Dynamic Assignment of Geospatial-Temporal Macro Tasks to Agents under Human Strategic Decisions for Centralized Scheduling in Multi-Agent Systems

Reza Nourjou, Stephen F. Smith, Michinori Hatayama, Norio Okada, and Pedro Szekely

Abstract—**Problem: This paper addresses a centralized scheduling problem in multi-agent systems in which the incident commander (IC) of a disaster-response team aims to coordinate the actions of the field units (rational agents) to minimize the total operation time in uncertain, dynamic, and spatial environments.**

Objective: The purpose of this paper is to propose an intelligent software system that assists the IC in the dynamic assignment of geospatial-temporal macro tasks to agents under human strategic decisions. This system autonomously executes a heuristic algorithm to minimize the maximum total dependent duration according to human high-level strategies.

Results: The result is a schedule composed of macro decisions, each comprised of seven types of information: 1) what task type is going to be accomplished, 2) who (a subset of agents) are assigned to this assignment, 3) where this task is to be performed (a road segment or zone as a macro geospatial object) containing a subset of tasks, 4) when operations start, 5) when operations finish, 6) how many tasks are estimated to be completed, 7) what task types and the estimated number to be revealed (identified" or "enabled) in this location to complete this job.

Conclusion: This result, which is a feasible solution for the addressed problem, permits the IC to coordinate agents, partially specify the activities of the agents in time and space, minimize the overall execution time for all the tasks, calculate the correct time to revise the strategic decisions, evaluate the efficiency of the high-level strategy, and estimate the makespan.

***Index Terms*—Task assignment, centralized scheduling, multi-agent systems, macro tasks, disaster emergency response, heuristic algorithm, coordination, incident commander**

I. Introduction

The objective of scheduling is to minimize the overall execution time for all tasks by properly allocating them to processors/machines without violating the precedence constraints among the tasks [1]. The input to these problems is usually a directed acyclic graph (DAG) that provides precedence, dependency, priority among tasks, cost, and other information for the tasks [2]. The total run-time is called the makespan.

Urban search and rescue (USAR) is considered the major function of disaster emergency response operations. The objective of USAR is to reduce the number of fatalities in the first days after a disaster [3]. Scheduling is a crucial issue for incident commanders (ICs) who must manage and coordinate the activities of the field units (rational agents) during the disaster emergency response. It aims to identify and schedule the actions of the agents by appropriately assigning and allocating their tasks. Although scheduling enables the agents to accomplish tasks in the minimum overall time, it is difficult to solve this problem under uncertain, spatial, and dynamic situations.

A responder team such as the INSARAG (International Search and Rescue Advisory Group) has a hierarchical structure consisting of two levels. The lower level in the hierarchy includes the rational agents. They are spatially distributed in a geographic environment. They perceive their local environment, execute strategic decisions made by the top node, rationally make decisions regarding their own actions, coordinate their actions with each other in a distributed approach, execute tasks, and report to the operation center [4]. An IC, located in the top node, has a global and timely view of the entire disaster situation. His role is to coordinate and command the agents using strategic planning and scheduling in a centralized approach. As a result, a team is a society of cooperative agents that attempt to maximize the global utility of the team.

Coordination is the act of managing interdependencies among activities [5]. Coordination in disaster emergency response includes the management of task flow, recourse, information, decisions, and responders [6]. This paper recognizes the importance of five main types of coordination in a disaster-response team:

1) **Task Dependencies**: The *Enabling* dependency between tasks specifies that when a task is completed, it triggers the initiation of other, dependent tasks. In other words, the time that a disenabled task becomes enabled depends on the finishing time of the task to which it is dependent. Agents can carry out only tasks whose state is enabled.
2) **Action Dependencies**: Heterogeneous agents can perform tasks that have been previously revealed by the actions of other agents. Some agents may wait until their tasks are released by others.
3) **Redundant Actions**: Multiple agents may possess overlapping capabilities that allow them to be involved in the same tasks. This can cause conflict between actions or redundant actions.
4) **Information Sharing**: Information sharing allows

Manuscript received August 13, 2013; revised November 20, 2013.

Reza Nourjou is with the Informatics Graduate School and DPRI, Kyoto University, Japan (e-mail: nourjour@imdr.dpri.kyoto-u.ac.jp).

Stephen F. Smith is with the The Robotics Institute, Carnegie Mellon University, USA (e-mail: sfs@cs.cmu.edu).

Michinori Hatayama is with the Disaster Prevention Research Institute, Kyoto University, Japan (e-mail: hatayama@imdr.dpri.kyoto-u.ac.jp).

Norio Okada is with the School of Policy Studies, Kwansei Gakuin University, Japan (e-mail: okada-n@kwansei.ac.jp).

Pedro Szekely is with the Information Sciences Institute, University of Southern California, USA (e-mail: pszekely@isi.edu).

DOI: 10.7763/IJMLC.2014.V4.383

agents to know 1) the current state of the tasks, 2) what tasks are going to be performed, by whom and when, and 3) what tasks are estimated to be revealed when, and where. These data enable an IC to coordinate the agents and facilitate multi-agent scheduling.

5) **Agent Allocation**: Agents are considered as limited resources or machines that must be optimally scheduled and allocated to tasks.

Many good approaches have been developed to address the scheduling problem. However, there remain challenges to the problem that are addressed by this paper. We categorize these limitations and deficiencies as follows:

1) These approaches explicitly and fully schedule and specify actions for the rational and autonomous agents who are located in uncertain and dynamic environments. They attempt to associate a specific task with a specific agent. They do not provide an efficient approach for an IC because situational awareness that an IC has from the global picture is different from the perception that the agents have from their local environment.
2) They are automated systems that do not involve a human in the loop. The primary obstacle is the scale of the operation. It is currently unrealistic for a fully automated system to effectively determine all of the possibilities that may arise during the execution of the tasks in a complex environment.
3) They do not address the scheduling problems in which macro tasks are spatially distributed in a complex environment. Moreover, geospatial reasoning and geo-information analyses play a significant role in problem solving. Spatiotemporal macro tasks and characteristics of this environment make the assignment problem more difficult.

This paper addresses the capability of GICoordinator in macro action scheduling. The IC is equipped with a computer that runs an instance of GICoordinator. GICoordinator is a GIS (Geographic Information Systems)-based intelligent software system that collaborates with an IC (as a human) and assists the IC in the action coordination of agents in complex situations such as disaster emergency response [7]. This system autonomously executes an algorithm for the assignment of spatiotemporal macro tasks to rational agents under human strategic decisions in a spatial and dynamic environment. The goal of this algorithm is to minimize the maximum total dependent duration by the dynamic assignment of idle agents to these tasks.

The contribution of this paper is a heuristic algorithm that calculates a feasible solution for the addressed problem. This algorithm enables us to develop a software agent that can collaborate and cooperate with an IC on scheduling in multi-agent systems for agent coordination. It calculates a set of macro (semi-strategic) decisions that temporally constrain the actions of the agents according to the human strategy.

II. Literature Review

The scheduling problem for multiprocessor systems is stated as "How can we execute a set of tasks *T* on a set of processors *P* subject to some set of optimizing criteria *C*?-"_-A taxonomy of scheduling algorithms in parallel and distributed systems was presented by [8]. Algorithms have been developed to solve this problem [1, 9].

The algorithm presented by [10] aims to minimize the maximum lateness and total delay for multi-machine problems concerned with the scheduling of single-operation jobs. Several heuristic algorithms were proposed by [11, 12] for the assignment of *n* independent tasks to *m* unrelated parallel machines to minimize the maximum completion time. This approach was extended by this paper to propose a new algorithm for the addressed problem.

To assign spatially distributed tasks to field units (agents) in USAR management, algorithms were suggested by [4, 13]. These algorithms involved geographic information and geospatial analyses in the effective assignment of tasks to agents.

There are several mechanisms, techniques, and algorithms for addressing task assignment to agents (humans, robots, or intelligent systems). They assist agents to achieve their objectives and to maximize the benefits of the system. There are works that address the task-scheduling problem in multi-agent systems [14, 15], multi-robot systems [16], disaster-emergency teams [17], robocup rescue simulations [18], and strategic decision making for coordinating actions of a USAR team[19].

III. Problem Statement

The problem addressed by this paper has characteristics described in the following subsections.

A. Problem Domain

USAR is a significant participant in earthquake disaster-response situations. The overriding goal is to rescue the greatest number of people, trapped under the debris of damaged buildings, in the shortest amount of time. USAR tasks involve *a sequence of dependent tasks*: (1) reconnaissance and assessment by collecting information on the extent of the earthquake damage, (2) search and locate victims trapped in collapsed structures, (3) extract and rescue trapped victims, and (4) transport and dispatch injured survivors to hospitals or shelters. Rescue tasks are classified into light, medium, and heavy rescue.

USAR tasks are location-based entities that are distributed in a geographical area. A specific task needs a specific capability or several synchronous capabilities to complete in an acceptable duration. Capability requirements determine what agents are allowed to do what tasks. There is an uncertainty in task duration, distribution of tasks, and outcome of tasks.

B. Agents

Field units are considered mobile, spatial, rational, heterogeneous, and semi-autonomous agents. Their principal roles are to perform tasks suited to their capabilities. They possess different capabilities that allow them to engage in executing tasks for which they can provide the required capability. Because agents perform their tasks with different speeds, there is an uncertainty in action duration, speed, and outcome.

The IC, as the planning human, attempts to coordinate the disaster-response management. S/he has inaccessible, global, and uncertain information regarding the state of the environment. His observation and perception of a disaster situation is different from that of the agents in their local environment.

C. Macro Tasks

Macro task information forms a global picture of the task environment for ICs. A macro task is the aggregation of all the tasks that 1) are from a same task type and 2) are spatially contained within a specific geographic object or adjacent to a road segment in a definite time. Macro tasks are encoded by road segment in this paper. Road segments are surrounded by buildings that contain USAR tasks. Therefore, a segment contains a number of macro tasks that are located near this geographic object.

A macro task contains two variables: 1) an enabled number of tasks, and 2) a disenabled number of tasks. The enabled variable indicates how many homogeneous tasks are observed, discovered, or revealed within an associated geo-object. Agents can only undertake tasks that are enabled. The disenabled variable indicates how many homogeneous tasks are estimated to be revealed or discovered in the future. They have dynamic and uncertain quantities that may vary over time because agents can complete an enabled task while other agents may reveal (enable)new ones. These numbers provide an estimation of the total duration and total capabilities that are required to complete this macro task.

There is a complex, interdependent relationship among variables of macro tasks that are associated with a geo-object. For example, a whole reconnaissance macro task (enabled number plus disenabled number) can reveal only the disenabled part of a search macro task.

D. Human Strategic Decisions

A high-level strategy specification enables an IC to express his intuition and initiative for solving multi-agent planning problems. A strategy decomposes a complex problem into a set of smaller problems under human supervision. A strategy contains a set of parallel, interdependent, and prioritized threads. A thread is a sub-problem that is composed of a unique ranking, a subset of agents, a subset of task types, and a subset of a geographic area [19].

Strategic decision-making is the problem of the distribution of agents among the threads in an explicit time. Because agents may be shared among threads, an agent should be assigned to only one thread at a time. This agent will be available for the next thread whenever s/he is no longer required and is released by the thread. A human strategic decision defines what agents are assigned to what threads in an explicit time. A thread constrains the actions of the assigned agents according to its specification. A strategic decision should be refined and adapted to new states.

E. Scheduling in Multi-Agent Systems

Scheduling is the problem of the assignment of macro tasks to rational agents under human strategic decisions in multi-agent systems. It does not explicitly specify actions for the agents, however, it does partially identify actions and dynamically constrains agent activities and behaviors in time and space. It allows rational agents to make and adapt their own decisions according to calculated scheduling. Scheduling includes the characteristics as follows:

Scheduling has a macro characteristic. More than one agent can be assigned to a macro task simultaneously. Assigned agents form a coalition that will execute this assignment cooperatively.

Assignments are dynamic. Over time, new agents can be assigned to a macro task to which a set of agents have been previously allocated.

Macro tasks retain assigned agents until the enabled tasks are completed. Released agents can be reassigned to other macro tasks.

The objective is to minimize the overall time for the USAR task execution.

Because of the spatial distribution of tasks, agents need travel time to reach macro tasks moving from one location to another using the road network. In addition, streets have varying states such as blocked or cleaned that affect the shortest paths.

A coalition formed by several professional agents can do a macro task faster than another coalition. An efficient coalition has many capabilities and can finish a macro task faster than another coalition.

Action scheduling is calculated with regard to a human strategic decision as long as this strategic decision is valid. It consists of a sequence of temporal scheduling.

Scheduling changes the state of the macro tasks. An assignment can define how much of a macro task is scheduled to be addressed, or it can define how much of another macro task will be revealed after completion.

IV. Approach

The proposed approach shown in Fig. 1 is used by ICs for multi-agent coordination. The focus of this paper is on the fourth component.

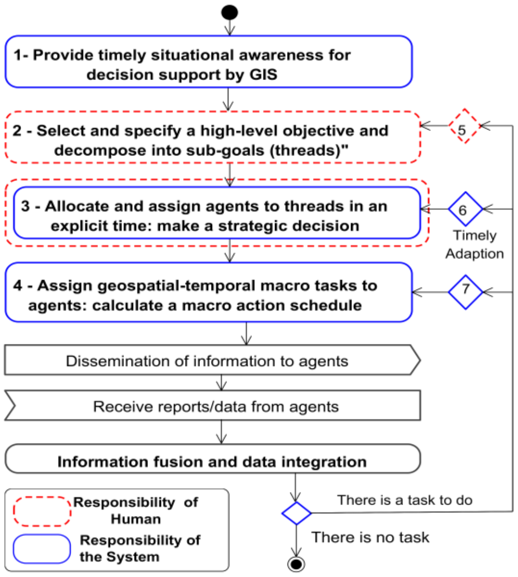

1- Provide timely situational awareness for decision support by GIS
2 - Select and specify a high-level objective and decompose into sub-goals (threads)"
5
3 - Allocate and assign agents to threads in an explicit time: make a strategic decision
6
Timely Adaption
4 - Assign geospatial-temporal macro tasks to agents: calculate a macro action schedule
7
Dissemination of information to agents
Receive reports/data from agents
Information fusion and data integration
Responsibility of Human
Responsibility of the System
There is a task to do
There is no task

Fig. 1. Role of the central scheduling problem (the 4th component) in the proposed approach for strategic planning and scheduling in disaster response team [20].

Whenever a human strategic decision is made for solving a problem, this system autonomously executes a heuristic scheduling algorithm for the dynamic assignment of spatiotemporal macro tasks to rational agents under human strategic decisions. The algorithm, which is shown in Algorithm 1, is composed of 1) select efficient agents, 2) select active macro-tasks, 3) identify the release time, 4) select idle agents, 5) nominate macro task, 6), calculate utilities, 7) find the highest utilities, 8) assign agents to the macro tasks, and 9) calculate the earliest finish time.

```
Data: $startTime$ : a time in which a new strategic decision is made
      by the incident commander. It triggers the scheduling algorithm.
$g2 \leftarrow startTime$;
$L \longleftarrow Create\_emptyset\_of\_legalAssignment()$;
while $true$ do
    $Select\_Efficient\_Agents()$;
    $T \longleftarrow Select\_Active\_MacroTasks()$;
    $A \longleftarrow Select\_Idle\_Agents(g2)$;
    $L \longleftarrow L \cup A$;
    if $Identify\_theRelease\_Time() = true$ then
        return;
    end
    while $true$ do
        if $|L| = 0$ or $|T| = 0$ then
            break;
        end
        $T2 \longleftarrow Nominate\_MacroTasks()$;
        $U \longleftarrow Calculate\_Utilities()$;
        $u \longleftarrow Find\_theHighest\_Utilities()$;
        if $u.benefitRation >= 10$ then
            $Assign\_Agents\_toMacroTasks()$;
            continue;
        end
        else
            $T \longleftarrow T - T2$;
            continue;
        end
    end
    if $|L| = 0$ and $|T| > 0$ then
        $g2 \leftarrow Calculate\_earliestFinishTime()$;
        continue;
    end
    else if $|L| > 0$ and $|T| = 0$ then
        if $Identify\_theRelease\_Time() = true$ then
            return;
        end
        else
            $g2 \leftarrow Update\_ProblemState()$;
            if $g2 = -99$ then
                return;
            end
            else
                continue;
            end
        end
    end
end
```

Algorithm 1. Heuristic algorithm for dynamic assignment of spatiotemporal macro tasks to rational agents under human strategic decisions for centralized scheduling in multi-agent systems

This paper applies the SAP data model [20] for the presentation and formulation of the problem. It enables us to design, implement, and develop the proposed algorithm. Fig. 2 shows a part of this data model that presents the important information classes.

A. Select Efficient Agents

This algorithm aims to select efficient agents for the macro tasks. It has two purposes. The first is to identify what agents can be assigned to what macro tasks. The second is to calculate the maximum ability that an assigned agent can provide to a macro task. The selection of the agent for the macro task is important because selecting an agent states that this agent is a useful candidate who can be legally assigned to and perform this macro task. This algorithm considers four key criteria to efficiently select agents for a definite macro task:

1) Does this macro task contain any task?
2) Does the human strategic decision allow this agent to be allocated to the location of this macro task?
3) Does the human strategic decision allow this agent to be assigned to the task type of this macro task?
4) Does this agent have any useful capability that is required by this macro task and what maximum capability is provide by this agent?

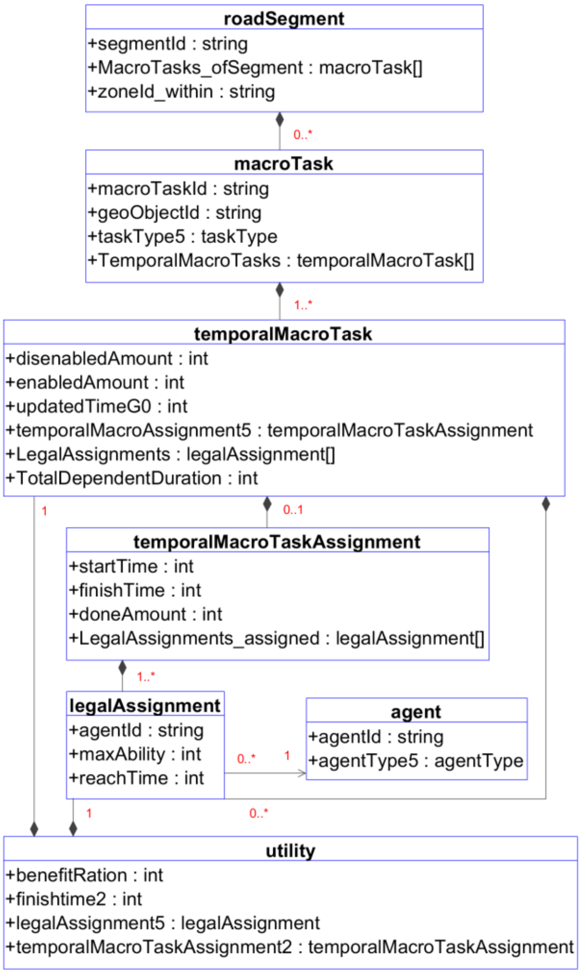


Fig. 2. Part of the SAP data model that includes important information classes for this paper [20]

An agent can be selected for more than one macro task and more than one agent can be selected for a macro task. After agent selection for a definite macro task, this algorithm creates a set of *legalAssignment* elements for this macro task. A macro task may contain a number of *legalAssignment* elements that define what agents are allowed to be assigned to this macro task. A legal assignment is composed of three kinds of information: 1) the Id of the agent, 2) a maximum capability that the associated agent provides with regard to the capability requirements of this macro task, and 3) an estimated time when this agent will arrive in the location of this macro task and will be available to begin.

An agent is efficient for a task if s/he provides any capability required by the task under the human strategic decisions. To simplify this algorithm, we assumed that each task type requires a specific capability and agents provide different capabilities. This assumption ignores tasks that may require simultaneous capabilities. This algorithm must also consider not only the enabled tasks but also the disenabled tasks. These tasks are estimated and expected to be revealed, enabled, or discovered by other tasks in the future. Because of the dynamic environment, this algorithm enables the system to automatically and timely recognize agents who are no longer efficient by macro tasks e.g. there is no task for these agents to carry out for this thread.

B. Select Active Macro Tasks

The aim of this algorithm is to select active macro tasks from the task environment. It is important to identify and recognize only a subset of the macro tasks to which agents can be potentially allocated. Agents can only work on enabled tasks. If agents are assigned to disenabled tasks, they become inactive and must wait until tasks are discovered, revealed, or enabled by completing other tasks.

This algorithm applies two criteria to identify a macro task as active:

1) Does this macro task contain any enabled tasks?
2) Does this macro task contain any *legalAssignment*?

C. Select Idle Agents

This algorithm aims to select idle agents from the team at the current time, *g2*. Selected agents are candidates who will be assigned accordingly to other active macro tasks.

There are two sources that dynamically and temporally provide idle agents for this algorithm:

1) IC who defines and enters new agents to the team.
2) Macro tasks that release assigned agents.

A macro task releases its agents assigned to enabled tasks by the time *g2*. This algorithm accesses the information from the macro tasks and agent allocation and identities all idle agents who have been released by a macro task at this time. A *legalAssignment* element is used to encode a released agent and present a last location (Id of a road segment). Finally, the algorithm adds the selected agents to the current collection, *L*, in the algorithm.

D. Identify the Release Time

This algorithm determines whether it is a correct decision to release a subset of idle agents at time *g2*. It is important, in a timely manner, to adapt a strategic decision to a new situation by releasing unneeded agents from threads and sending (entering) them to other threads that may require them. This algorithm helps the system to select a choice between: 1) continue scheduling with all the idle agents under the same strategic decision, or 2) stop scheduling and let the system revise the current strategic decision.

As Algorithm 2 shows, this algorithm considers two criteria to keep an idle agent for task assignment:

1) Is there any macro task for which this agent has been categorized as an efficient agent previously?
2) Is it estimated that an identified macro task release any new task (enabled tasks or disenabled tasks) at the current time?

E. Nominate Macro Tasks

This heuristic algorithm aims to select a subset of the active macro tasks. Idle agents are assigned to the nominated macro tasks. Unlike a search algorithm, this algorithm selects the best node and extends it to reach the goal node at which all the tasks are completed.

```
Data: $Lf$ :a set of ”legalAssignment” elements that express idle agents.
Data: $S$ :a set of ”segment” elements that contain macro tasks.
Result: $b$ :a boolean variable to show whether it is a right time to
        release a subset of idle agents.
$Ls \longleftarrow Lf$;
for $l \in Lf$ do
    for $s \in S$ do
        for $m \in s.MacroTasks\_ofSegment$ do
            $c \leftarrow |m.TemporalMacroTasks|$;
            $t \longleftarrow m.TemporalMacroTasks[c]$;
            $g \longleftarrow t.temporalMacroAssignment$;
            if $l \in t.LegalAssignments$ then
                $n \leftarrow t.disenabledAmount +$
                $t.enabledAmount - g.doneAmount$;
                if $n > 0$ then
                    $Ls \longleftarrow Ls - \{l\}$;
                end
            end
        end
    end
end
if $|Ls| > 0$ then
    $b \leftarrow true$;
end
else
    $b \leftarrow false$;
end
```

Algorithm 2. “Identify_theRelease_Time()” algorithm that determines whether it is a correct decision to release a subset of idle agents from a thread now

The logic of this algorithm is to select macro tasks that have a strong effect on minimizing the makespan. This algorithm nominates macro tasks that have a “longest total dependent duration”. Two sources of information are important for the calculation of this dynamic variable for each macro task for the current time:

1) How many enabled tasks are contained in this macro task? Alternatively, when does the enabled portion of this macro task complete according to the previous schedule?
2) What is the estimate of how many dependent tasks this macro task can reveal?

F. Calculate Utilities

This algorithm aims to calculate a utility matrix of idle agents for the nominated macro tasks. It is important to answer the question “How much can a macro task benefit from allocation of a specific agent?” This matrix assists in selecting and allocating the best agents for the macro tasks. The benefit statement asks “How much will an agent reduce the finish time of a macro task, if this agent is assigned to the task?” This algorithm takes into account several factors in the calculation of the benefit of a specific agent for a specific macro task:

1) Is it legal for this idle agent to be assigned to this nominated macro task?
2) When will this agent start doing tasks? The dynamic amount of this time is dependent on three factors: 1) moving speed of this agent, 2) release time of this agent, and 3) travel time and distance between two locations in the road network.
3) What capability does this agent provide for this macro task? With what speed can this agent carry out tasks?
4) How much can this agent improve performance of a coalition that has previously been assigned to this macro task?

This utility matrix is used to present a matrix of utility elements that idle agents provide for nominated macro tasks. Rows are assigned to macro tasks and columns are assigned to agents. A set of numbers in a cell shows the utility information that the relevant agent can provide for the relevant macro task.

G. Find the Highest Utilities

The goal of this algorithm is to determine if there is a subset of idle agents who can provide a benefit-ratio higher than 10%. This algorithm uses the utility matrix to find these agents who can provide acceptable benefits for the nominated macro tasks. These agents will be assigned to the related macro tasks. If a suitable agent is not found, all nominated macro tasks will be removed by the algorithm.

H. Assign Agents to Macro Tasks

This heuristic algorithm is shown in Algorithm 3. It aims to select the proper agents and assign them to the macro tasks. This algorithm uses the utility matrix and applies a minimax algorithm to select the best agents for the best macro tasks. This algorithm selects all agents that decrease the finish time of the macro tasks by more than 10%. If an agent is allocated to a macro task, the associated row and column of the matrix is removed and the assigned agent is removed from the list of idle agents.

```
Data: $TU$ : a set of macroTask-Utilities that a member includes a set of
      ”utility” elements for a nominated macro task.
Data: $Lf$ :a set of ”legalAssignment” elements that express idle agents.
while $|TU| > 0$ do
    $MiN \longleftarrow \emptyset$;
    for $tU \in TU$ do
        $U2 \longleftarrow Find\_theHighest\_Utilities\_in\_tU(10)$;
        if $|U2| = 0$ then
            $MiN \longleftarrow MiN + \{0\}$;
        end
        else
            $MiN \longleftarrow$
            $MiN + \{Select\_theMin\_FinishTime\_in\_U2()\}$;
        end
    end
    $m \leftarrow Select\_theMax\_Number\_in\_MiN()$;
    if $m = 0$ then
        break loop;
    end
    $u \longleftarrow Find\_theUtility(m)$;
    $t \longleftarrow Find\_theMacroTask(m)$;
    $a \longleftarrow Find\_theAgent(m)$;
    $n \leftarrow |t.TemporalMacroTasks|$;
    $t[n].temporalMacroAssignment5 \longleftarrow$
    $CreateUpdate\_with\_u()$;
    $Lf \longleftarrow Lf - \{a\}$;
    $TU \longleftarrow TU - \{Find\_tU(t)\}$;
    for $tU \in TU$ do
        $tU \longleftarrow tU - \{u\}$;
    end
end
```

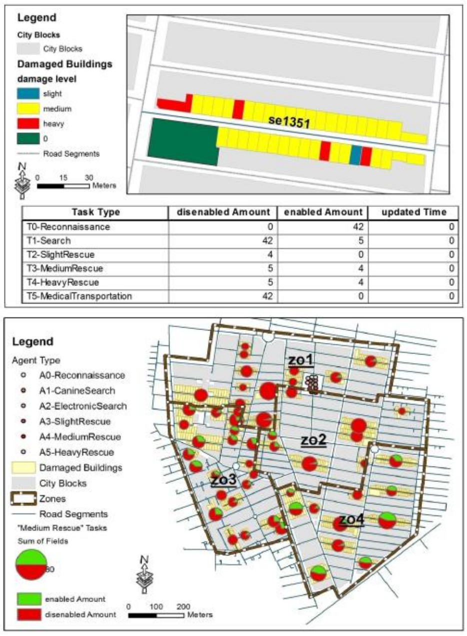
Algorithm 3. Agent selection for assignment of tasks to agents.

I. Calculate the Earliest Finish Time

This algorithm aims to calculate the earliest time at which agents will be released by a macro task and become idle. It updates the problem state that includes agents, tasks, scheduling, and the finish time of each macro task. This algorithm enables us to simulate the problem and estimate future states. A new subset of macro tasks is identified. The results allow other algorithms to continue scheduling.

V. Evaluation and Results

To evaluate the proposed approach, a USAR scenario simulated by GIS was used. As Fig. 3 shows, a rapid response team composed of an IC and 12 agents was assigned to the operational area. The global objective was to accomplish, with this team, USAR tasks in a minimum time. Table 1 shows the strategic decision that was used for scheduling at time 0 (the initial time).



Task Type	disenabled Amount	enabled Amount	updated Time
T0-Reconnaissance	0	42	0
T1-Search	42	5	0
T2-SlightRescue	4	0	0
T3-MediumRescue	5	4	0
T4-HeavyRescue	5	4	0
T5-MedicalTransportation	42	0	0

Fig. 3. Two thematic maps of spatial distribution of macro tasks [20].

TABLE 1. Human High-Level Strategic Decision Valid From Time 0 and Used for Scheduling

thread Id	set of zones	set of task types	set of agents
1	zo1, zo2	T0-Reconnaissance	a0, a1, a2, a3, a4, a5
2	zo1, zo2	T1-Search	a7
3	zo1, zo2	T2-SlightRescue, T3-MediumRescue, T4-HeavyRescue	a9, a11
4	zo3, zo4	T0-Reconnaissance, T1-Search, T2-SlightRescue, T3-MediumRescue, T4-HeavyRescue	a6, a8, a10

The two main results achieved by executing the proposed algorithm are: 1) a feasible schedule, and 2) an adaption/refinement time. In addition, two minor results are achieved: 1) the makespan, and 2) the total schedule.

The schedule is composed of a number of macro decisions that partially specify the actions of the agents. This schedule is calculated according to the valid strategic decision shown in Table 1. A macro decision contains seven types of information:

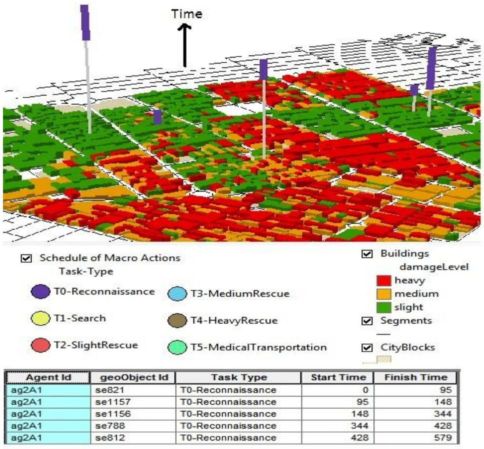
1) A task type as a sub-goal that is scheduled to be completed.
2) A set of agents who are allocated to this decision and are responsible for performing this decision.
3) A geographic object such as a road segment that presents the operational area of this decision.
4) A time that indicates when this decision execution starts.

5) A time that indicates when this decision ends.
6) A quantity of the number of tasks that are estimated to be completed during this decision.
7) A quantity of tasks with different types that are estimated to be revealed at the end of this decision, at the same location.

The adaption time is extracted from the schedule. It estimates a correct time when the current human decision is required to be revised by the IC. Furthermore, this time states that this schedule ends at this time and a new schedule will be calculated based on the strategic decision adapted.

The overall time can be calculated if the scheduling algorithm and the planning algorithm are executed in sequence as shown in Fig. 1. Search algorithms can apply this method to calculate the total schedule with the estimated makespan in order to find an optimal solution for the scheduling problem.

Results showed that 1) the first human strategic decision expired at time 579 (minutes), 2) the total makespan is 16859 (minuts), and 3) a sequence of twelve strategic decisions was made. For example, Fig. 4 shows a geo-visualization of the macro actions scheduled for an agent. An example of the task assignment and action scheduling is presented in Fig. 5 using another data set in order to view a better perception of the results.



Agent Id	geoObject Id	Task Type	Start Time	Finish Time
ag2A1	se821	T0-Reconnaissance	0	95
ag2A1	se1157	T0-Reconnaissance	95	148
ag2A1	se1156	T0-Reconnaissance	148	344
ag2A1	se788	T0-Reconnaissance	344	428
ag2A1	se812	T0-Reconnaissance	428	579

Fig. 4. 3D map that geo-visualizes macro actions scheduled for a specific agent

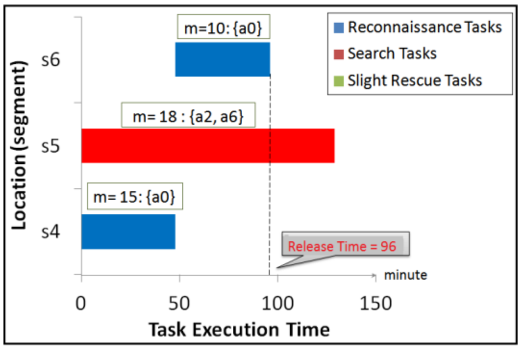


Fig. 5. Example of assignment of macro tasks to agents for another data set

A. Discussion

This algorithm calculated a feasible, fast, and semi-optimal solution for the addressed problem. However, the results showed that this algorithm did not guarantee optimality of the assignments because the task list showed that some agents sometimes became idle. In an efficient coordination, agents should remain active. Minimizing the makespan is dependent on the quality of five parameters: 1) human strategy, 2) strategic decision, 3) search algorithms, 4) scheduling algorithms, and 5) distribution coordination among agents.

The adaption time of a strategic decision is dependent upon the structure of a scheduling algorithm and other factors. If several algorithms are applied to the same data, they will reach different results. We cannot assume that the optimal assignment minimizes the adaption time.

Tasks may include deadlines that influence maximizing the global utility. Humans can manage this issue by specifying an effective high-level strategy.

This paper proposed an algorithm for multi-agent scheduling. There is a value to supporting humans by providing a set of algorithms for the IC and allowing him to select a suitable algorithm for each thread. The system can apply a specific algorithm for a thread.

This paper studied a single IC. A disaster emergency response may involve several ICs and there are interdependencies among the team actions. It would be valuable to study the distributed coordination among multiple ICs and their scheduling.

This paper did not study synchronized actions, facilitating task dependency, or resource allocations in the proposed algorithm. These subjects could be important issues for some domains and teams.

A comprehensive approach for multi-agent coordination includes a loop consisting of scheduling, execution, monitoring, adapting, and learning. Uncertain and dynamic environments disrupt scheduling, making it necessary to execute decisions, continuously monitor execution, and adapt scheduling to unexpected events. Learning algorithms could provide efficient tools to improve the efficiency and performance of this system.

VI. Conclusion

This paper presented a heuristic and greedy algorithm that appropriately assigns macro tasks to rational agents and schedules the macro actions of agents according to human strategic decisions. This algorithm was used to develop GICoordinator that assists the IC in coordinating a team of agents.

A schedule provides an IC with a feasible solution for a coordination problem in a team. Multi-agent scheduling partially specifies the actions of the rational agents and constrains them temporally and dynamically. The IC authorizes agents to autonomously make their own tactical decisions (planning and scheduling) or adapt their activities under these macro decisions.

The proposed system executes this algorithm while the current strategic decision is valid. Therefore, this algorithm calculates a time at which this strategic decision should be adapted. The result has two advantages: 1) It enables the IC to refine this strategic decision at the correct time, and 2) it is used in a search algorithm that estimates a makespan, calculates a complete solution, and finds an optimal solution for the multi-agent planning and scheduling.

Macro tasks enable the system to model and present the distribution of the tasks in different geographic scales. Spatial topology between spatial objects enables the system to extract new views of the tasks. These tasks can be used for task assignment. Limited only by the quality of the data, an IC can select different geographic layers such as zones or buildings and allocate tasks to them. The IC can specify and contain actions in different spatial accuracies.

Information fusion algorithms extract new and useful information from the scheduling results. They provide an efficient situational awareness for an IC to view the state of the global environment. They support human decisions using tools such as geospatial reasoning, geo-visualization, and complex queries. For example, they can answer the questions “What activities will be completed in a specific zone during a specific period?”, “When will a specific task be completed and by whom?”, and “What is the task list of a specific agent?”

Future works to improve the current contribution may include:

1) In some cases, the tasks environment contains tasks that may require synchronous capabilities. This means that the actions of more than one agent must be coordinated to provide the required capabilities for performing a specific task simultaneously. It is important that an ideal algorithm form a proper coalition comprised of suitable agents and assign this group to the task.
2) Decentralized coordination of distributed schedules is a significant issue in which multiple teams are involved in performing the tasks. It is possible that interdependencies between team actions exist. To maximize the joint objective, it is necessary to apply algorithms to coordinate the distributed schedules prepared by each IC.

ACKNOWLEDGMENT

Reza Nourjou is grateful for the financial support of GCOE-HSE of Kyoto University. This enabled him to be a visiting scholar at the Information Sciences Institute of the University of Southern California and the Robotics Institute of Carnegie Mellon University from November 2011 to November 2012.

REFERENCES

[1] Y. K. Kwok and I. Ahmad, "On multiprocessor task scheduling using efficient state space search approaches," *Journal of Parallel and Distributed Computing*, vol. 65, no. 12, pp. 1515-1532, 2005.

[2] Y. K. Kwok and I. Ahmad, "Benchmarking and comparison of the task graph scheduling algorithms," *Journal of Parallel and Distributed Computing*, vol. 59, no. 3, pp. 381-422, 1999.

[3] F. Fiedrich, F. Gehbauer, and U. Rickers, "Optimized resource allocation for emergency response after earthquake disasters," *Safety Science*, vol. 35, no. 1, pp. 41-57, 2000.

[4] R. Nourjou, M. Hatayama, and H. Tatano, "Introduction to spatially distributed intelligent assistant agents for coordination of human-agent teams’ actions," in *Proc. 2011 IEEE International Symposium in Safety, Security and Rescue Robotics*, 2011, pp. 251-258.

[5] W. T. Malone and K. Crowston, “The interdisciplinary study of coordination,” *ACM Computing Surveys*, vol. 26, no. 1, pp. 87-119, 1994.

[6] R. Chen, R. Sharman, H. R. Rao, and S. J. Upadhyaya, "Coordination in emergency response management," *Communications of the ACM*, vol. 51, no. 5, pp. 66-73, 2008.

[7] R. Nourjou, M. Hatayama, S. F. Smith, A. Sadeghi, and P. Szekely, “Design of a gis-based assistant software agent for the incident commander to coordinate emergency response operations,” presented at Workshop on Robots And Sensors Integration in Future Rescue Information System, 2013.

[8] S. J. Chapin, "Distributed and multiprocessor scheduling," *ACM Computing Surveys*, vol. 28, no. 1, pp. 233-235, 1996.

[9] Y. Wen, H. Xu, and J. D. Yang, "A heuristic-based hybrid genetic-variable neighborhood search algorithm for task scheduling in heterogeneous multiprocessor system," *Information Sciences*, vol. 181, issue. 3, pp. 567-581, 2011.

[10] W. A. Horn, "Some simple scheduling algorithms," *Naval Research Logistics Quarterly*, vol. 21, no. 1, pp. 177-185, 1974.

[11] Davis, Ernest and J. M. Jaffe, "Algorithms for scheduling tasks on unrelated processors," *Journal of the ACM (JACM)* vol. 28, no. 4, pp. 721-736, 1981.

[12] C. N. Potts, "Analysis of a linear programming heuristic for scheduling unrelated parallel machines," *Discrete Applied Mathematics*, vol. 10, no. 2, pp. 155-164, 1985.

[13] A. R. Vafaeinezhad, A. A. Alesheikh, M. Hamrah, R. Nourjou and R. Shad, "Using gis to develop an efficient spatio-temporal task allocation algorithm to human groups in an entirely dynamic environment case study: earthquake rescue teams," in *Proc. Computational Science and Its Applications ICCSA*, Springer Berlin Heidelberg, 2009, pp. 66-78.

[14] V. Lesser, K. Decker, T. Wagner *et al*., "Evolution of the GPGP/TAEMS domain-independent coordination framework," *Autonomous Agents And Multi-Agent Systems*, vol. 9, no. 1-2, pp. 87-143, 2004.

[15] E. T. O. Opiyo, E. Ayienga, K. Getao, W. O. Odongo, B. Manderick and A. Nowe, "Game theoretic multi-agent systems scheduler for parallel machines," *International Journal of Computing and ICT Research*, vol. 1, no. 1, 2008.

[16] P. B. Gerkey and J. M. Matari, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939-954, 2004.

[17] P. Scerri, A. Farinelli, S. Okamoto, and M. Tambe, "Allocating tasks in extreme teams," in *Proc. The Fourth International Joint Conference on Autonomous Agents And Multiagent Systems*, ACM, 2005, pp. 727-734.

[18] S. D. Ramchurn, A. Farinelli, K. S. Macarthur, and N. R. Jennings, "Decentralized coordination in robocup rescue," *The Computer Journal*, vol. 53, no. 9, pp. 1447-1461, 2010.

[19] C. A. Chapman, R. A. Micillo, R. Kota, and N. R. Jennings, "Decentralized dynamic task allocation using overlapping potential games," *The Computer Journal*, vol. 53, no. 9, pp. 1462-1477, 2010.

[20] R. Nourjou, P. Szekely, M. Hatayama, M. G. Ashtiany, and S. F. Smith, "Data model of the strategic action planning and scheduling problem in a disaster response team," *Journal of Disaster Research*, 2013.

Reza Nourjou was born in Urmia, Iran in 1979. He received the B.Sc. degree in geomatics engineering and the M.Sc. degree in geographic information systems (GIS) from the K.N.Toosi University of Technology, Iran, in 2002 and 2007, respectively.

In 2006, he joined the Risk Management Research Center, International Institute of Earthquake Engineering and Seismology (IIEES), Tehran, Iran as a GIS expert and a research assistant. Since 2010, he has been a Ph.D. candidate in graduate school of Informatics, Kyoto University, Japan.

His current research interests include GIS, multi-agent systems, decentralized/distributed coordination, automated planning and scheduling, emergency/crisis response systems, software agents, spatial agent-based model, geoinformatics, and intelligent mobile GIS.

He was the recipient of the first prize in the GIS competition ranked first in the national student competition, Iran, in 2001. He received the Japanese Government Scholarship for the Ph.D. program in 2009. He was a visiting scholar at both the Information Sciences Institute of University of Southern California and the Robotics Institute of Carnegie Mellon University from November 2011 to November 2012.