https://doi.org/10.1016/j.knosys.2015.09.027

Tang P., Shen G.Q.P. (2015). Decision-making Model to Generate Novel Emergency Response Plans for Improving Coordination during Large-scale Emergencies, Knowledge-based Systems, Vol 90, Pages 111–128, DOI: 10.1016/j.knosys.2015.09.027, December. (SCI, 5-Year impact factor: 3.433, Ranked 17/130 in Computer Science, Artificial Intelligence by JCR in 2015).

Decision-making Model to Generate Novel Emergency Response Plans for Improving Coordination during Large-scale Emergencies

Pan Tang [a, b, *], Geoffrey Qiping Shen [b]

a School of Public Management/ Emergency Management, Research Center of Emergency Management, Jinan University, Guangzhou, China

b Department of Building and Real Estate, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract

Developing joint emergency response plans is an effective method to coordinate multi-agency response endeavors. This study presents a novel emergency response plan structure that considers emergency command operation requirements, such as explicitly expressing the incident objective decomposition structure, formalizing decisions in a context-sensitive manner, supporting the synchronization of responding activities with variable interval, encoding complex temporal constraints, and providing temporal flexibility. A decision-making model is developed to generate these domain-specific action plans automatically based on integrating hierarchical task network (HTN) planning and scheduling technologies. This model presents several valuable contributions to existing state-based forwarding HTN planning paradigms. First, an enhanced HTN is designed to record traversed HTN exploration space for constructing of incident objective decomposition structure and decision-making contexts. Second, the model generates temporal flexible action plans that enable the handling of temporal uncertainty in the emergency response domain. A novel concurrency controlling mechanism to ensure the parallelism of response activities with variable intervals is also proposed based on the temporally enhanced planning state that represents a dynamic emergency situation. Finally, the proposed model explicitly represents the starting and ending time of all tasks in the task network to provide complete temporal flexibility. In particular, a dedicated temporal management method taking full advantage of the decomposition structure induced by the HTN planning process is proposed for propagating time constraints on the underling Simple Temporal Network (STN) incrementally. An empirical study on typhoon evacuation demonstrates that the presented model is suitable for solving real-world problems. Therefore, the decision-making model can be applied as a computational model for the development process of emergency response plans, and will be embedded as a reasoning logic in an emergency command decision support system.

Keywords Emergency command, Emergency response plan, Hierarchical task network planning, Time propagation

1. Introduction

Large-scale emergencies constantly result in disastrous consequences. The emergency response process is beyond specific organizational boundaries and involves efforts from various functional departments, such as the police and fire departments, medical corps, military, civil organizations, and multiple jurisdictions [1]. Responding organizations focus on individual efforts, and may be unaware of the response activities taken by others. The allocation of resources to certain tasks may result in these resources becoming unavailable for other tasks. Therefore, disastrous situations present complex interdependencies and conflicts among the response tasks. This case is complicated by various factors, such as high uncertainty, considerable time pressure and urgency, severe resource shortage, and multi-authority and massive personnel involvement [2]. Effective coordination is an essential element of emergency response management. The crux for coordination in emergency response is that

---

[*] Corresponding author.
E-mail address: ptanghust2010@gmail.com

precise actions and responsibilities of responding organizations cannot be pre-defined [3]. Existing literature confirms that effective coordination during large-scale emergencies requires the development of unified action plans, which handles the rapidly changing dynamics of an emergency environment, to enable a coherent response process [4, 5, 6, 7]. Emergency command involves planning, directing, and controlling operations to achieve the identified incident objectives and coordinate the response activities, as well as relies heavily on the unified action plans, called emergency response plans, to adapt to the dynamic emergency situation [8]. These plans describe the solutions to a current emergency situation to achieve a set of identified incident objectives using limited available resources. These plans also define a course of response tasks and associated constraints to guide all participating organizations. Planning for the response process will enable arrangement of the heterogeneous tasks implemented by multiple organizations, reduce repetitive work and conflicts between responding operations, and improve the efficient use of emergency resources.

The uniqueness of a disastrous situation requires that emergency response plans be designed from scratch [9]. Therefore, the process of developing emergency response plans to coordinate and control the response activities is a critical and complex decision problem because emergency situations are complex, and incident objectives interact with each other. Decision-making process involves a sequence of decision points to select the appropriate methods to achieve response tasks under a highly volatile emergency response environment [10]. Consequently, developing emergency response plans requires deliberation mechanisms. High-level reasoning and task planning are also essential and often computationally expensive [11]. Therefore, planning for an emergency response process is clearly a challenge to the decision-making capability of emergency managers. Cognitive-level decision support is required [12]. This idea is the motivation for conducting the current study that aims to design a decision-making model to generate emergency response plans, thereby reducing the decision load of emergency managers during large-scale emergencies.

Methods and computational models to provide the aforementioned decision support should be based on an understanding of the cognitive-level process involved [13], as well as account for the characteristics of emergency command operation. These plans particularly aim to provide guidance for directing and controlling response activities. Planning for emergency response necessitates a set of requirements for the development process of the plan, and poses a challenge to the decision support tools that assist emergency managers. First, the information generated during the development process should be recorded to track and describe the structure of the incident objective decomposition. Second, the dynamic emergency situations may invalidate the execution of the emergency response plans; such situations require that the plan describe conditions representing the decision contexts based on which the actual responding efforts are monitored [14]. Third, the precise starting and ending time of response activities cannot be determined when planning for response operations because of the uncertainty of the emergency response environment. Instead, these plans should be temporally flexible and support parallel actions performed by multiple geographically dispersed task forces. The deadline of the incident objectives and other complex temporal constraints between response tasks should also be handled effectively.

Hierarchical task network (HTN) [15, 16] planning is an artificial intelligence planning technique used to search for a solution to achieve an initial task network as objectives in the initial state. The planning process proceeds by continuously decomposing tasks until all compound tasks in the task net are decomposed. Compared to classical planning paradigms, HTN planning is effective and scalabe due to that it takes structured domain knowledge to guide search process [17]. HTN is recognized for providing a particularly suitable decision support technique to formulate emergency response plans because it demonstrates several advantages. First, the problem-solving mechanism of HTN planning is similar to the underlying decision logic of planning for emergency response [17]. Second, HTN planning enables encoding of domain knowledge at different levels of abstraction in the emergency response domain. Finally, HTN planning assists in a better interpretation and understanding of decision points with different situational contexts and constraints.

To date, several HTN planning paradigms have been designed and applied to the formulation of emergency response plans. The state-based forward HTN planner called simple hierarchical ordered planner 2 (SHOP2) [18] and its predecessor, SHOP [19], are applied to assist military commanders in planning evacuation actions [20], 21]. SIADEX is another state-based forward HTN planner [22, 23] that achieves efficient and expressive handling of time and is used for the assisted design of forest fire fighting plans [24, 25]. Biundo proposed a hybrid planning paradigm that integrates HTN and partial-order causal-link (POCL) planning to provide flexible support to flood crisis management [26, 27].The domain-independent HTN planning architecture called Open Planning Architecture 2 (O-Plan2) [28] has been applied in the disaster relief domain to generate a course of action [29]. Another domain independent HTN planner, called System for Integrated Planning and Execution (SIPE-2) [30], has been applied to oil spills [31] and joint military operations planning [32]. Finally, the HTN planning paradigm XePlanner provide a highly expressive representation of domain knowledge to describe response tasks and domain-specific constraints accurately, and has been used to provide support for emergency managers in developing incident action plans during flood controlling [33]. In addition, case-based reasoning (CBR) approaches were applied to give recommendations to support emergency decision makers based on knowledge from previous disaster events. Amailef proposed an ontology-supported case-based reasoning (OS-CBR) approach by integrating ontology and case based reasoning to provide solutions to emergency managers [34]. The ontology structure is applied for real-time information extraction and case representation, which supports the case-based reasoning process for generating solutions. Comparing to HTN planning paradigm, CBR relies heavily on the past response cases and cannot handle complex disastrous situations that have no experiences before.

However, the aforementioned existing HTN planning paradigms don’t take into account for all the requirements of emergency command operation. First, all available HTN planners, except for SIADEX, provide rigid representation of time arrangement of actions, and cannot generate temporal action plans where time arrangement is unknown. Second, the objective decomposition structure and execution conditions are not represented explicitly. Consequently, these plans cannot provide support for monitoring execution process to control the response activities in all existing HTN planning paradigms. Several other deficiencies exist, which hinder the application from supporting the development process of emergency response plans. For example, temporal HTN planners, such as SIDEX, O-Plan and XePlanner, adopt PC-2 or PC-CL-2 to check the temporal consistency of the task network. However, planning time increases considerably as the number of tasks scale up. The achievement of an efficient method for handling temporal constraints remains a pending task for most temporal HTN planners. This method prevents them from being applied to solve real-world problems. These gaps are the impetus of the present study to design and develop a decision-making model based on HTN planning and scheduling technologies. Therefore, the ultimate objective of this study is to embed this model as a decision logic in an emergency decision support system.

In this study, a decision-making model that aims to improve coordination in large-scale emergency response is designed and developed based on state-based forward HTN planning and scheduling technologies. The objective is to generate temporal hierarchical emergency response plans that take into account requirements of the emergency command operation. In the following paper, the challenges involved in planning for an emergency response are analyzed, and a novel structure of emergency response plans satisfying all the requirements is proposed. With the objective of supporting the emergency command operation, this new plan structure has several domain-specific characteristics, such as expressing the objective decomposition structure, recording all decision nodes and their contexts to monitor the execution process of the plan, handling interdependencies and synchronization of the response activities with variable intervals, and providing temporal flexibility to adapt to the uncertainty and dynamic nature of the response process. The proposed decision-making model presents several valuable extensions in addition to existing HTN planning paradigms to provide better decision support to the emergency command operation and to generate the emergency response plans of this novel structure. First, an enhanced HTN is designed to record the HTN exploration space, in which the objective decomposition structure and execution monitoring conditions are extracted. Second, a new concurrent controlling mechanism based on temporally enhanced planning state and controlling rules is proposed to ensure parallelism of the response activities with the variable intervals. Finally, this plan structure allows handling of extensive temporal knowledge, such as temporal causal dependencies, deadlines, temporal landmarks or synchronization schema, and represents the starting and ending times of all the tasks explicitly in the hierarchical task network to provide temporal flexibility by integrating the Simple Temporal Network (STN) [35] with the task network. STN is used extensively to encode and reason quantitative temporal constraints over variables. In particular, a dedicated STN solver taking full advantage of the task decomposition structure induced by the HTN planning process is proposed and embedded in the decision-making model to propagate generated time constraints on the underlying STN incrementally.

This paper is organized as follows. Section 2 introduces a novel structure for an emergency response plan that accounts for the emergency command operation requirements. Section 3 introduces the knowledge formalism of the decision-making model. Section 4 presents the planning algorithm. Section 5 lists an incremental temporal management method embedded in the proposed model. Section 6 presents the method for extracting the emergency response plans. Section 7 introduces an empirical study in a typhoon evacuation domain, as well as the relevant experimental results. Section 8 discusses the related work on HTN planning and temporal constraints propagation. Finally, section 9 concludes this study with a discussion of its contributions and future work.

2. Extended emergency response plan

The enduring characteristics of emergency command operation propose requirements for an action plan to guide and direct responding task forces during emergencies. These requirements should be considered when planning for emergency response to improve coordination. In traditional planning paradigms, the produced action plans are a series of actions that are unsuitable to support the emergency command [36]. This section introduces the planning process for emergency response during an emergency command operation and proposes a number of requirements of emergency response plans. Consequently, a novel emergency response plan structure that considers all the requirements is introduced, as well as guides in the design of the decision-making model based on the integration of the HTN planning and scheduling technologies.

2.1 Planning for emergency response

Emergency response management is an example of a complex and uncertain work domain [37]. The scale of an emergency is beyond the preparation of any responsible organization [38]. An important element in an effective rapid-response effort is to develop emergency response plans quickly, coordinate efforts among multiple agencies, and guide responders by allowing them to improvise plans formulated on site based on the local emergency situation and the prepared standard operation procedures. Planning is the glue that integrates emergency services and resources not only within but also across organizations [39]. An emergency response plan need to be very specific about who is going to do what, what resource will be need, where exactly will they be working, and who is the point of contact for particular tasks [40]. Determining the precise actions and responsibilities of the involved responders is the crux of the coordination problem.

Once large-scale emergencies break out and are reported to the emergency operation center (EOC), emergency managers will collect and maintain information on the current and forecast situations, as well as the resources

Tang P., Shen G.Q.P. (2015). Decision-making Model to Generate Novel Emergency Response Plans for Improving Coordination during Large-scale Emergencies, Knowledge-based Systems, Vol 90, Pages 111–128, DOI: 10.1016/j.knosys.2015.09.027, December. (SCI, 5-Year impact factor: 3.433, Ranked 17/130 in Computer Science, Artificial Intelligence by JCR in 2015).

assigned to the incident on scene. The complexity and unpredictability of the scenarios have resulted in emergency managers not being unable to develop a detailed action plan ahead of time. Achieving the identified incident objectives is also beyond the capabilities of any organization. Therefore, emergency managers develop emergency response plans by breaking down incident objectives into more specific response tasks implemented by task forces from multiple organizations with responsibilities. This process involves a sequence of decision-making points to determine the tactical direction and specific resources, reserves, and support requirements to implement the selected strategies [41]. Emergency response plans represent the intention of emergency managers, which define the work of involved responders, to cope with disastrous situations. They also need to ensure that all responding activities undertaken are defined and time conscious. The involved sequence of decisions are dependent on the on-going incidents and response efforts throughout the emergency response plan development process. Finally, emergency managers disseminate incident orders, which are precisely the planned activities in the emergency response plan, and direct responding task forces to achieve the incident objectives collaboratively. While the individual task forces are performing specific response activities based on local situations, emergency managers monitor on-site response efforts and emergency situations, and compare the planned progress with the actual process to evaluate the validity of the current plan.

2.2 Requirements of emergency response plans

Responding to large-scale emergencies is typically characterized as dynamic, highly time-dependent, and subject to considerable uncertainty. These characteristics determine the context and constraints in developing and deploying emergency response plans. These plans should also consider the enduring characteristics of the application domain to provide suitable support for the emergency command operation. Based on the analysis of planning for emergency response, the following requirements are necessary.

(1) Expressing decomposition structure of incident objectives

Based on the planning procedure for emergency response, the identified incident objectives are decomposed step by step, until all obtained tasks can be disseminated to the responding organizations for execution. Tasks generated during the planning process are either performed by one or multiple task forces from geographically dispersed responding organizations. Each task represents an essential element to achieve the incident objectives with available resources in the current situation, possibly via the collaborative execution of several sets of subtasks by more than one responding organizations. Therefore, the hierarchies are clearly the structural characteristics necessary to understand and conceptualize the emergency response plan. Related to levels in the emergency response organizational structure, emergency response plans should reflect alternative means to achieve high levels of tasks. The hierarchical task structure is suitable for expressing the means to achieve the joint intentions of emergency managers; these intentions should be achieved by multiple responding task forces in different incident locations [42]. If the task can be represented, identified, and analyzed properly, then developing coordination mechanisms to manipulate non-local tasks based on the task features is possible [43]. This point of view on emergency response plans defines the following requirement.

Requirement 1: Extended emergency response plans should define the hierarchical task structure to express the joint intentions and its decomposition structure to achieve the incident objectives.

(2) Formalizing decisions in a context-sensitive manner

Developing emergency response plans involves a sequence of decision points to select the proper methods to achieve complex tasks. These decisions are also formed based on the incident status, responding efforts, and other constraints. The decision contexts describe specific conditions under which tasks in the emergency response plans can be achieved. However, the highly dynamic environment changes constantly and unpredictably while planning for emergency response. The difficulty of developing an accurate and precise model on how the environment evolves over time may result in the developed emergency response plans not being incapable of adapting to the existing situation. The uncertainty from the execution process of the response activities also results in unexpected effects. Consequently, discrepancies between the assumed and actual emergency situations occurs frequently during the emergency response plan development and deployment processes. Therefore, emergency managers should evaluate whether the current executing plan remains valid during an emergency command operation. The design requirement based on the analysis is listed as follows.

Requirement 2: Decision contexts should be formalized explicitly by defining attached conditions and constraints in the emergency response plans to monitor plan validity during emergency response.

(3) Supporting synchronization of the planned activities with variable intervals

During emergencies, planning products should be disseminated to multiple geographically dispersed responding task forces on scene based on the chain of command. Responders undertake specific orders by implementing complex processes that must satisfy the given constraints and produce the execution effects based on the operation procedures and local emergency situation. Therefore, the specific start and end time of planned activities in the emergency response plan cannot be pre-determined during the emergency command operation. Instead, the exact execution time is determined by field responders in actual local situations.

Planned activities can be conducted simultaneously and independently if one activity does not affect the others because the response task forces are also geographically dispersed. Consequently, improving synchronization among the response activities in emergency response plans can save considerable response time, thereby resulting in better performance. By contrast, if the execution of an activity affects or is affected by another one, then interdependencies exist between these activities. In the first case, one planned activity generates the execution effects, which provides the precondition to execute another one. Therefore, a cause–effect relationship exists between these activities. In the other case, if more than one planned activity requires common resources, such that rescuing the victims under debris in different locations requires the only available search-and-rescue team and equipment, then they should be carried out one after the other depending on the scheduling rules. Based on the analysis, the requirement is detailed as follows.

Requirement 3: The extended emergency response plan should be sufficiently elaborate to describe the interdependencies and synchronization between planned activities to define coordination across multiple responding task force units.

(4) Encoding complex temporal constraints and providing temporal flexibility

Temporal constraints should be considered while developing emergency response plans. First, incident objectives with deadlines must be achieved before emergencies result in disastrous effects. Second, partial order relationships between response tasks without detailed information on durations [44] are represented and handled explicitly to synchronize and coordinate the response work of multiple responding organizations. Complex synchronization, such as deadlines or temporal landmarks of response tasks and other synchronization schemas between them, also defines temporal constraints that should be taken into account while planning for emergency response. By contrast, causal-effect dependencies between response tasks carried out by multiple responding organizations imply temporal constraints between them. Consequently, multiple types of temporal constraints underlying HTN should be encoded and handled explicitly.

Unanticipated events, such as traffic congestion that delays the arrival of fire rescue personnel, changes in weather conditions, and bad weather that prevents the necessary equipments from arriving on site, may affect the execution process to complete the response tasks. This situation demonstrates that the precise start and end time of response tasks cannot be determined during the development process of the plan. Therefore, emergency response plans should provide temporal flexibility to adapt to the uncertain and changing environment. Based on the preceding analysis, the following designing requirement is provided.

Requirement 4: The precise start and end time of the response task in emergency response plans should satisfy all defined temporal constraints, and cannot be pre-determined before execution.

2.3 Extended emergency response plan structure

Traditional action plans in existing HTN planning paradigms cannot satisfy the aforementioned design requirements. That is the impetus of the current study to extend the action plans with enriched syntax and semantics to represent the emergency response plan. In this section, the hierarchical task network generated during the planning process is recorded to express the structure of the incident objective decomposition. The conditions are attached to HTN to encode the decision contexts of multiple decision points during the emergency response plans developing process. Finally, unlike action plans with determined start and end times, STN is integrated into the action plans to represent synchronization among the planned activities and encode all the underlying time constraints to provide temporal flexibility. The extended emergency response plan structure is listed as follows.

Definition 1: The emergency response plan has the following form $emResPlan = < TemRefTaskNet, eResProcess >$, where the element $TemRefTaskNet$ is a temporal-refining task network representing the incident objective decomposition structure and underlying time constraints and the element $eResProcess$ defines emergency response business process to coordinate all responding task forces.

The detailed structure of these two elements in emergency response plan is listed as follows.

(1) Temporal refining task network

The temporal refining task network is listed as the following:

Definition 2: The temporal refining task network has the form: $TemRefTaskNet =< refTaskNet, exStn, actSet >$, where the first element is a hierarchical task network describing the incident objective decomposition structure; the second one is an extended STN encoding time constraint underlying all tasks; and the last one is a set of planned activities to achieve primitive tasks, which are exactly the leaf nodes of the hierarchical task network.

1) The variable $refTaskNet = \{exTaskNode_i =< task_i, st_i, et_i, tType_i, desConList_i, parTask_i, childTaskSet_i >\}$ ($1 \le i \le M$, where M is the total number of task nodes) consists of a set of extended task nodes. Each task node represents a task $task_i$, which is executed during the interval between its start time st_i and end time et_i. The boolean variable $tType_i$ represents the type of $task_i$. If $tType_i$ is true, it is a compounded task. Otherwise, it is a primitive one. As the task in the HTN planning paradigm, the primitive task node is executed through a planned activity directive, and the variable $childTaskSet_i$ is initialized by the unifying planned activity. Instead, each compounded task node is achieved by performing a set of child task nodes recorded in the variable $childTaskSet_i$. The variable $parTask_i$ is the parent task of task $task_i$. Finally, the variable $desConList_i = \{desCon_i^j\}$ ($1 \le j \le N$, where N is the total number of means for achieving task $task_i$) is a set of conditions or constraints that the current situation and available resources should satisfy when the current task node can be achieved by those in the variable $childTaskSet_i$.

2) The variable $exStn$ is an extended STN underlying the hierarchical task network to encode all time

constraints on the start to the end time with the task nodes. Unlike traditional action plans that determine the exact execution time or define the order between tasks, the variable provides temporal flexibility for adapting to the uncertainty of emergency response processes by encoding all time constraints underlying the emergency response plan structure. The detailed structure of the extended STN will be introduced in Section 5.

3) The variable $actSet$ is a set of planned activities describing the execution processes of associated primitive tasks. The detailed paradigm will be introduced in Section 3.1. Each planned activity achieves a primitive task in the hierarchical task network.

The role of the abovementioned structure maintains an instance of HTN exploration for the current planning problem and provides guidance for emergency managers to direct and supervise the response efforts.

(2) Emergency response business process

The emergency response business process consists of a set of planned activities with order and synchronization relationships. The planned activities achieving the primitive task nodes in the temporal refining task network describe the response actions carried out by geographically dispersed responding task forces, such as firefighting, searching for victims, transporting evacuee, setting up tents, and so on. According to the analysis, the emergency response business process is defined as the follows.

Definition 3: The emergency response business process has the form $eResProcess = \{actPlus_i =< act_i, Prev_i, Succ_i >\}$ ($1 \le i \le L$, where L is the total number of actions plus in the process). The action plus $actPlus_i$ is an extended action model for recording ordered relationships between them in the business process. In each action plus model, the variable act_i represents a planned activity, where the variables Pre_i and $Succ_i$ record a set of immediate previous action plus and a set of immediate successor ones of the action act_i.

For the immediate successor action plus set of a given action plus $actPlus_i$, if an action plus $actPlus_j$ exists in $actPlus_i.Succ_i$, then another action plus $actPlus_k$ does not exist, such that $actPlus_i.act_i \prec actPlus_k.act_k$ and $actPlus_k.act_k \prec actPlus_j.act_j$. In the same way, if an action plus $actPlus_j'$ exists in $actPlus_i.Prev_i$, then another action plus $actPlus_k'$ does not exist, such that $actPlus_k'.act_k' \prec actPlus_i.act_i$ and $actPlus_j'.act_j \prec actPlus_k'.act_k$.

The planned activity is also represented by an operator instance as shown in Section 3.1, which describes the detailed execution process of a specific primitive task node. Its start and end times are exactly the same as those of the associated primitive task node. Moreover, the assumed execution effects representing response efforts describe the way the emergency situation will change during the execution interval. The assumed execution effects also set the objectives for task forces performing this activity and provides the criteria for monitoring the execution progress for emergency managers. As a result, the task forces will carry out the responding activities according to the local situation to produce the defined effects. Finally, the sequence and parallel relationships are also defined explicitly to provide support for emergency managers in EOC to issue the incident orders.

The abovementioned emergency response plan also describes the incident objective decomposition structure generated by the emergency response plan developing process. Second, the plan structure represents decision contexts explicitly. This representation enables emergency managers to associate monitors to different abstraction levels of tasks and detect situations that may invalidate the current plan. Third, the start and end times of the planned activities are not pre-determined until execution to enhance the temporal flexibility of the entire plan. The following content introduces the design of a decision-making model that will generate this plan structure based on integrating HTN planning and scheduling technologies.

3. Knowledge formalism

This section proposes the knowledge formalism of our decision-making model based on AI planning technologies, such as the temporal enhanced operator, temporal planning state, and enhanced hierarchical task network. They provide basis for the design of a planning algorithm that will produce the abovementioned emergency response plans. The other knowledge formalisms are similar to the existing HTN planning paradigm XePlanner [33].

3.1 Temporal enhanced operator

An operator describes the responding activities performed by task forces, such as searching for victims, firefighting, transporting evacuees, and other tasks. According to the design requirements in Section 2.2, the responding activity is executed by a complex process carried out by responding organizations. Hence, for adapting to the characteristics, a temporal enhanced operator is proposed by extending operator formalism in the existing HTN planner, such as XePlanner [33] and SHOP2 [18]. The extensions are listed as the following:

(1) The instantaneous precondition of operators can be satisfied before the responding activity starts or at any time during execution. The invariant conditions in operators are also formalized explicitly and describe the conditions that should be satisfied in any interval of the execution process.

(2) The duration of the operator cannot be pre-determined. Instead, the enhanced operator provides terms to represent the start and end times of the response activities, which cannot be estimated exactly because of the uncertainty in the emergency response process.

An instance of a temporal enhanced operator is shown in Figure 1. This knowledge formalism describes an activity which drives a vehicle ?t from location ?loc-from to location ?loc-to. Unlike the operator in PDDL 2.1 [36], the duration of this operator is not pre-defined. In addition, the effects occurs at any time during the execution interval.

```
(:operator (!drive ?t ?loc-from ?loc-to)
 ;;instantaneous precondition
 ((((at ?t ?loc-from)(distance ?loc-from ?loc-to ?d)
   @ 0))
 ;;invariant conditions
 ()
 ;;delete list for instantaneous effects
 ((at ?t ?loc-from))
 ;;add list for instantaneous effects
 ()
 ;;delete list for the delayed effects
 ()
 ;;add list for the delayed effects
 (((at ?t ?loc-to) + et))
 ;;cost
 50)
```

Fig. 1. Example of an operator

3.2 Temporal enhanced planning state

In classical HTN paradigms, the planning state is represented by a set of predicates that assume that these propositions are true, while propositions not in the classical planning state are false. In this section, an enhanced planning state is proposed for recording the time when the predicates are generated or triggered. Therefore, the temporal enhanced predicate is also called the temporal predicate. The positive and negative predicates are all recorded in the enhanced planning state. If the enhanced predicate is a positive one, then the predicate p is added to the world state after being triggered. Otherwise, the predicate p is deleted from the world state. The abovementioned analysis shows that the temporal predicate has the following form.

Definition 4: A temporal predicate is a triple $< p, flag, tp >$. The variable p represents the proposition. The variable $flag$ is a boolean variable expressing whether the predicate is a positive or a negative predicate. If this variable is true, then the predicate is positive; otherwise, it is negative. The variable tp is the time stamp that records the time when the predicate is generated.

During the planning process, temporal predicates are given by the initial planning state or generated by the execution effects when the operators are applied to achieve the primitive tasks. Therefore, if a predicate is specified in the initial state, then the variable tp is initialized by TR representing the reference time of the emergency response process. Otherwise, the variable tp is specified by the formula $t \pm delta$. If the symbol "$+$" is given, then variable t represents the start time of an operator; otherwise, it represents the end time. The variable $delta$ is the time interval relative to the start or end time when the execution effects are generated. As a result, the temporal enhanced planning state consists of a set of temporal enhanced predicates.

Unlike traditional planning paradigms [16], the abovementioned planning state assumes that the properties of the world change over time and records the time when these properties change. Instead of a single "global time," multiple "local times," which are the time stamps of each predicate, are recorded in the enhanced planning state. They provide basis for discovering the cause-effect relationships between the planned activities and the design of concurrent controlling mechanism to ensure parallelism of multiple planned activities. The details will be introduced in section 4.

3.3 Enhanced hierarchical task network

During the HTN planning process, the plan-refining steps include decomposing a compounded task and applying a primitive one. The enhanced hierarchical task network is designed to record the entire decomposition structure of the incident objectives generated during the entire planning process. Its formalism is listed below.

Definition 5: The enhanced hierarchical task network has the form $enHieTaskNet = \{tplus_i\}$ $(1 \le i \le M$, where M is the total number of task pluses) and consists of a set of task pluses for recording the generated information during the search process. The task plus is an extended task model that expresses explicitly the start time, end time, type, and decomposition structure. The definition of task plus is provided in detail as below.

Task plus has the form $taskPlus_i =< task_i, st_i, et_i, tType_i, refinPreconList_i >$, where $task_i$ is a task atom similar to that in classical HTN planning paradigms, st_i and et_i represent the start and end times of the task respectively, and $tType_i$ is a boolean variable expressing the type of $task_i$. If the variable $tType_i$ is true, then $task_i$ is a compounded task. Otherwise, it is a primitive one. The variable $refinPreconList_i = \{refinPrecon_i^j\}(1 \le j \le N)$ records a set of plan-refining steps to achieve task $task_i$. These steps are generated when all preconditions of the methods unifying the current task are evaluated to determine whether they are satisfied in the current planning state.

The formalism of the plan-refining step is listed as below.

Definition 6: The plan-refining step has the form $refinPrecon_i^j = (preList_i^j, \mathrm{var}BindArrayMean_i^j)$. The

Tang P., Shen G.Q.P. (2015). Decision-making Model to Generate Novel Emergency Response Plans for Improving Coordination during Large-scale Emergencies, Knowledge-based Systems, Vol 90, Pages 111–128, DOI: 10.1016/j.knosys.2015.09.027, December. (SCI, 5-Year impact factor: 3.433, Ranked 17/130 in Computer Science, Artificial Intelligence by JCR in 2015).

variable $preList_i^j$ is a list of preconditions that record the instantaneous preconditions of methods or operators for refining the given task $task_i$. If the evaluation shows these preconditions have been satisfied in the current planning state, then variable $vaBindMeanList_i^j = \{vaBindArrayMean_k\}$ $(1 \le k \le L)$ records all alternative achieving means $vaBindArrayMean_k$ to refine the given task in this specific instantaneous precondition list, and L is the total number of alternative achieving means. Otherwise, it is initialized by null. The variable $vaBindArrayMean_k$ can achieve a given task when the instantaneous precondition list $preList_i^j$ is satisfied in the current planning state with a specific variable binding array. Its detailed formalism is listed as below.

Definition 7: The achieving means has the form $vaBindArrayMean_k = (varBindArray_k, children_k)$, where the variable $vaBindArray_k$ is an array of variable bindings specifying that the given instantaneous precondition list is satisfied in the current planning state with it. If the task plus is a compounded one, then the variable $children_k = \{taskPlus_k{}^m\}(1 \le m \le L)$ is a set of task pluses called child task pluses used to achieve the given task $task_i$ with the variable array $varBindArray_k$. Otherwise, the variable records the operator to achieve the primitive task plus with this variable array.

According to the abovementioned definition, the enhanced hierarchical task network records the traversed search track by decomposing the initial incident objectives. Obviously, the track can be used to extract the abovementioned emergency response plan. An example of the enhanced hierarchical task network is shown in Figure 2 to describe its structure. As shown, the icons in this figure represent the variables in definition 6 and 7.

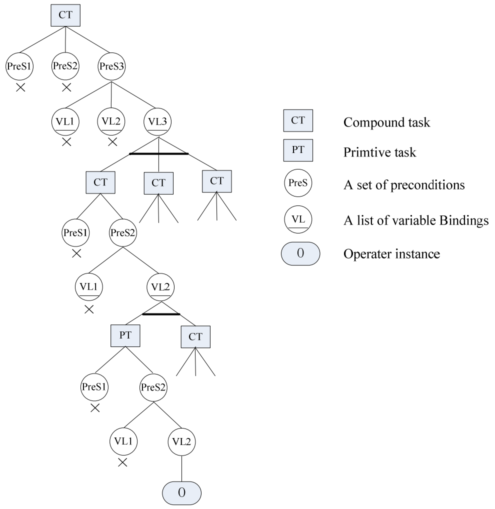

Figure 2. An example of the enhanced hierarchical task network

4. State-based forward planning process

Developing emergency response plans during large-scale emergencies involves knowledge-based high-level cognitive control [45], which challenges the decision-making capabilities of emergency managers. The planning process is a computational model for providing cognitive-level support to the development process of emergency response plan, which involves a sequence of decision points as shown in section 2.1. This section introduces the main algorithm and key procedures of the planning process.

4.1 Main algorithm

The input of the planning algorithm is the initial planning state representing the emergency situation at the beginning of the emergency response and all the identified incident objectives to be achieved. Similar to SHOP2, this planning process is a state-based forward and proceeds as the enhanced hierarchical task network is expanded or the planning state is refreshed. The planning process is also advanced to expand a common search space consisting of search nodes with a special structure. The detailed definition of a search node is listed as below.

Definition 8: The search node has the form $sn = < ActSet, s, taskPlusSet, exStn >$, where the variable $ActSet$ is a set of planned activities represented by the generated operator instances, the variable s is the current planning state, the variable $taskPlusSet$ is a set of leaf task pluses in the enhanced hierarchical task network to be refined, and the variable $exStn$ is an extended STN encoding all the time constraints underlying the associated hierarchical task network.

In the search node formalism, the start and end times of each operator instance in $ActSet$ are the start and end times of the unifying primitive task plus. All parent task pluses of the generated operator instances and task pluses in $taskPlusSet$ constitute the hierarchical task decomposition structure associated with the current search node. This structure can be extracted from the common enhanced hierarchical task network. The extended STN will be introduced in Section 5.

The search space recording all search nodes is initialized by $< \varnothing, s_0, taskPlusSet, Stn, \varnothing >$ and is exploited by advancing the planning process. A common data structure of the enhanced hierarchical task network is expanded and records all plan-refining information. The variable $enHieTaskNet$ is initialized by all task pluses recorded in $taskPlusSet$, which represent all the incident objectives to be achieved.

Similar to XePlanner, this algorithm performs a depth-first search process in the space of all decompositions of the given initial task network, returns a set of search nodes sorted by the metric values, and outputs an optimal one when the emergency managers determine to interrupt it. In each iteration, a search node is selected from search space $openList$. Thereafter, the tasks with no predecessors in this search node are selected to be refined. The plan-refining steps decompose the compounded tasks and apply the primitive tasks, as introduced in Section 4.2. When all compounded tasks in the current search node are decomposed and all primitive tasks are applied, or when the planning time exceeds the time limitation given by emergency managers, the planning process terminates, and the emergency response plans are extracted as shown in Section 6.

The key procedures in the presented model, such as expanding the search space, discovering the cause-effect relationships between tasks, and concurrent controlling mechanism of multiple planned activities, are novel and will be introduced in the following section, and are different form those in XePlanner [33].

4.2 Key procedures

In this part, the key procedures involved in the main planning algorithm are presented, and the processes of providing specific functions by the presented decision-making model are introduced.

4.2.1 Expanding the search space

Given a search node, the plan-refining steps include decomposing compounded tasks and applying primitive ones. The details are listed below.

(1) Decomposing compounded tasks. When a compounded task t with no predecessors is selected to be refined, the unifying methods are used to decompose it. First, the preconditions of each branch are evaluated individually on whether they are satisfied by the current planning state. If they are satisfied with a variable binding array, then the generated cause-effect relationships are added to the variable $exSTN.cauEftSet$, which records all cause-effect relationships between these two time points. All subtasks in this branch are added to the variable $enHieTaskNet$. Their time points and the underlying time constraints are also added to the extended STN $exSTN$.

(2) Applying primitive tasks. When a primitive task t with no predecessors is selected to be refined, it is applied by the unifying operators. First, the preconditions of this operator are evaluated as decomposing the compounded tasks. If they are satisfied by the current planning state, then the enhanced STN is updated in the same manner. The operator instance is also added to the variable $enHieTaskNet$ and the search node. As a result, the planning state is updated by the execution effects of this operator instance.

4.2.2 Discovering the cause-effect relationships between tasks

An important function of emergency response plans is to discover and manage interdependencies between response tasks and activities. One of the principles of this plan is coordinating multiple responding task forces based on unified action plans during emergency command operation. The basic interdependencies between tasks are cause-effect relationships [22] that show that the producing task provides the execution conditions of the consuming task. During the search space expanding process, two basic plan-refining steps discover and generate cause-effect relationships among the tasks, such as decomposing compound tasks and applying primitive ones. When a compounded task is decomposed, the instantaneous precondition list for each branch of the unifying methods is evaluated to check the satisfiability. For each literal in an instantaneous precondition list with time point t_i that matches a positive temporal predicate with time point t_j in the planning state, a cause-effect relationship is generated to record the causal structure between the tasks and is defined on these time points. The same procedure is applied for a primitive task.

From the abovementioned analysis, the knowledge formalism of cause-effect relationships is listed as below.

Definition 9: A cause-effect relationship has the form $ce_l =< t_i \xrightarrow{p} t_j >$, where the variable t_i is the time stamp of the last positive temporal predicate generating p, and variable t_j is the time stamp of the instantaneous precondition satisfied by the former predicate.

An instantaneous precondition $instantPre_i = (logicalExp_i @ t_i)$ in a method or an operator refining a task implies that the complex logical expression $logicalExp_i$ must be satisfied by the planning state before time t_i. The logical expression $logicalExp_i$ is a logical atom or a complex expression, such as conjunction, disjunction, negation, assignment expression, and call expression as in SHOP2 [18]. The time point t_i is either the start time or any time point during the interval of a compound task or a primitive one.

From the abovementioned definition, our decision-making model supports complex logical expressions for checking the satisfaction of preconditions, and the details of cause-effect relationship generation are listed as follows.

(1) If a logical expression $logicalExp_i$ is a logical atom and is satisfied by a temporal enhanced predicate

$< p_j, t_j >$ in the current planning state, then a cause-effect relationship $ce_l =< t_i \xrightarrow{p_j} t_j >$ is generated.

(2) If a logical expression $logicalExp_i = (and\ L_1 \cdots L_N)$ ($1 \le j \le N$) is a conjunction, where N is the total number of logical atoms in this expression. If a logical atom L_j is satisfied by a temporal enhanced predicate $< p_j, t_j >$ in the current planning state, then a set of cause-effect relationships $\{ce_j =< t_i \xrightarrow{p_j} t_j >\}$ are generated.

(3) If a logical expression $logicalExp_i = (or\ L_1 \cdots L_N)$ ($1 \le j \le N$) is a disjunction, where N is the total number of logical atoms in this expression. If there exist a logical atom L_j satisfied by a temporal enhanced predicate $< p_j, t_j >$ in the current planning state, then a cause-effect relationship $\{ce_j =< t_i \xrightarrow{p_j} t_j >\}$ is generated.

(4) If $logicalExp_i$ is a disjunction, negation, assignment expression, or call expression, then no cause-effect relationship is generated.

Obviously, unlike the causal links between two operators in POCL [26], the presented cause-effect relationships are temporal and are defined on two time points. As a result, the temporal causal-effect relationships are more precise because the time points can be any time point during the execution interval of operators. The relationships provide an expressive representation and exploit the greatest level of concurrency between planned activities. Moreover, unlike SIADEX [22], which only defines causal-effect links between planned activities at the lowest level, the abovementioned cause-effect relationship formalism represents the causal structure between the tasks in different abstraction levels. In addition, when a cause-effect relationship $ce =< tp_i \xrightarrow{p_j} tp_j >$ is generated, a temporal constraint $t_i - t_j < 0$ is added to the STN of the search node for encoding it.

4.2.3 Concurrent controlling of multiple planned activities

The abovementioned planning algorithm is a state-based forward planning paradigm similar to SHOP2 and XePlanner, in which the generation process of planned activities is the same as the execution process. Despite the actions are generated one by one during the planning process, they are partially ordered, and their execution intervals may overlap. This section introduces a concurrent controlling mechanism of multiple operators with variable intervals.

For an instantaneous precondition $instantPre_i = (logicalExp_i\ @t_i)$ of a method or an operator, the planning algorithm chooses the earliest possible positive unified timed predicate $< p, true, t_1 >$ that supports the logical expression $logicalExp_i$. In addition, all the following conditions should also be satisfied.

(1) There is a positive timed predicate $< p, true, t_2 >$, which satisfies logical expression $logicalExp_i$.

(2) There is no negative timed predicate $< p, false, t_3 >$ in the planning state, such that $t_2 < t_3 < t_1$.

Tang P., Shen G.Q.P. (2015). Decision-making Model to Generate Novel Emergency Response Plans for Improving Coordination during Large-scale Emergencies, Knowledge-based Systems, Vol 90, Pages 111–128, DOI: 10.1016/j.knosys.2015.09.027, December. (SCI, 5-Year impact factor: 3.433, Ranked 17/130 in Computer Science, Artificial Intelligence by JCR in 2015).

(3) There is no other positive timed predicate $< p, ture, t_4 >$, such that $t_4 < t_2$, and satisfing the above two conditions.

During the planning process, the operators are applied sequentially in the planning state. An operator a is applicable in the temporal enhanced planning state if the following conditions are satisfied.

(1) The instantaneous and interval preconditions of this operator a are satisfied in the current planning state.

(2) The effects of this operator a do not interfere with the current planning state and any recorded invariant conditions.

3) A negative timed predicate $< \neg p, t_1 >$, which interferes with the invariant conditions $< p, (t_2, t_3) >$ of this operator a, does not exist in the planning state.

In addition, the interference is defined as the violation of any of the following conditions:

(1) If a timed predicate $< p, false, t_1 >$ exists in the planning state that causes $\neg p$ at time t_1, then the effect of this operator a that caused p should be added after time t_1. In the same way, if a timed predicate $< p, true, t_2 >$ exists in the planning state that causes p at time t_2, then the effect of action a that caused $\neg p$ should be added after time t_2.

(2) If a effect of this operator a deletes a predicate p and is protected by an invariant condition $< p, (t_1, t_2) >$ in the current planning state, then this operator cannot delete the predicate before t_2.

(3) If this operator a defines an invariant condition $< p, (t_1, t_2) >$ and a negative timed predicate $< \neg p, t_3 >$ exists in the current planning state that causes $\neg p$ in time t_3, then the negative timed predicate should occur after this invariant condition terminates. That is to say, the time constraint $-\infty \le t_2 - t_3 \le 0$ is added to the current STN. (In this paper, the end time of the invariant condition happens before the negative timed predicate, that is, time point t_3 is ordered to time point t_1.)

According to the abovementioned concurrent controlling mechanism for ensuring parallelism of multiple planned activities, all the cause-effect relationships are discovered and encoded by the STN when primitive tasks are applied. As a result, the generated actions are arranged properly and do not interfere with one another.

5. Incremental temporal management method based on semi-STN

The abovementioned decision-making model that aims to support the development process of emergency response plans provides excellent expressing capability to represent qualitative and quantitative time constraints. When the compounded tasks are decomposed or when primitive tasks are applied to expand the search space, the time constraints are generated and added to the search nodes to encode interdependencies between the tasks. Obviously, temporal conflicts may occur. In the presented decision-making model, the start and end times of all the tasks are represented explicitly. Consequently, the entire STN is larger compared to existing planning paradigms, and more planning time is required to check the temporal consistency of the underlying STN. This section proposes a temporal constraint propagation algorithm called Prop-PC2-STP, which takes full advantage of the hierarchical decomposition structure induced by the HTN planning process. This algorithm is embedded in the planning process for propagating time constraints.

5.1 Extended simple time net

Our decision-making model provides an expressive representation of time constraints between tasks. During the planning process, the generated time constraints, which are either pre-defined beforehand or induced by the plan-refining process, are added incrementally to the STN for encoding the independencies between tasks in a search node. On the one hand, when a compounded task is decomposed, four types of time constraints are generated and should be encoded by the STN. The details include the following: C-Type1: time constraints produced by checking the satisfaction of the preconditions in the method, as shown in 4.2.2, C-Type2: time constraints encoding the start time of the parent task is ordered to the start time of all its child tasks, and those representing the end time of all the child tasks are before the end time of the parent task, C-Type3: time constraints encoding the start time of all the child tasks are ordered to their end time, C-Type4: qualitative and quantitative time constraints between the child tasks, which are defined in the method formalism. On the other hand, when a primitive task is applied, two types of time constraints are generated, and are listed as follows: P-Type1: time constraints produced by checking the satisfaction of the preconditions for the operator unifying the primitive task, as shown in 4.2.2, and P-Type2: time constraints produced by applying the concurrent controlling rules.

When all the time constraints mentioned above are added to the STN and remain consistent, the current plan-refining steps can be executed. The STN is a framework widely used for checking temporal consistency and deriving the minimal network [34]. Aiming at improving the effectiveness of time management, the traditional STN is extended to encode all time constraints and hierarchical decomposition structures induced by the HTN planning process. First, the time point cluster structure is recorded to divide the underlying STN into multiple smaller subnets. Second, the cause-effect relationships defining on two time points representing the interdependencies between different tasks are represented explicitly. The definition of the extended STN is stated as below.

Definition 10: The extended STN has the form $exSTN =< tpSet, tcSet, cauEftSet, clusterStructure >$.

(1) $tpSet = \{TR, tp_{2i-1}, tp_{2i}\}$ ($1 \le i \le L$, where L is the number of all the tasks) is a set of time points representing the start or end times of all tasks in the hierarchical task network and the reference time point of the planning process.

(2) $tcSet = \{tc_j =< a \le tp_j^1 - tp_j^2 \le b >\}$ ($1 \le j \le M$, where M is the number of all the time constraits) represents the time constraints in the extended STN.

(3) $cauEftSet = \{ce_k =< tp_k^1 \xrightarrow{p} tp_k^2 >\}$ ($1 \le k \le N$, where N is the number of all the cause-effect relationships) is a set of cause-effect relationships as shown in Definition 9.

(4) $clusStruct = \{C_i =< tpSet_i, ngClusList >\}$ is the time point cluster structure consisting of a set of time points subcollection in the entire STN and their intersection set, called separation. The variable $tpSet_i$ is a set of time points in time point cluster C_i, and the variable $ngClusList = \{ng_k =< sep_i^j, C_j >\}$ records a list of

neighboring time point clusters. The variable ng_k describes that C_j is the neighbor time point cluster of C_i, and there is a separation sep_i^j between these two time point clusters. The separation sep_i^j records the time points, which are both in C_i and C_j simultaneously.

During the planning process, when a cause-effect relationship $ce_k =< tp_k^1 \xrightarrow{p} tp_k^2 >$ is generated, it is added to the cause-effect relationship set $exSTN.cauEftSet$. At the same time, the induced time constraint $tp_k^1 - tp_k^2 < 0$ is added to $exSTN.tcSet$, and the temporal consistency of the current extended STN is checked.

Each time point cluster always consists of TR and the start and end time points of a compounded task and its child tasks. Moreover, the first time point cluster in the extended STN is initialized by TR and the start and end time points of all the tasks representing the incident objectives. During the planning process, the time point cluster structure is expanded as the following:

Rule 1: When the time constraints of C-Type1, P-Type1, and P-Type2 are generated and added to the extended STN, if the pre-time point $preTp$ and post-time point $postTp$ are in the same time point cluster C_j for each time constraint $a \le preTp - postTp \le b$, then the time point cluster structure of the STN remains unchanged. Otherwise, given that the pre-time point $preTp$ belongs to time point cluster C_i and that the post-time point $postTp$ belongs to time point cluster C_j, the pre-time point $preTp$ is added to time point cluster C_j and the separator between time point clusters C_i and C_j.

Rule 2: When the time constraints of C-Type2, C-Type3, and C-Type4 are generated, given that the start and end time of the compounded task are found in time point cluster C_i, a new time point cluster C_j is created and initialized by TR . Moreover, the start and end time of the compound task to be decomposed and those of all its subtasks are added to time point cluster C_j. The separator between time point clusters C_i and C_j is initialized by the start and end times of this compounded task and TR.

As a result, the time point cluster C_j is called the affected time point cluster, and the sub-STN underlying C_j is called the initially affected sub-STN, to which the generated time constraints are added. From the abovementioned analysis, the entire STN is divided into multiple sub-nets underlying each time point cluster. Each sub-net encodes all the time constraints underlying the time points in the associated time point cluster.

5.2 Temporal propagation algorithm

The study takes full advantage of the hierarchical decomposition structure to propagate time constraints on the underling STN. A new time propagation algorithm, called Prop-H-STN, is proposed and triggered once the time points and their underling time constraints are added incrementally. The Prop-H-STN algorithm is triggered in three cases during the planning process. First, time constraints of C-Type1 are generated and added to the extended STN when the precondition in the method is checked. Second, the time constraints of C-Type2, C-Type3, and C-Type4 are generated and added to the extended STN when a compound task is decomposed. Third, the time constraints of P-Type1 and P-Type2 are generated and added to the extended STN when a primitive task is applied.

In the existing temporal HTN planner, the added time constraints are propagated individually on the entire underling STN directive. All generated time constraints in Prop-H-STN are instead added to the initial affected sub-STN and are propagated simultaneously. The time point cluster structure changes according to Rules 1 and 2 in Section 5.1. The detailed procedures of Prop-H-STN algorithm is shown in Figure 3.

```
0: function prop-H-STN( $exStn$ , $tcSet$ , $C_j$ )
1:     propagate $tcSet$ in initial affected sub-STN underling $C_j$ ;
2:     for each neighbor time point cluster $C_i$ of $C_j$
3:         for each $tc.preTp \in C_i \cap C_j$ and $tc.postTp \in C_i \cap C_j$
4:             $tightenTcSet \leftarrow$ { $tc \mid tc$ is tighten after sub-STN underling $C_j$ is propagated};
5:         end for each
6:         if ( $tightenTcSet$ is not null)
7:             prop-incremental-PC2( $edStn$ , $tightenTcSet$ , $C_i$ );
8:         end if
9:     end for each
10: end function
```

Fig 3. Prop-incre-PC2 algorithm

The inputs of Prop-H-STN algorithm are an extended STN, a set of time constraints, and the affected time point cluster. All time constraints are added to the initial affected sub-STN underling of the affected time point cluster C_j. First, the time constraint propagation algorithm (Figure 6) is applied to propagate and check the temporal consistency in the underling sub-STN of the current affected time point cluster C_j (Line 1). For separators between time point cluster C_j and each of its neighbor time point clusters C_i, time constraint is added to the variable *tightenTcSet* (Lines 3–5) once it is tightened after time propagation. This algorithm continues to propagate time constraints recursively in *tightenTcSet* (Lines 6–9) if the variable is not null. The algorithm stops once no time constraint in the sub-STN can be tightened. Thus, all underling sub-STNs of each time point cluster are local minimal. That is, the entire STN, which is a partial minimal network, is consistent [46].

All local minimal time constraints are propagated in the sub-STN underling of the neighbor time point cluster if the time constraints defined on the time points in separators between current time point cluster and its neighbor time point clusters are tight.

The temporal propagation process in a sub-STN, as shown in Line 1 of Prop-H-STN algorithm, tightens the edges representing time constraints in the sub-STN underling a time point cluster. The details of the procedure are listed in Figure 4.

```
0: function propagate-subSTN( $tcSet$ , $sub\_STN$ )
1:   $Q \leftarrow \varnothing$ ;
2:   for each time constraint $tc \in tcSet$
3:       for each time point $tp \in C_j.tpSet$
4:           if $tp \neq tc.preTp$ and $tp \neq tc.postTp$
5:               $Q \leftarrow \cup\{(tc.preTp, tc.postTp, tp)\}$ ;
6:           end if
7:       end for each
8:   end for each
9:   while $Q \neq \varnothing$
```

```
10:         select and delete a triangle from $Q$;
11:         $Revise(\Delta)$;
12:         for each $tc$ having been updated in triangle $\Delta$
13:            for each time point $tc \in C_j.tpSet$
14:                 $Q \leftarrow \cup\{(tc.preTp, tp, tc.postTp), (tc.postTp, tp, tc.preTp)\}$
15:            end for each
16:         end for each
17:     end while
18:     end function
```

Fig 4. Time propagation algorithm in a sub-STN

The above algorithm handles multiple time constraints in variable $tcSet$ and simultaneously propagates them in the current sub-STN. The associated triangles of a time constraint are inspired by the ΔSTP algorithm [47], which provides a new perspective on temporal problems as composed by a set of triangles. These triangles consist of vertexes representing two time points of this time constraint and each time points in a given time point cluster, except for the two former ones. The variable Q in propagate-subSTN algorithm records all triangles to be checked (Line 1) and are initialized by the associated triangles for each input time constraint (Lines 2–8). In each iteration, a triangle is selected and computed (Line 10). In line 11, one edge of triangle Δ is tightened and updated similar to PC2 algorithm, or all three edges are tightened at once compared to the ΔSTP algorithm. Given that each tightened edge represents a time constraint, their associated triangles are added to the variable Q. They can be added to the front, the end, or any position in the queue. The manner in which the triangles are inserted in the queue also affects the performance of this temporal propagation algorithm. The experiments will be introduced in section 7. Finally, the algorithm terminates, and a minimum sub-STN is obtained once the variable Q is null.

6. How to extract the emergency response plan

This section introduces the method for extracting the emergency response plans of the new structure from the generated planning information once the planning process terminates. The generated planning information, which is recorded by the data structure of the enhanced hierarchical task network, as well as the generated search node, is introduced in above sections. The emergency response plan consists of two elements. The first element keeps a trace of the selected decompositions during the HTN exploration, which records a sequence of decision points while planning for emergency response. The second element corresponds to the action plan, which consists of a set of planned activities in the lowest level of temporal refining task network. The extraction of the two elements is introduced in this section.

6.1 Extracting the temporal refining task network

A method of constructing the temporal refining task network (Section 2.3) is proposed according to the information elements generated by the planning process. The detailed process of extracting each element in the temporal refining task network is listed as follows:

(1) The extended task node in the temporal refining task network is defined as $exTaskNode_i = < task_i, st_i, et_i, tType_i, conList_i, parTask_i, childTaskSet_i >$. The first three elements are the same as the relevant elements in the formalism of the unifying task plus $taskPlus_i$ in an enhanced hierarchical task network.

The variable $desConList_i = \{desCon_i^j\}$ representing decision contexts for refining the tasks is a list of decision conditions $desCon_i^j$. According the enhanced hierarchical task network formalism, the condition has the form $desCon_i^j =< tp_i^j, pre_i^j, varBindArray_i^j >$. This form indicates that the precondition pre_i^j should be satisfied by an emergency situation with variable binding array $varBindArray_i^j$ at time point tp_i^j, when a compound task node $exTaskNode_i$ is achieved by all of its subtasks or a primitive one is executed by an operator instance. The variable tp_i^j is a time point representing the start, end, or any time point during the interval of task $task_i$. Given that the task plus $taskPlus_i$ in the enhanced hierarchical task network is associated with a given task node $tNode_i$, a decision condition $desCon_i^j =< tp_i^j, refinPrecon_i.preList_i^j, refinPrecon_i.varBindArrayList_i^j.varBindArray_k\ )$ is generated for each $refinPrecon_i^j$ in $taskPlus_i.refinPreconList_i$. Finally, the variable $parTask_i$ is the task node unified to the task plus $taskPlus_i$ of the current task node and is extracted by the same procedure. The variable $childTaskSet_i$ is a set of task nodes unifying to the task pluses in $taskPlus_i.refin\Pr econ_i^j.varBindArrayList_i^j.varBindArrayMean_k.children_k$.

(2) The variable $exStn$ is initialized by the extended STN in the output search node.

(3) The planned activity has the form $act_i =< head, delPre, addPre >$, which describes the detailed execution process of a specific primitive task. The name of the planned activity $head$ is initialized by the head of the associated operator instance. The negative execution effects, $delPre = \{delEvent_i =< predicate_i, -, tp_i >\}$, are a set of events that represent a negative temporal predicate instance, and are initialized by a deleted list of instantaneous and delayed effects in the associated operation instance. The defined predicate $predicate_i$ is deleted from the current state when an event $delEvent_i$ is triggered at time tp_i during execution. The positive execution effects, $addPre = \{addEvent_i =< predicate_i, +, tp_i >\}$, are also initialized by the added list of instantaneous and delayed effects. The defined predicate $predicate_i$ is added to the current state when an event $addEvent_i$ is triggered at time tp_i. Thus, the emergency situation changes during the execution interval of the planned activities.

Therefore, all elements in the temporal refining task network can be extracted from the generated search node $curNode$ and the enhanced hierarchical task network $enHieTaskNet$ in the presented decision-making model.

6.2 Extracting the emergency response business process

The emergency response business process represents the execution flow of the emergency response plan. In the processs, the planned activities represent the execution process of all operation instances in the generated search node (Section 2.3). The order relationships between the activities are encoded by the semi-structure STN. The process of extracting elements of the emergency response business process is introduced in this section.

The order relationships between tasks are defined by the semi-structure STN if their start and end time points are in the same time point cluster. That is, one task is ordered to another task if the end time point of the former task is before the start time of the latter one. The ordered relationships between tasks are also defined in different levels. If one task is ordered to another task, all planned activities achieving the former task are before those achieving the latter one.

Thus, all the order relationships between each pairs of planned activities are generated. In this paper, these relationships are represented by a boolean matrix called adjacent matrix. The ith planned activity should be executed before the jth one if the element line i and row j are true. The ith planned activity is called first planned activity if all the elements in the ith line are false. That is, the ith planned activity should be executed first. The jth planned activity is called the last planned activity if all the elements in the jth row are false. That is, no other planed activities should be executed after the jth planned activity.

The immediate previous action plus set $actPlus_i.Prev_i$ and immediate successor action plus set $actPlus_i.Succ_i$ for each planned activity plus $actPlus_i =< act_i, Prev_i, Succ_i >$ in the emergency response business process are initialized according to the above adjacent matrix. The reachable matrix of this adjacent matrix, which represents a planned activity ordered to all others, is computed. Then, for each successor action plus set $actPlus_i.Succ_i$ of each planned activity, if there exist a planned activity $actPlus_j$ in $actPlus_i.Succ_i$ is before a planned activity $actPlus_k$ in $actPlus_i.Succ_i$ according to the reachable matrix, the plan activity $actPlus_k$ is removed from the variable $actPlus_i.Succ_i$.

Two virtual action plus $Start$ and End, which represent the source and sink nodes respectively, are added to the business process. This addition is inspired by the workflow definition. The variable $Start.Prev$ is initialized by an empty set. The variable $Start.Succ$ is initialized by all the first planned activities. Moreover, the virtual action plus $Start$ is added to the immediate previous action plus set of all the first action pluses. The variable $End.Succ$ is also initialized by an empty set. The variable $End.Prev$ is initialized by all the last planned activities. Moreover, the virtual action plus End is added to the immediate successor action plus set of all last action pluses. Hence, the execution process emergency response business process starts execution from the virtual action plus $Start$ and terminates at the virtual action plus End.

7. Case study and experimental results

A practical case of typhoon evacuation and the experimental results are presented in this section to demonstrate the applicability of the presented decision-making model for providing support to emergency command operations in EOC. First, a typhoon evacuation domain that reflects the characteristics of emergency response is introduced. An emergency response plan generated by a decision-making model for coping with an emergency situation case is presented. The decision-making model is also compared with existing planning paradigms. Finally, a set of experiments are performed to show the performance of Prop-H-STN algorithm based on the semi-structured STNs embedded in the model.

7.1 A typhoon evacuation domain

A typhoon is a typical large-scale disaster in China’s South-East coastal areas. Given the strong winds and rainstorms, collaboration and coordination between multiple responding organizations are required to achieve identified incident objectives [48]. Such objectives include evacuating and settling residents in low-lying communities, controlling floods, and patrolling water conservancy facilities. In this section, a typhoon evacuation domain is designed to test the decision-making model by investigating a local jurisdiction of Shenzhen in the southeast coastal region of China.

Fig. 5. Sketch of an urban region in China's South-East coastal area

This urban region (Fig. 5) is in coastal area under the foot of the hills, with a river flowing through it. Seven low-lying communities are along the sea and river. The residents may be in danger and should be evacuated to assigned shelters once the orders are received. These communities, including Com-A, Com-B, Com-C, Com-D, Com-E, Com-F, and Com-G, lie in locations Loc 1, Loc 5, Loc 11, Loc 17, Loc 19, Loc 20, and Loc 21, respectively. Residents in Com-A and Com-B should be evacuated to Shelter-A in Loc 9. The assigned shelter of residents in Com-C is Shelter-B in Loc 11. The shelter of Com-D and Com-E residents lies in Loc 14. The shelter of residents in Com-F and Com-G is Shelter-D is in Loc 13.

When a typhoon or rainstorm comes, emergency managers in EOC in Loc 7 should assess and identify the communities in danger and evacuate residents in these low-lying areas to specific shelters before the disaster happens. Once incident orders are received, task forces from multiple responding organizations, such as police, fire control, medical, civil administration, and transport departments, respond and carry out specific tasks to evacuate and settle residents. Close collaboration and cooperation among the responding organizations are essential to achieve a coherent response to typhoon disasters.

7.2 Experimental results of application case

In a simulated emergency situation, the typhoon will arrive in 12 hours, the water level at Station 1 in location Loc 22 is 23 meters, and the water level at Station 2 in Loc 16 is 28 meters. Emergency managers in EOC locating in Loc 7 identify two incident objectives by assessing the emergency situation. One objective with the priority “1”

Tang P., Shen G.Q.P. (2015). Decision-making Model to Generate Novel Emergency Response Plans for Improving Coordination during Large-scale Emergencies, Knowledge-based Systems, Vol 90, Pages 111–128, DOI: 10.1016/j.knosys.2015.09.027, December. (SCI, 5-Year impact factor: 3.433, Ranked 17/130 in Computer Science, Artificial Intelligence by JCR in 2015).

describes that residents in Com-A should be evacuated to the assigned shelter within 18 hours as the deadline. The other objective with priority “2” describes that residents in Com-F should be evacuated to the assigned shelter within 24 hours. The emergency response plan automatically generated by the presented decision-making model is shown in figure 6 and figure 7, and achieves the identified incident objectives in the given emergency situation. The temporal refining task network of the plan is presented in figure 6, where the objective decomposition structure is described clearly. The yellow rectangles represent compound tasks, the red ones represent primitive tasks, and the blue ones represent planned activities. All the planned activities and order relationships are represented by the emergency response business process as shown in figure 7.

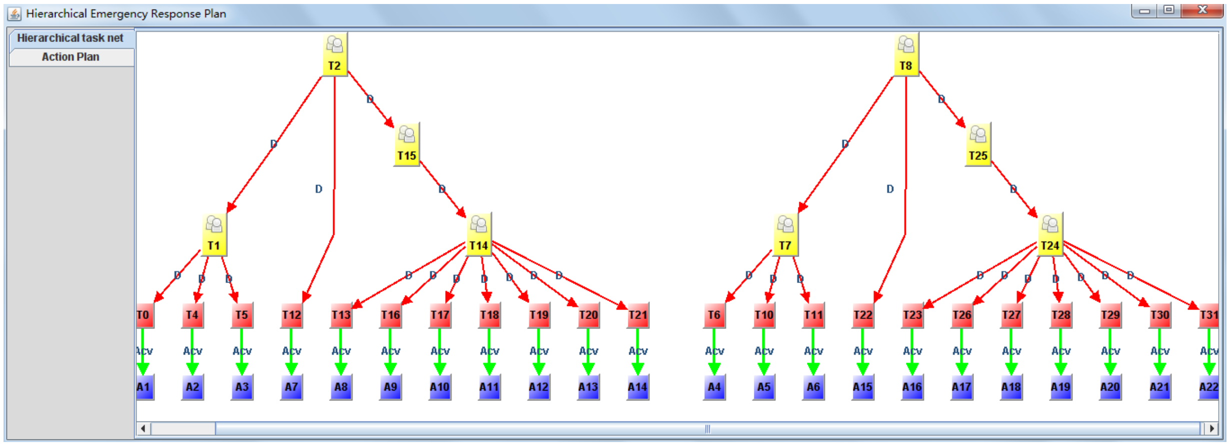

Fig. 6. Temporal refining task network of the generated emergency response plan in the typhoon evacuation domain

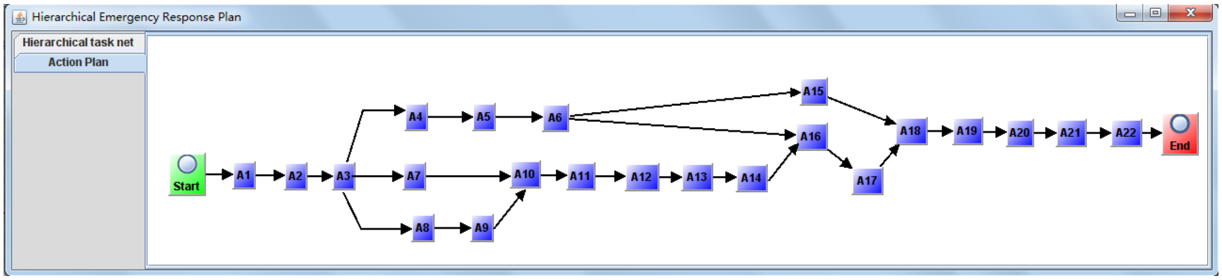

Fig. 7. Emergency response business process of the emergency response plan in the typhoon evacuation domain

7.3 Comparison of existing HTN planners and our decision-making model

Two experiments are performed to test the performance of our decision-making model. In the first experiment, XePlanner is compared with our decision-making model in the typhoon evacuation domain. In the other experiment, a domain-independent heuristic forward chaining planner Sapa [49] and the presented decision-making model are performed on the same planning problems in ZenotravelTime domain [50].

(1) Comparison of XePlanner and our decision model with typhoon evacuation domain

Our decision-making model and Xeplanner [33] are state-based forward planners applied in generating action plans for emergency response and can encode multiple types of domain knowledge in emergency management. Comparing with our decision-making model, STN only defines tasks in the lowest level of the hierarchical task network in XePlanner. Therefore, the STN underling the hierarchical task network has a significantly smaller scale in this planner. Given that XePlanner and our decision-making model are developed for the same purpose and applied in planning for emergency response, a set of planning problems are selected randomly in typhoon evacuation domain to test their performance. The plan metric value of generated action plans is quadruple make-span of the produced action plan in XePlanner. The make-span of a generated action plan is defined as the earliest end time of the emergency response business process in the generated emergency response plan in our decision-making model, because the generated action plans are temporal flexible. The planning time of solving all the problems and the plan metric values of produced action plans are listed in Table 1.

Despite the larger scale of the underling STN in our model, our decision-making model performs better than the XePlanner in planning time. The main reason for the reduction in planning time is the high performance of incremental temporal management method, which propagates time constraints on semi-STN introduced in Section 5. The method is more efficient than the PC-2 embedded in XePlanner when propagating temporal constraints. This finding is also verified in Section 7.4. The plan metric values of produced action plans are the same, because the similar state-based forwarding and non-backtracking algorithm is used to expand the search space in these two planning paradigms. In fact, our decision-making model is developed by extending XePlanner to provide temporal flexibility and higher computational performance. The experimental results demonstrate that our decision-making model is practical to solve real-world emergency response planning problems in such short time. That is very important during emergency command operation, where decisions should be made quickly.

Table 1. Comparison of our decision-making model and XePlanner

Problem	XePlanner		Our decision-making model	
	Planning time (second)	Plan metric value	Planning time (second)	Plan metric value
Problem 1	59.751	25.5	0.249	25.5
Problem 2	11.360	29.5	0.265	29.5
Problem 3	78.822	25.5	0.266	25.5
Problem 4	32.059	18.0	0.484	18.0
Problem 5	37.987	29.5	0.499	29.5
Problem 6	47.362	17.17	0.484	17.17
Problem 7	100.738	24.5	1.061	24.5
Problem 8	92.852	25.4	0.983	25.4
Problem 9	98.077	30.4	1.077	30.4
Problem 10	147.446	19.8	1.061	19.8

(2) Comparison of Sapa and our decision model in ZenotravelTime domain

This experiment is performed to compare the performance of our decision making model with that of existing state-of-the-art planners. The selecting planner is Sapa [49], which is a domain-independent heuristic forward chaining planner which can handle durative actions, metric resource constraints, and deadline goals. Because Sapa cannot express the characteristics of typhoon evacuation domain, ZenotravelTime domain is selected in this experiment, which is the benchmark of the third international planning competition [50]. A set of planning problems are generated randomly and are solved by Sapa and our decision-making model. The plan metric value of produced action plans generated by Sapa is quadruple make-span of the generated action plan similar to XePlanner. The plan metric value of action plans generated by our decision-making model is defined as the previous experiment. The experimental results are shown in Table 2.

Our decision-making model has outperformed Sapa in almost all problems, in both the planning time and the plan metric value of generated action plans. Compared with Sapa, our model provides high expressiveness and temporal flexibility for emergency response domain. Despite the additional computation overload of our model for handling time constraints, the planning time is short in almost all problems except Problem 2. According to the literature, Sapa is capable of handling the multi-objective nature of metric temporal planning to generate optimal action plans [49]. However, the plan metric values of produced action plans generated by our planner are more optimal in all problems than that of Sapa. These results demonstrate that our planner can handle preferences effectively during planning process.

Table 2. Comparison of Sapa and our decision making model

Problem	Sapa		Our decision-making model	
	Planning time (second)	Plan metric value	Planning time (second)	Plan metric value
Problem 1	0.086	20.61	0.056	13.56
Problem 2	0.099	24.55	0.115	24.08
Problem 3	0.082	23.43	0.031	23.43
Problem 4	8.266	26.98	0.062	15.71
Problem 5	0.095	29.32	0.062	25.84
Problem 6	0.101	47.92	0.094	38.30
Problem 7	0.142	22.45	0.053	13.66
Problem 8	0.149	27.35	0.091	13.66
Problem 9	0.158	45.72	0.078	20.14
Problem 10	0.172	61.24	0.150	22.66
Problem 11	0.183	44.95	0.289	26.25
Problem 12	0.185	53.65	0.155	22.90
Problem 13	0.174	56.69	0.247	36.73
Problem 14	1.469	199.83	0.156	78.94
Problem 15	13.984	234.76	0.259	78.94

7.4 Experimental results on the semi-structure STN

A key characteristic of our decision-making model is that a semi-structure STN is embedded in it. The semi-structure STN encodes and handles time constraints defined on the underling the hierarchical task network. An incremental temporal management method is also proposed to propagate the generated time constraints incrementally during the planning process. In this section, we design a set of experiments in typhoon evacuation domain to evaluate the performance of the time management method based on the semi-structure STN introduced in section 5.

(1) Comparison of PC-2 algorithm and Prop-H-STN algorithm

PC-2 is the most popular incremental temporal propagation algorithm and is embedded in the existing HTN planners with temporal management function, such as SIADEX [22, 23], XePlanner [33], and SIPE-2 [30, 31]. Our decision-making model represents all start and end times of each task in the hierarchical task network, and the STN is of large scale and consumes more time to propagate generated time constraints. Hence, a more efficient time propagation algorithm is needed in our decision-making model. Prop-H-STN algorithm is proposed in Section 5 to propagate generated time constraints dynamically based on the semi-structure STN. A number of planning problems are generated randomly in the typhoon evacuation domain to evaluate the performance of the new temporal propagation algorithm. Moreover, the PC-2 and Prop-H-STN algorithms are embedded in the presented decision-making model to assume temporal management function. The experimental results are listed in Table 3. As shown, the Prop-H-STN algorithm sharply decreases the planning time compared with PC-2 algorithm.

Table 3. Comparison of planning times of PC-2 and Prop-H-STN

Problem	**Planning time (second)**	
	PC-2	**Prop-H-STN**
Problem 1	108.888	0.219
Problem 2	117.824	0.203
Problem 3	119.307	0.218
Problem 4	117.485	0.202
Problem 5	133.025	0.280
Problem 6	0.936	0.031
Problem 7	0.858	0.030
Problem 8	0.905	0.031
Problem 9	25.812	0.062
Problem 10	26.005	0.047

(2) Comparison of the three methods for managing the queue of triangles in propagating sub-STN

The updated edges in the Prop-H-STN algorithm can be added to the queue in three method, such as the front, the end, and any position in the queue. Ten sets of planning problems are generated randomly in the typhoon domain to evaluate these methods for managing the queue of triangles in our temporal propagation algorithm. The experimental results are shown in figure 8. The addition of the associated triangles of tightened edges to the end of the queue resulted in the algorithm outperforming the other two method when the temporal consistency is checked. The algorithm displayed the worst performance when the associated triangles were added to the front of the queue. Moderate performance was achieved when the associated triangles were added to any position of the queue. Hence, propagating the time constraints as early as possible across the time constraints graph is more effective. The propagation also requires less time to checek temporal consistency when the associated triangles of updated edges are added to the front of the queue.

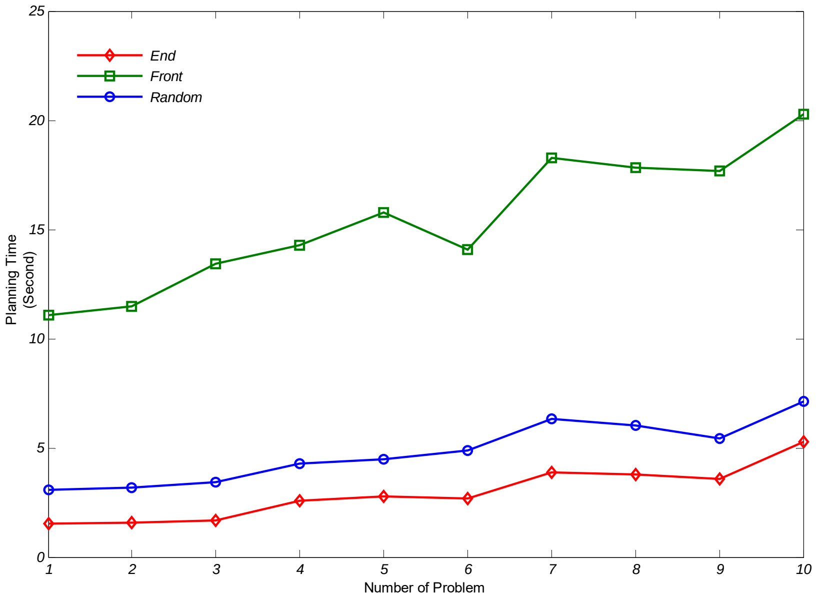

Fig. 8. Comparison of the three methods for managing the queue of triangles

Tang P., Shen G.Q.P. (2015). Decision-making Model to Generate Novel Emergency Response Plans for Improving Coordination during Large-scale Emergencies, Knowledge-based Systems, Vol 90, Pages 111–128, DOI: 10.1016/j.knosys.2015.09.027, December. (SCI, 5-Year impact factor: 3.433, Ranked 17/130 in Computer Science, Artificial Intelligence by JCR in 2015).

(3) Comparison for checking one edge and all edges of triangles simultaneously

Only one edge in the selected triangle Δ in PC-2 algorithm is computed and may be tightened and updated. Inspired by the perspective of ΔSTP algorithm [47], all the edges representing time constraints in selected triangle can be computed and updated as a whole during the planning process. Ten sets of planning problems are designed in the typhoon evacuation domain to evaluate the effectiveness of these two methods. The experimental results in figure 9 demonstrate significant improvements when our decision-making model simultaneously computes all edges in the selected triangle. The results indicate excellent performance in terms of CPU time when the presented decision-making model checks temporal consistency of all edges in a triangle simultaneously.

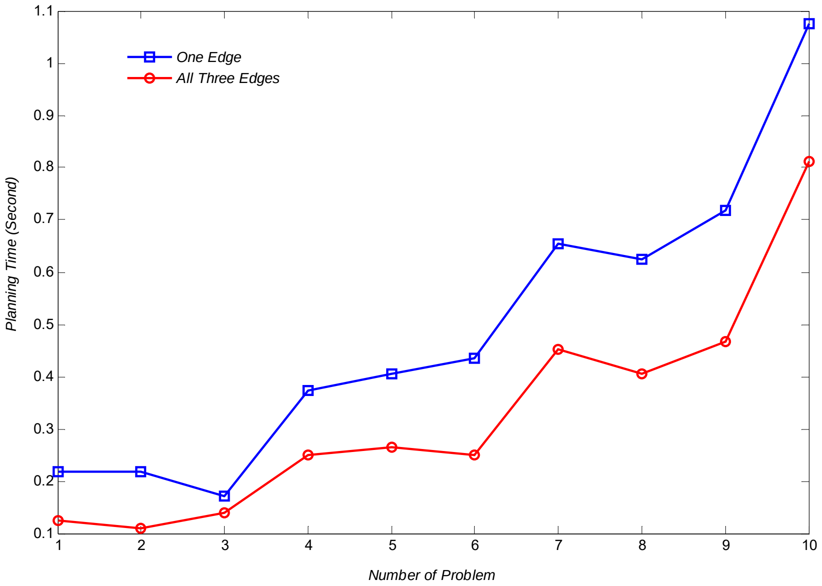

Fig. 9. Comparison of running times to check the temporal consistency in two methods

8. Related work and discussion

HTN planning and temporal constraint propagation provide the theoretical basis for the designing of the presented decision-making model in this paper. This section discusses literatures relevant to this research.

8.1 HTN planning

HTN planning is a branch of artificial intelligence planning technology that represents and handles hierarchies [51]. The first HTN planner, called NOAH, was proposed in 1975 [16]. Erol (1994) proposed the formal semantics of HTN planning that was more expressive than traditional planning technologies [52]. A series of HTN planners have been developed and applied to solve practical problems in real-world application domains. The most distinguished planning paradigms are the simple hierarchical ordered planner (SHOP) and its successor SHOP2, which won one of the top four awards in the 2002 International Planning Competition. SHOP and SHOP2 are distinguishable from others because they plan for tasks in the order that the tasks are executed. Thus, the planner knows the current state of the world at each step during the planning process [18]. HTN planners demonstrating this characteristic are called state-based forwarding HTN planners (i.e., SHOP, SHOP2, SIADEX, and XePlanner).

The ultimate objective is the application of HTN planning in real-world domains. We are interested mainly in the process of addressing the problems of developing emergency response plans by using HTN planning. It requires more expressive representation and capability than those provided by traditional HTN planning paradigms, which produce a sequence of actions and cannot satisfy the requirements of emergency command operation. Moreover, these planners cannot handle time constraints and actions explicitly executed in variable intervals.

A number of HTN planners have been applied in emergency response domain. SIADEX is an integrated HTN planning and scheduling paradigm that supports decision making during crisis episodes by enhancing the time reasoning of HTN planner [23, 24, 25]. The generated action plan is realistic with temporally annotated actions, in which the start time of actions is determined dynamically to provide flexibility. However, the actions are executed in fixed intervals. In addtion, it implements a depth-first search and backtracking algorithm during the search process. Such process is unlike the decision-making process for developing emergency response plans. This planner only implements time reasoning between primitive tasks and cannot handle possible time conflicts generated by decomposing compound tasks. Hence, the process cannot satisfy the proposed requirements in Section 2.2. XePlanner is a heuristic HTN planner for generating incident action plans at any time [33]. This planner can produce the first feasible plan quickly and improve the quality of the plans as more time is available. Moreover, the preferences of emergency managers and the priorities of incident objectives are effectively taken into account and handled during the planning process. However, the produced action plan is not temporally flexible, and the planning process consumes more time because of the time reasoning procedures. Thus, the proposed action plan cannot be applied to support real-world emergency command operation. This planner encodes time constraints on the task network with STN and handles them by the PC-2 algorithm. As a result, its low computation efficiency becomes an obstacle for solving real-world problems. Wang et al. enhanced the HTN planning paradigm to represent and reason hierarchical resources explicitly to provide support for emergency decision-making [53]. The resource and temporal constraints are encoded and propagated on resource timelines to handle hierarchical resource constraints. However, the expression capability cannot adapt to the application domain in this paper, and the produced plans are a sequence of rigid actions with invariable intervals.

Comparing with the existing state-based forwarding HTN planners, the presented decision-making model expresses the objective decomposition structure, record all decision nodes and their context to monitor plan execution process, provide temporal flexibility for adapting to the uncertain and dynamic nature of the response process, and handle the interdependencies and synchronization of response activities with a variable interval. It also represents the start and end times of each task explicitly in the hierarchical task network. Depite the associated STN in our model is of large scale and requires more planning time to check temporal consistency, a new time reasoning algorithm based on semi-STN is embedded to overcome the obstacle effectively.

8.2 Temporal constraints propagation

Time constraints are an enduring characteristic that should be addressed during the planning for emergency response in the emergency response domain. The domain knowledge can be applied to speed up temporal constraint propagation with structural information and prune unnecessary propagation effort. Lin proposes a new algorithm called ΔSTP, which triangulates STN and propagates time constraints, to achieve high performance of computing minimal network of STN [47]. This algorithm provides a new perspective on temporal problems, which are composed by a set of triangles, two of which are connected if they have one common edge. Therefore, constraint propagations can be performed according to this new graph of triangles. Inspired by this algorithm, we design our temporal constraints propagation algorithm based on triangles in STN. Yorke-Smith first exploited the structure of an HTN plan in performing temporal propagation on an underling STN [54]. The introduced algorithm, sibling-restricted propagation (SR-PC), transverses a tree of sub-STNs that corresponds to the expansions in the hierarchical task network. The considered STNs, encoding constraints between parent and child task nodes and between sibling ones, are smaller compared with the STNs corresponding to the entire plan. It demonstrates an order of magnitude improvement. However, it doesn’t operate on general temporal networks generated by planning process and cannot propagate time constraints incrementally. Instead, in the SR-PC algorithm, all time constraints are given at the beginning. As a result, the SR-PC algorithm cannot be embedded in HTN planning paradigms to assume time propagation function directive. Fusun presented temporal milestones in hierarchical task network to enable the complex synchronization of tasks and introduced an efficient temporal reasoning algorithm called the D-PC [55]. This algorithm propagates temporal constraints incrementally without re-computing the STNs induced by HTN and is a generalized version of SR-PC. However, this algorithm cannot handle temporal propagation in HTN planning process. Planken proposed IPPC algorithm to solve sparse STNs incrementally by enforcing partial path consistency [56]. However, the IPPC algorithm needs to triangulate the STN before propagating time constraints and is designed to handle general sparse STN without domain knowledge. Finally, the Prop-STP algorithm applies variable elimination to exploit the tree-decomposition method in which messages are represented compactly as sub-STNs and an efficient message passing scheme is designed to compute the minimal constraints of sub-STNs [46]. However, this algorithm requires the STN to be triangulated and operates over the set of maximal cliques of the triangulated constraint graph. The time complexity of Prop-STP is $\mathrm{O}(Kw^3)$, where K is the number of cliques and w is the induced tree width. Therefore, Prop-STN for STNs with known and bounded tree width achieves linear time complexity, indicating a substantial improvement over the use of PC-2 algorithm. However, the cluster tree structure is constructed by triangulating the STN in Prop-STP algorithm. In addition, time constraints are not added and propagated incrementally.

Inspired by temporal propagation algorithms, such as SR-PC and Prop-STP, the entire STN is divided into a number of sub-STNs to improve efficiency in the incremental temporal management method introdueced in section 5. The time point clusters in this temporal propagation algorithm are not constructed by triangulating the entire STN, or defined by parent and child tasks relationships. The temporal propagation algorithm divides the entire STN directive by considering the hierarchical task network and underling cause-effect relationships in our decision-making model. The presented time constraint propagation algorithm also handles multiple time constraints simultaneously in each sub-STNs and demonstrates a much higher performance comparing with PC-2.

9. Conclusion and future work

Joint emergency response plans are an effective method for coordinating multi-organizational response during emergencies. The formation of a joint emergency response plan is a real-world problem faced by emergency managers during emergency command operation. Enormous efforts are invested in designing decision support tools to develop emergency response plans. This research concentrates mainly on what emergency response plans are required during emergency command operations and how to generate them by integrating HTN planning and scheduling technologies. In this paper, an extended emergency response plan structure that accounts for the requirements of emergency command operation is proposed. The extended structure expresses the decomposition structure of the incident objective explicitly, records decision contexts, and provides temporal flexibility. A decision-making model is developed to generate this plan structure intelligently. This model presents several valuable extensions comparing with other existing state-based forwarding HTN planners. First, an enhanced hierarchical task network is designed to record traversed HTN exploration space of initial sets of incident objectives. Second, the generated plan is temporally flexible, in which the start and end times of actions are not pre-determined. That is important for handling temporal uncertainty in emergency response domain. A new concurrent controlling mechanism to ensure parallelism of response activities with variable interval is also proposed. Moreover, it enables representation of the dynamic emergency situation. Third, the start and end times of all tasks in the hierarchical task structure are represented explicitly. All the time constraints defined on them are encoded and handled by a semi-STN attached to the task network. The study also proposes a dedicated STN solver that takes full advantage of the decomposition structure induced by the HTN planning process to propagate on the underling semi-STN incrementally. Despite the scale of induced STN of our decision-making model is larger comparing with existing HTN planners, such as XePlanner, its computation efficiency is much higher. That enables our model to overcome the obstacles faced by other temporal HTN planners, such as SIADEX and XePlanner, which cost much more planning time when checking the consistency of STN. Empirical research on typhoon evacuation demonstrates that the model is suitable for solving real-world problems when planning for emergency response in practical application. Therefore, our decision-making model makes an actual contribution to HTN planning by extending the existing planning framework and temporal propagation algorithm.

Larger-scale emergencies are highly volatile. They change quickly and can have unpredictable consequences. These enduring characteristics determine that effective emergency management requires inter-organizational collaboration and needs to adapt quickly to dynamic situations [57]. Fast and dynamic decision making is necessary when making decisions within deadlines. Therefore, planning for emergency response is ubiquitous in nature. The action plans evolve as situations change and new information is received during emergencies. In our future work, we intend to design an integrated planning and execution model [58] based on this novel emergency response plans structure as support for the development, deployment, and repair of plans. It aims to provide support for emergency command operation. This approach will allow quick and effective response to changes in the environment, which has been advocated by many researchers.

Given the serious consequences of large-scale emergencies, generating emergency response plans should be implemented by human-machine cooperation to improve fidelity of action plans. Emergency managers should be supported to enable them to make good decisions on how to plan and respond to disastrous situation. However, computers cannot assume the role of commanders to evaluate incident status and make critical decisions on the appropriate response. Thus, the decisions involved in the development process of emergency response plans should be implemented collaboratively by decision support paradigms and emergency managers. One of our next research objectives is to design human computer interaction interfaces for providing cognitive-level support to emergency managers. This approach combines both HTN planning and mixed-initiative approaches. Emergency managers interact with the decision reasoning process of computational models to advance the development process of emergency response plans. Thus, not only the explicitly readable domain knowledge recorded by computer, but also implicit experiences in the mind of emergency managers are all integrated to make good decisions. Moreover, the reasoning capability of computers and evaluation capability of emergency managers are all applied to complement each other with advantages, which is the ultimate objective of human and computer collaboration process.

Acknowledgment

The work was supported by National Science Foundation Grant of China (Project No. 71303093), Humanities and Social Science Youth Fund of the Ministry of Education in China (Project No. 13YJC630145) and the Fundamental Research Funds for the Central Universities of China (Project No. 21612301).

References

[1] W. Smith, J. Dowell (2000) A case study of coordinative decision-making in disaster management. Ergonomics 43(8):1153-1166

[2] Rui Chen, Raj Sharman, H. Raghav Rao, Shambhu J. Upadhyaya (2008) Coordination in emergency response management. Communications of the ACM 51(5): 66-73

[3] Murray Turoff, Michael Chumer, Bartel Van de Walle, and Xiang Yao (2003) The design of a dynamic emergency response

management information system. Journal of Information Technology Theory and Application 5(4):1-36

[4] Murray Turoff (2002) Past and future emergency response information systems. Communications of the ACM 45(4): 29-32

[5] Rui Chen, Raj Sharman, H. Raghav Rao, Shambhu J. Upadhyaya (2007) Design principles for critical incident response systems. Information Systems and E-Business Management, 5(3):201-227.

[6] U.S. Department of Homeland Security. National Incident Management System. http://www.icisf.org/articles/NIMS.pdf

[7] Rui Chen, Raj Sharman, H. Raghav Rao, Shambhu J. Upadhyaya (2005) Design principle of coordinated Multi-incident emergency response system. In: Proceedings of the IEEE International Conference on Intelligence and Security Informatics(ISI), pp 81-98

[8] J. M. Tseng, C. Y. Kuo, M. Y. Liu, C. M. Shu. (2008) Emergency response plan for boiler explosion with toxic chemical releases at Nan-Kung industrial park in central Taiwan. Process Safety and Environment Protection. 86: 415-420.

[9] Scenario-Based Planning and Exercise, http://www.osha.gov/SLTC/etools/ics/prepare_implement.html#Scenario

[10] M. David, W. William Al. (2007) A cognitive model of improvisation in emergency management, IEEE Transactions on Systems Man and Crbernetics Part A-Systems and Humans. 37(4):547-561

[11] D. E. Wilkins, M. DesJardins (2001) A call for knowledge-based planning, AI Magazine 22(1): 99-115

[12] D. Mendonça (2007) Decision support for improvisation in response to extreme events: Learning from the response to the 2001 World Trade Center attack. Decision Support Systems 43:952– 967

[13] D. Mendonça, Giampiero E.G. Beroggi, William A. Wallace (2001) Decision support for improvisation during emergency response operations. International Journal of Emergency Management 1(1):30-39.

[14] Alexander Smirnov, Michael Pashkin, Nikolai Chilov, Tatiana Levashova. Context-Based Disaster Management Support. In: Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications.

[15] K. Erol, J. Hendler, and D. Nau, Complexity results for Hierarchical Task Network planning, Annals of Mathematics and Artificial Intelligence 18 (1) (1996), 69-93.

[16] I. Georgievski, M. Aiello (2015) HTN planning: Overview, comparison, and beyond, Artificial Intelligence 222:124-156

[17] D. E. Wilkins and M. desJardins (2001) A call for knowledge-based planning. AI Magazine 22 (1):99-115

[18] D. Nau, A. Tsz-Chiu, O. Iighami, U. Kuter, J. W. Murdock, D. Wu, F. Yaman (2003) SHOP2: An HTN planning system, Journal of Artificial Intelligence Research 20:379-404

[19] D. Nau, Y. Cao, A. Lotem, H. Munoz-Avila (2001) The SHOP planning system, AI Magazine 22 (3):91-94

[20] H. Munoz-Avila, D.W. Aha, L. Breslow, D. Nau (1999) HICAP: An Interactive Case-Based Planning Architecture and its Application to Noncombatant Evacuation Operations. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence. AAAI, pp.870-875

[21] D. Nau, A. Tsz-Chiu, O. Ilghami, U. Kuter, D. Wu, F. Yaman, H. Muñoz-Avila, J. W. Murdock, Applications of SHOP and SHOP2, IEEE Intelligent Systems 20 (2) (2005), 34-41

[22] L. Castillo, J. Fdez-Olivares, Garcí a-Pérez, F. Palao (2006) Temporal Enhancements of an HTN Planner, Lecture Notes in Computer Science , Springer Verlag, Spanish pp.429-438.

[23] L. Castillo, J. Fdez-Olivares, Garcí a-Pérez and F. Palao (2006) Efficiently handling temporal knowledge in an HTN planner. In: Proceedings of Sixteenth International Conference on Automated Planning and Scheduling. Association for Computing Machinery, pp 63-72.

[24] M. De La Asunción, L. Castillo, J. Fdez-Olivares, Ó. Garcí a-Pérez and F. Palao (2005) SIADEX: An integrated planning framework for crisis action planning. AI Communications 18 (4):257-268.

[25] Marc de la Asuncion, Luis Castillo, Juan Fdez-Olivares, Oscar Garcıa-Perez, Antonio Gonzalez, Francisco Palao, Knowledge and plan execution management in planning fire fighting operations

[26] S. Biundo and B. Schattenberg (2001) From abstract crisis to concrete relief. a preliminary report on flexible integration on nonlinear and hierarchical planning. In: Proceedings of the 6th the European Conference on Planning (ECP-01). Springer Verlag, Spain, pp.157-168.

[27] B. Schattenberg and S. Biundo, M. Ghallab, J. Hertzberg and P. Traverso (2002) On the Identification and Use of Hierarchical Resources in Planning and Scheduling. In: Proceedings of the 6th International Conference on Artificial Intelligence Planning Systems. pp.263–272.

[28] A. Tate, B. Drabble and R. Kirby (1994) O-Plan2: an Open Architecture for Command, Planning and Control, Morgan Kaoufmann

[29] C. Siebra and A. Tate (2003) I-Rescue: A Coalition based System to support Disaster Relief Operations, In: Proceeding of 3rd International Conference on Artificial Intelligence and Applications

[30] D. E. Wilkins, K. L. Myers, J. D. Lowrance, and L. P. Wesley (1995) Planning and reacting in uncertain and dynamic environments, Journal of Experimental and Theoretical Artificial Intelligence 7(1):197–227

[31] J. M. Agosta and D. E. Wilkins (1996) Using SIPE-2 to plan emergency response tomarine oil spills, IEEE Expert 11 (6):6-8

[32] D. E. Wilkins and R. V. Desimone (1994) Applying an AI Planner to Military Operations Planning, Intelligent Scheduling. Morgan Kaufmann Publishers Inc., San Mateo.

[33] Pan Tang, Hongwei Wang, Chao Qi,Jian Wang (2012) Anytime heuristic search in temporal HTN planning for developing

incident action plans. AI Communications 25:321~342

[34] R. Dechter, I. Meiri, J. Pearl (1991) Temporal constraint networks. Artificial Intelligence 49:61~95

[35] K. Amailef, J. Lu (2013) Ontology-supported case-based reasoning approach for intelligent m-Government emergency response services. Decision Support Systems 55:79~97

[36] M. Fox, D. Long (2003) PDDL2.1: An Extension to pddl for Expressing Temporal Planning Domains, Journal of Artificial Intelligence Research 20:61~124

[37] CA. Ntuen, O. Balogun, E. Boyle, A. Turner (2006) Supporting command and control training functions in the emergency management domain using cognitive system engineering. Ergonomics 49:1415-1436

[38] L. Hossain, J. Hamra, R. T. Wigand, S. Carlsson (2015) Exponential random graph modeling of emergency collaboration networks. Knowledge-based Systems 77:68-79

[39] GA. Bigley, KH. Roberts (2001) The incident command system: High-reliability organizing for complex and volatile task environments. Academy of Management Journal 44(6):1281-1299

[40] M. Ryan (2013) Planning in the emergency operations center, Technological Forecasting and Social Change 80(9):1725-1731

[41] National Incident Management System. http://www.icisf.org/articles/NIMS.pdf

[42] Barbara J. Grosz, Sarit Kraus (1996) Collaborative plans for complex group action. Artificial Intelligence 86(2):269–357

[43] Wei Chen. Designing an Extended Set of Coordination Mechanisms for Multi-agent System, PHD Thesis. University of Delaware, 2006

[44] C. Siebra (2005) Planning requirement for Hierarchical Coalitions in disaster Relief Domains. Expert Update 8 (1):20-24

[45] C. A. Ntuen, O. Balogun, E. Boyle, A. Turner (2006) Supporting command and control training functions in the emergency management domain using cognitive systems engineering. Ergonomics 49(12):1415-1436

[46] Neil Yorke-Smith, Hung H. Bui (2010) Efficient variable elimination for semi-structured simple temporal networks with continuous domains. The Knowledge Engineering Review 25(3): 337-351

[47] L. Xu, BY. Choueiry (2003) A new efficient algorithm for solving the simple temporal problem. In: Proceeding of 10th International Symposium on Temporal Representation and Reasoning/4th International Conference on Temporal Logic. pp:212-222

[48] R. McMaster, Chris Baber (2012) Multi-agency operations: Cooperation during flooding, Applied Ergonomics 43:38-47

[49] M. B. Do, S. Kambhampai (2003) Sapa: A Multi-objective metric temporal planner, Journal of Artificial Intelligence Research, 20:155-194.

[50] M. Fox, D. Long (2002) Domains of the 3rd international planning competition. In: Artificial Intelligence Planning Systems.

[51] M. Ghallab, D. Nau, P. Traverso (2004), Automated Planning: Theory & Practice, Morgan Kaufmann Publishers Inc.

[52] K. Erol, D.Nau, J. Hendler. (1994). HTN planning: Complexity and expressivity. In: Association for the Advancement of Artificial Intelligence conforence.

[53] Zhe Wang, HongWei Wang, Chao Qi, JianWang (2013) A resource enhanced HTN planning approach for emergency decision-making. Appllied Intelligence 38:226–238

[54] Yorke-Smith, N. (2005) Exploiting the structure of Hierarchical Plans in Temporal Constraint Propagation, 2005, In Proceeding of AAAI-05, Pittsburgh, PA

[55] Fusun Yaman, Brett Benyo, Alice M. Mulvehill (2009) Temporal Milestones in HTNs. In: Proceeding of the Twenty-Seventh AAAI conference on Artifical Intelligence, pp.1005-1011

[56] L´eon Planken, Mathijs deWeerdt, Neil Yorke-Smith (2009) Incrementally Solving STN by Enforcing partial path consistency, In Proceedings of the Twentieth International Conference on Automated Planning and Scheduling, pp129-136

[57] Murray Turoff, Connie White, Linda Plotnick, Starr Roxanne Hiltz (2008) Dynamic Emergency Response Management for Large Scale Decision Making in Extreme Events. In: Proceedings of the 5th International ISCRAM Conference. pp.462-470

[58] M. Ghallab, D. Nau, P. Traverso (2014) Theactor'sviewofautomatedplanningandacting: Apositionpaper. Artificial Intelligence. 208: 1-17