpha, MT\_\beta]$

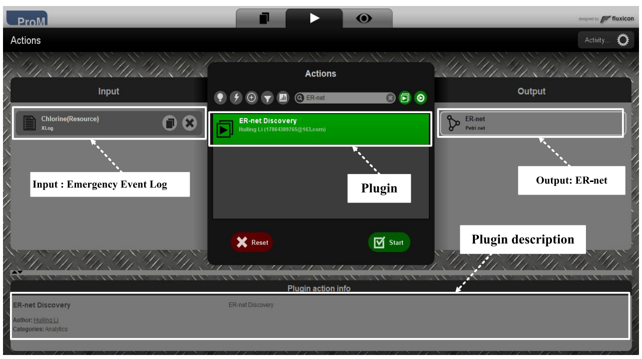

Fig. 3 Snapshot of the discovery plugin

7. Tool Support

The open-source (Pro)cess (M)ining framework *ProM 6* has been developed as a plugable environment for process event log anlaysis. The framework can be downloaded freely†.

The proposed emergency response process model discovery approach have been implemented as a plug-in, called *ER-net Discovery*, in our *ProM 6* package††. It takes an *XES*-based emergency event log as input, and returns an emergency response process model represented by an ER-net. A snapshot of the tool is shown in Fig. 3.

8. An Emergency Case Study

To demonstrate the effectiveness of the proposed emergency resource allocation approaches, we used the emergency event log collected from a complex chlorine tank explosion response process case. In general, the collected emergency event log, denoted as Chlorine_complex†††, contains 747 cases, 19422 events, and 12 emergency activities in total. In addition, each event involves a group of attributes, such as emergency activity, timestamp (the start time and the end time), required resources, etc. A fragment of this log with two cases is depicted in Table 3.

†http://promtools.org/
††https://svn.win.tue.nl/repos/prom/Packages/ShandongPM/
†††https://pan.baidu.com/s/1AKgDf17FhTYupLyVpSBBDQ, Access Number: mt8m

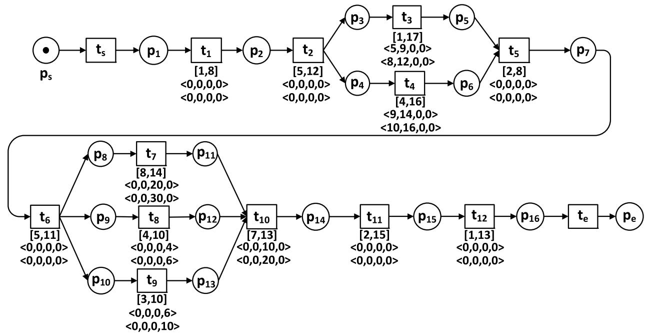

Fig. 4 ER-net of complete chlorine tank explosion response process

By taking as input the Chlorine_complex log, Algorithm 1 returns an ER-net model to represent the behavior of the complex chlorine tank explosion response process, as shown in Fig. 4. According to Fig. 4, each activity is labelled with three functions that indicating the minimum and maximum execution time, the minimum resource requirement and the maximal resource requirement. Then, we calculate the shortest time of each activity on the basis of the obtained ER-net. Time information for each activity is shown in Table 4, based on which we can obtain the shortest execution time interval of the complex chlorine tank explosion response process is [35, 111].

Then, we detect the resource dependency sets by applying Algorithm 2. Activities t_3 and t_4, and t_8 and t_9 are in resource dependency. Next, we can calculate the minimum consumable resource vector (MX_{MC}), the minimum reusable resource vector (MX_{MR}), and the reliable reusable resource vector(X_{RR}) by running Algorithms 3 as follows: $MX_{MC} =< 50 >, MX_{MR} =< 10, 16, 10 >$ and $X_{RR} =< 18, 28, 16 >$.

Next, we show the applicability of the SET and LRC strategies. Generally speaking, the SET strategy aims to guarantee the shortest execution time of the whole response process. By applying the SET strategy, the required consumable and reusable resource vectors, X_{AC} and X_{AR}, are assigned as: $X_{AC} = MX_{MC}$ and $X_{AR} = X_{RR}$, i.e., $X_{AC} =< 50 >$ and $X_{AR} =< 18, 28, 16 >$. Under this resource allocation condition, consumable resources can meet the minimum requirement and reusable resources are sufficient to avoid potential resource conflicts. Therefore, the whole process can be finished in its shortest execution time interval. The execution time interval is [35, 111] in case activities are finished in the maximum execution time, as shown in Table 4.

Different from SET strategy, the LRC strategy aims to optimize the emergency resource utilization rate, i.e., to maintain the least resource consumption of the whole response process. By applying the LRC strategy, the required consumable and reusable resource vectors, X_{AC} and X_{AR}, are assigned as: $X_{AC} = MX_{MC}$ and $X_{AR} = MX_{MR}$, i.e., $X_{AC} =< 50 >$ and $X_{AR} =< 10, 16, 10 >$. Under this resource allocation condition, both consumable and reusable re-

Table 3 A fragment of emergency event log generated from a complex chlorine tank explosion response process (t_1: publish the chlorine leakage news; t_2: rescue team rush to the site; t_3: investigators investigate the injuries; t_4: investigators investigate the chlorine leakage situation; t_5: summary the investigation results t_6: make effective disposal plan; t_7: emergency personnel deal with leaked chlorine; t_8: treatment to slightly injured people; t_9: treatment to severely injured people; t_{10}: conducting hazard mitigation operations; t_{11}: emergency evaluation; t_{12}: process post-treatment; r_1: investigators; r_2: investigation equipment; r_3: slaked lime; and r_4: doctors and nurses).

$\#_{case}$	*Event*	$\#_{act}$	$\#_{stime}$	$\#_{etime}$	$\#_{num}(r_1, e)/$ $\#_{type}(r_1)$	$\#_{num}(r_2, e)/$ $\#_{type}(r_2)$	$\#_{num}(r_3, e)/$ $\#_{type}(r_3)$	$\#_{num}(r_4, e)/$ $\#_{type}(r_4)$
1	e_1	t_1	10:23 Jan 02 2016	10:27 Jan 02 2016	0/0	0 /0	0 /1	0 /0
1	e_2	t_2	10:28 Jan 02 2016	10:38 Jan 02 2016	0 /0	0 /0	0 /1	0 /0
1	e_3	t_3	10:39 Jan 02 2016	10:46 Jan 02 2016	8 /0	9 /0	0 /1	0 /0
1	e_4	t_4	10:47 Jan 02 2016	11:02 Jan 02 2016	10 /0	15 /0	0 /1	0 /0
1	e_5	t_5	11:03 Jan 02 2016	11:08 Jan 02 2016	0 /0	0 /0	0 /1	0 /0
1	e_6	t_6	11:09 Jan 02 2016	11:18 Jan 02 2016	0 /0	0 /0	0 /1	0 /0
1	e_7	t_9	11:19 Jan 02 2016	11:25 Jan 02 2016	0 /0	0 /0	0 /1	6 /0
1	e_8	t_7	11:26 Jan 02 2016	11:36 Jan 02 2016	0 /0	0 /0	24 /1	0 /0
1	e_9	t_8	11:37 Jan 02 2016	11:44 Jan 02 2016	0 /0	0 /0	0 /1	6 /0
1	e_{10}	t_{10}	11:45 Jan 02 2016	11:57 Jan 02 2016	0 /0	0 /0	16 /1	0 /0
1	e_{11}	t_{11}	11:58 Jan 02 2016	12:08 Jan 02 2016	0 /0	0 /0	0 /1	0 /0
1	e_{12}	t_{12}	12:09 Jan 02 2016	12:14 Jan 02 2016	0 /0	0 /0	0 /1	0 /0
2	e_{13}	t_1	00:49 Jan 20 2016	00:53 Jan 20 2016	0 /0	0 /0	0 /1	0 /0
2	e_{14}	t_2	00:54 Jan 20 2016	01:04 Jan 20 2016	0 /0	0 /0	0 /1	0 /0
2	e_{15}	t_4	01:05 Jan 20 2016	01:14 Jan 20 2016	9 /0	16 /0	0 /1	0 /0
2	e_{16}	t_3	01:15 Jan 20 2016	01:25 Jan 20 2016	5 /0	9 /0	0 /1	0 /0
2	e_{17}	t_5	01:26 Jan 20 2016	01:32 Jan 20 2016	0 /0	0 /0	0 /1	0 /0
2	e_{18}	t_6	01:33 Jan 20 2016	01:39 Jan 20 2016	0 /0	0 /0	0 /1	0 /0
2	e_{19}	t_7	01:40 Jan 20 2016	01:51 Jan 20 2016	0 /0	0 /0	29 /1	0 /0
2	e_{20}	t_8	01:52 Jan 20 2016	01:58 Jan 20 2016	0 /0	0 /0	0 /1	4 /0
2	e_{21}	t_9	01:59 Jan 20 2016	02:03 Jan 20 2016	0 /0	0 /0	0 /1	8 /0
2	e_{22}	t_{10}	02:04 Jan 20 2016	02:15 Jan 20 2016	0 /0	0 /0	11 /1	0 /0
2	e_{23}	t_{11}	02:16 Jan 20 2016	02:24 Jan 20 2016	0 /0	0 /0	0 /1	0 /0
2	e_{24}	t_{12}	02:25 Jan 20 2016	02:33 Jan 20 2016	0 /0	0 /0	0 /1	0 /0

Table 4 $T_{e1}(t)$ and $T_{e2}(t)$ for each activity of Fig. 4.

Activity	t_s	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_e
$T_{e1}(t)$	0	0	1	6	6	10	12	17	17	17	25	32	34	35
$T_{e2}(t)$	0	0	8	20	20	37	45	56	56	56	70	83	98	111

sources can meet the minimum requirement. Thus, the whole process can be finished with the smallest resource consumption.

However, potential resource conflicts may exist during the process execution, and the execution of conflicting activities may be postponed because of waiting for reusable resources which are exclusively occupied by others. As a result, time performance of the emergency response process is affected. For the current case, we have detected resource dependency between t_3 and t_4, and t_8 and t_9. In addition, t_3 requires 8 investigators and 12 investigation equipment, t_4 requires 10 investigators, and 16 investigation equipments according to the obtained ER-net model. Considering the fact that the total number of available investigators and investigation equipment are 10 and 16, respectively. Therefore, t_3 and t_4 may be in conflicts if t_3 and t_4 execute simultaneously, i.e., their execution duration overlaps. Similarly, t_8 requires 6 doctors and nurses, t_9 requires 10 doctors and nurses. As the total number of available doctors and nurses are 10, t_8 and t_9 may be also in conflicts and corresponding execution time will be prolonged. By executing Algorithm 4, the real execution time interval is [36, 133] in case activities are finished in the shortest and longest time respectively.

Based on the analysis, we see that these two different strategies can be used to handle different emergency requirements, i.e., the SET strategy can be used in case emergency resources are sufficient and the emergency disposal is really urgent and should be finished as soon as possible and the LRC strategy can be selected if the emergency resource is limited and the situation is not that urgent. The emergency stakeholder can make the proper decision based on the real emergency requirement.

9. Conclusion

Decision support of resource allocation for an emergency response process is of vital importance. To some extent, this will influence emergency resource utility rate and whole emergency mission success. However, there is no formal method to deal with this problem. In our work, we first propose an emergency response process model that is represented as ER-net. Then, based on the discovered ER-net, the minimum resource requirements for both reusable and consumable resources are obtained. Next, two strategies, i.e., the Shortest Execution Time (SET) strategy and the Least Resource Consumption (LRC) strategy, are proposed to support an efficient emergency resource allocation decision-making. Finally, a chlorine tank explosion case study is used to show the applicability of our approach.

This work also opens the door for the following research: (1) emergency response disposal processes typically involves multiple organization, cross-organization process mining techniques, e.g., [1], [30], can be applied to uncover such complex emergency processes; and (2) Considering the LRC strategy, resource conflicts may exist. Effective resource conflict resolution strategies, e.g., [31], [32], should be applied to optimize the time performance of emergency response processes.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant 61902222 and 61702307, the Taishan Scholars Program of Shandong Province under Grant tsqn201909109 and ts20190936, the Humanities and Social Science Research Project of the Ministry of Education under Grants 20YJCZH159.

References

[1] C. Liu, H. Li, Q. Zeng, T. Lu, and C. Li, "Cross-organization emergency response process mining: An approach based on petri nets," Mathematical Problems in Engineering, pp.1–12, 2020.

[2] W. Van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," IEEE transactions on knowledge and data engineering, vol.16, no.9, pp.1128–1142, 2004.

[3] S.J.J. Leemans, D. Fahland, and W.M. van der Aalst, "Discovering block-structured process models from event logs - a constructive approach," International conference on applications and theory of Petri nets and concurrency, pp.311–329, Springer, 2013.

[4] Q. Li, Y. Deng, C. Liu, Q. Zeng, and Y. Lu, "Modeling and analysis of subway fire emergency response: An empirical study," Safety science, vol.84, pp.171–180, 2016.

[5] F. Lu, Q. Zeng, Y. Bao, and H. Duan, "Hierarchy modeling and formal verification of emergency treatment processes," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol.44, no.2, pp.220–234, 2013.

[6] H. Duan, C. Liu, Q. Zeng, and M. Zhou, "Refinement-based hierarchical modeling and correctness verification of cross-organization collaborative emergency response processes," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol.50, no.8, pp.2548–2559, 2020.

[7] Q. Zeng, C. Liu, H. Duan, and M. Zhou, "Resource conflict checking and resolution controller design for cross-organization emergency response processes," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol.50, no.10, pp.3685–3700, 2019.

[8] H. Duan, C. Liu, Q. Zeng, and M. Zhou, "A package reduction approach to modeling and analysis of cross-organization emergency response processes with privacy protected," IEEE Access, vol.6, pp.55573–55585, 2018.

[9] S.-S. Wang, Y. Wang, and J. Sun, "An optimized emergency resources allocation algorithm for large-scale public emergency," 2007 International Conference on Machine Learning and Cybernetics, pp.119–123, IEEE, 2007.

[10] Y. Huang and Y. Fan, "Modeling uncertainties in emergency service resource allocation," Journal of infrastructure Systems, vol.17, no.1, pp.35–41, 2011.

[11] R. Kondaveti and A. Ganz, "Decision support system for resource allocation in disaster management," 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp.3425–3428, IEEE, 2009.

[12] A.Y. Chen and F. Pena-Mora, "Decentralized approach considering spatial attributes for equipment utilization in civil engineering disaster response," Journal of computing in civil engineering, vol.25, no.6, pp.457–470, 2011.

[13] S. Wang, Y. Wang, and J. Sun, "Compromised resources allocation model for emergency response," Third International Conference on Natural Computation (ICNC 2007), pp.446–450, IEEE, 2007.

[14] J. Zhou and G. Reniers, "Petri-net based modeling and queuing analysis for resource-oriented cooperation of emergency response actions," Process Safety and Environmental Protection, vol.102, pp.567–576, 2016.

[15] Y. Gao, C. Liu, Q. Zeng, and H. Duan, "Two effective strategies to support cross-organization emergency resource allocation optimization," Mobile Information Systems, vol.2021, 2021.

[16] J. Zhou and G. Reniers, "Simulation analysis of the use of emergency resources during the emergency response to a major fire," Journal of Loss Prevention in the Process Industries, vol.44, pp.1–11, 2016.

[17] W. Van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer Publishing Company, Incorporated, 2011.

[18] C.W. Günther and W.M.P. Van Der Aalst, "Fuzzy mining – adaptive process simplification based on multi-perspective metrics," International conference on business process management, vol.4714, pp.328–343, Springer, 2007.

[19] T. Liu, Y. Cheng, and Z. Ni, "Mining event logs to support workflow resource allocation," Knowledge-Based Systems, vol.35, pp.320–331, 2012.

[20] Z. He, Q. Wu, L. Wen, and G. Fu, "A process mining approach to improve emergency rescue processes of fatal gas explosion accidents in chinese coal mines," Safety science, vol.111, pp.154–166, 2019.

[21] F. Dai, Q. Mo, Z. Qiang, B. Huang, W. Kou, and H. Yang, "A choreography analysis approach for microservice composition in cyber-physical-social systems," IEEE Access, vol.8, pp.53215–53222, 2020.

[22] J. Cheng, C. Liu, M. Zhou, Q. Zeng, and A. Ylä-Jääski, "Automatic composition of semantic web services based on fuzzy predicate Petri nets," IEEE Transactions on Automation Science and Engineering, vol.12, no.2, pp.680–689, 2014.

[23] C. Liu, "Automatic discovery of behavioral models from software execution data," IEEE Transactions on Automation Science and Engineering, vol.15, no.4, pp.1–12, 2018.

[24] T. Lu, C. Liu, H. Duan, and Q. Zeng, "Mining component-based software behavioral models using dynamic analysis," IEEE Access, vol.8, pp.68883–68894, 2020.

[25] C. Liu, "Discovery and quality evaluation of software component behavioral models," IEEE Transactions on Automation Science and Engineering, pp.1–12, 2020.

[26] C. Liu, Q. Zeng, L. Cheng, H. Duan, M. Zhou, and J. Cheng, "Privacy-preserving behavioral correctness verification of cross-organizational workflow with task synchronization patterns," IEEE Transactions on Automation Science and Engineering, vol.18, no.3, pp.1037–1048, 2021.

[27] C. Liu, H. Duan, Q. Zeng, M. Zhou, F. Lu, and J. Cheng, "Towards comprehensive support for privacy preservation cross-organization business process mining," IEEE Transactions on Services Computing, vol.12, no.4, pp.639–653, 2019.

[28] C. Liu, Q. Zeng, H. Duan, M. Zhou, F. Lu, and J. Cheng, "E-net modeling and analysis of emergency response processes constrained by resources and uncertain durations," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol.45, no.1, pp.84–96, 2014.

[29] W. Reisig, Petri Nets: An Introduction, Springer Science & Business Media, 2012.

[30] Q. Zeng, S.X. Sun, H. Duan, C. Liu, and H. Wang, "Cross-organizational collaborative workflow mining from a multi-source log," Decision support systems, vol.54, no.3, pp.1280–1301, 2013.
[31] Q. Zeng, C. Liu, and H. Duan, "Resource conflict detection and removal strategy for nondeterministic emergency response processes using petri nets," Enterprise Information Systems, vol.10, no.7, pp.729–750, 2016.
[32] Q. Zeng, H. Wang, D. Xu, H. Duan, and Y. Han, "Conflict detection and resolution for workflows constrained by resources and non-determined durations," Journal of Systems and Software, vol.81, no.9, pp.1491–1504, 2008.
[33] C. Liu, Q. Zeng, L. Cheng, H. Duan, and J. Cheng, "Measuring similarity for data-aware business processes," IEEE Trans. Autom. Sci. Eng., 2021. DOI: 10.1109/TASE.2021.3049772
[34] C. Liu, "Formal modeling and discovery of multi-instance business processes: A cloud resource management case study," IEEE/CAA Journal of Automatica Sinica, online, pp.1–10, 2021.
[35] C. Liu, "Quantitative evaluation of software component behavior discovery approach," IEICE Trans. Inf. Syst., vol.E104-D, no.1, pp.117–120, 2021.
[36] C. Liu, Y. Pei, L. Cheng, Q. Zeng, and H. Duan, "Sampling business process event logs using graph-based ranking model," Concurrency and Computation: Practice and Experience, vol.33, no.5, pp.1–14, 2021.

Huiling Li received her B.S. degree from Shandong University of Technology, Zibo, China in 2019. She is current working towards her M.S. degree in Shandong University of Technology. Her research interests are in the areas of business Process mining, emergency management.

Cong Liu received the B.S. and M.S. degrees in computer software and theory from the Shandong University of Science and Technology, Qingdao, China, in 2013 and 2015, respectively, and the Ph.D. degree in computer science and information systems from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2019. He is currently a full professor in the School of Computer Science and Technology in Shandong University of Technology. He has more than 60 publications in journals and conferences like TSC, TSMC, TASE, TBD, TCC, TITS, IoTJ, FGCS, DSS, ICWS, ICPC, EuroVIS, and etc. His current research interests include business process management, process mining, Petri nets, and emergency management.

Qingtian Zeng received the BS degree and the MS degree in computer science from Shandong University of Science and Technology, Tai'an, China, in 1998 and 2001 respectively, and the PhD degree in computer software and theory from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Professor with Shandong University of Science and Technology, Qingdao, China.

Hua He received her Ph.D. degree in computer application technology from Tianjin University. She is a lecturer at Shandong University of Technology. Her research interests include Petri nets, big data processing, and performance analysis methods.

Chongguang Ren received the B.S. degree in computer science and technology from the Shandong University of Technology, Zibo, China, in 2005, the M.S. degree in software engineering from Tongji University, in 2009, and the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2013.

Lei Wang received the B.S. degree from Ludong University (LDU), Yantai, China, in 2008 and the Ph.D. degree from South China University of Technology (SCUT), Guangzhou, China, in 2013. Now he is a lecture in Shandong University of Technology. His research interests include big data processing, image registration, and multi-modal image fusion.

Feng Cheng is an Assistant Professor with the Southwest Jiaotong University, Chengdu, China. She received the B.S. and M.S. degrees in mathematics from Southwest Jiaotong University in 2006 and 2009 respectively. In 2013, she received the Ph.D. degree in management from Tongji University, Shanghai, China. Her research interests include stochastic operations and algorithms.