Towards CSP-based mission dispatching in C2/C4I systems

Gerald Steinbauer
Graz University of Technology
Graz, Austria
steinbauer@ist.tugraz.at

Alexander Kleiner
Linköping University
Linköping, Sweden
alexander.kleiner@liu.se

Abstract — One challenging problem in disaster response is to efficiently assign resources such as fire fighters and trucks to local incidents that are spatially distributed on a map. Existing systems for command and control (C2/C4I) are coming with powerful interfaces enabling the manual assignment of resources to the incident commander. However, with increasing number of local incidents over time the performance of manual methods departs arbitrarily from an optimal solution. In this paper we introduce preliminary results of building an interface between existing professional C2/C4I systems and Constraint Satisfaction Problem (CSP)-solvers. We show by using an example the feasibility of scheduling and assigning missions having deadlines and resource constraints.

Keywords: *C2/C4i, mission control, task allocation, CSP*

I. Introduction

In the case of a disaster, whether local or global, the fundamental task of disaster response teams is to allocate available resources to particular incidents at the right time to minimize their negative consequences. Incidents can be very divers in their nature like fires in building, wildfires, earthquakes or floods. Resources can either be *consumable* like extinguishing water, sand bags and food or *renewable* like personal, vehicles and equipment. Obviously, the available amount of consumable resources shrinks with each application. The available resources clearly determine the way disaster response teams are able to cope with an incident. The effectiveness of a response team depends on how well the team is able to allocate its resources to sub-problems, to prioritize sub-problems and to decide the best operational mode for a sub-problem. This problem is far from trivial because of its high number of parameters and its combinatorial complexity. This is even more difficult for large disasters with a large number of individual incidents and resources involved. Even if responder are able to achieve good solutions to the problem, these solutions might be far from the optimal assignment. Looking for an optimal solution makes perfectly sense since it reduced the time (impact) and resources (costs) needed for mitigating a disaster. In most cases responders follow rules of thumb or previously agreed operational rules that usually do not consider actual information on resource distribution, complex cost factors like travel times, operation costs for different types of equipment, and various deadlines. For instance, evacuating a flooded kindergarten obviously has to have an earlier deadline than extinguishing a hayrick.

Research in Artificial Intelligent (AI) has a long tradition in finding optimal and close-to-optimal solutions for planning and scheduling problems involving resources and costs. These types of problems are provably among the hardest problems in AI, and thus, elegant representations and efficient heuristic algorithms have been developed in the past. One recent and very successful method to approach these types of problems is to formalize them as constraint satisfaction problems (CSPs) [2] and to solve them by efficient CSP-solvers [10], [20] that are available.

The aim of the presented work is to emphasize the integration of advanced problem solving techniques into the procedures and tools used by disaster response teams, to evaluate their use in realistically complex and large-scale scenarios, and to compare them to results achieved by response teams using their actual procedures and tools. In order to achieve this goal, a general abstraction of the resource allocation problem existing during real mission planning is needed. On the one hand side, such an abstraction will allow to easily apply and compare different commercially or as open source available automated solving methods to problem instances arising in the disaster response domain. On the other hand side, interfacing real systems for command and control (C2/C4I), as they are already used by first responders, will facilitate the extraction of interesting and more importantly, realistic problem instances in an online or offline fashion.

The contribution of the presented work is threefold. First the paper tackles interesting real-world resource allocation problems that arise from the resource allocation problem in disaster response. Second the paper presents a general problem description that allows easy exchange of problem instances and solutions between researchers working on disaster response or scheduling and planning. Finally, the paper proposes an interface to integrate the ideas into running C2/C4I systems.

The reminder of the paper is organized as follows. In the next section we will discuss related research. In Section III we present a proposal for enhancing existing C2/CI4 system. The section is followed by a general problem definition. Section V shows a concrete formalization of the problem using CSP-techniques. The following section reports preliminary experimental results using example problems. In Section VIII we draw some conclusions and give an outlook on future work.

II. Related Work

There have been several efforts in the past utilizing tools from AI for decision support and mission planning. SAR-Plan (Search And Rescue Planning) is a geographic decision support system designed to assist the Canadian Forces in the optimal planning of search missions. Its primary purpose is to ensure that the available search resources are deployed in a way to maximize the mission's probability of success. The tool provides optimization modules based on search theory, on gradient search methods, and on constraint satisfaction programming [1]. While SARPlan supports the search for both mobile and stationary targets it is mainly designed for search missions at sea and does not take into account complex mission scenarios as they might occur during large scale disaster mitigation.

Tactical decision support through software is available commercially [18], [8] and even with search plan preparation functionality [5], [9]. For USAR, the identification of the search area is straightforward, but the resource demand for individual buildings is difficult to assess and has been addressed [21]. The resource allocation problem during USAR was tackled by [4] and recently, even more fundamentally in [6]. All decision support systems are limited on the one hand, by the availability of pre-incident knowledge such as resources and infrastructure and on the other hand, by the capability of the emergency team to transcribe and disseminate post-incident knowledge such as damages and progress during relief operations. The project I-LOV aims to increase the efficiency of USAR operations on the one hand by ameliorating search technologies and on the other hand assisting the decision-making process [3]. This is based on the centralized representation of all collected information, seamless exploitation of various information sources, information fusion [11], and *Points of Interests* suggestions where further actions should be performed. Methods for task allocation and mission planning where not considered within I-LOV.

Constraint Satisfaction has been widely used to model combinatorial problems in AI. Solvers to these problems are either computing solutions that are satisfying constraints (classical constraint satisfaction), or optimize an objective function by selecting configurations with particular costs or utility (constraint optimization). For both classes of solvers there exists either centralized or distributed methods, i.e., solutions are either computed by a single instance or by multiple instances communicating with each other in order to exchange local results.

However, most constraint satisfaction and optimization problems considered in the past were mainly of static nature, such as the graph coloring problem. Only little attention has been devoted to problems occurring in dynamically changing environments such as the assignment of resources such as responders or trucks to dynamic targets or dynamically changing areas of an environment. Koes and colleagues introduced the *COCoA* architecture for handling rescue mission related coordination by CSP-based solvers. In contrast to the presented work, they did not discuss connectivity to real systems for mission dispatch [12].

In the literature there exists a rich set of both complete and incomplete algorithms for solving Distributed Constraint Optimization Problems (DCOPs). Well known solvers are the Distributed Stochastic Algorithm (DSA) [22], and the distributed versions of arc consistency called distributed soft arc consistency (SAC) [16]. SAC algorithms simplify a DCOP into a soft arc consistent DCOP in a distributed manner. Each agent knows only about the constraints involving its variable and must thus communicate with neighboring agents to exchange information. A classic complete DCOP search algorithm is Asynchronous Distributed Optimization (ADOPT) [17]. ADOPT uses lower and upper solution bounds in a distributed and asynchronous manner for backtracking.

There have been several extensions to the DCOP formulation proposed. Maheswaran et al. introduced a translation from real-world related distributed multi-event scheduling problems into the DCOP formalism [14]. They argued that there indeed exist fundamental differences between the classical static scenarios solved by DCOP algorithms, such as the graph coloring problem, and real-world related dynamic problems, such as sensor networks and meeting scheduling. Therefore, they proposed in [14] several extensions to the ADOPT algorithm enabling them to solve problems with 50 agents.

III. System Overview

The system proposed in this section is based on discussion with fire fighters in Austria and Germany on actual operational rules and their best practices used. Moreover, the system resembles results of an examination of the official C2/C4I system for disaster response used in the province of Styria in Austria. Please note that in Austria and Germany most of the disaster response is done by fire brigades.

The term C2/C4I origins from the military domain and resemble command and control (C2) and command, control, communications, computers, and intelligence (C4I). It deals with making decision, executing decision, allocating and commanding resources, and acquiring and presenting information. All these issues are handled in order to handle a mission. The same mechanism applies also to civil disaster mitigation missions. In recent years professional C2/C4I computer systems have been developed in order to support such missions. Figure 1 shows an example for a C2/C4I system for fire brigades.

Nowadays the allocation of resources to incidents is done in a hierarchy with different levels of abstractions with respect to time, space and resources. The involvement of the different hierarchy levels depends on the severity and the outreach of a disaster. One can roughly distinguish three levels: (1) incident commander and platoon (operational), (2) command center (strategical) and (3) staff command (strategical).

In case of moderate disasters such as local fires and floods the command center, e.g. the local fire or police station, receives the emergency calls directly and thus dispatches local resources, e.g., by assigning incident commanders and platoons of responders to different incidents. Each platoon posses a set of resources needed to mitigate an incident. The composition of the platoon may change from incident to incident and is guided by operational rules. These rules are predefined and map a particular emergency type, e.g., car accident or fire, to a particular platoon composition. Once an incident commander is assigned, he begins to mitigate its incident by allocating the platoon's resources to sub-tasks in the right order and at the right time. For instance, the incident commander may decide to use some resources to setup a water supply chain first and to use the remaining resources for extinguishing the fire. If a disaster becomes really serious

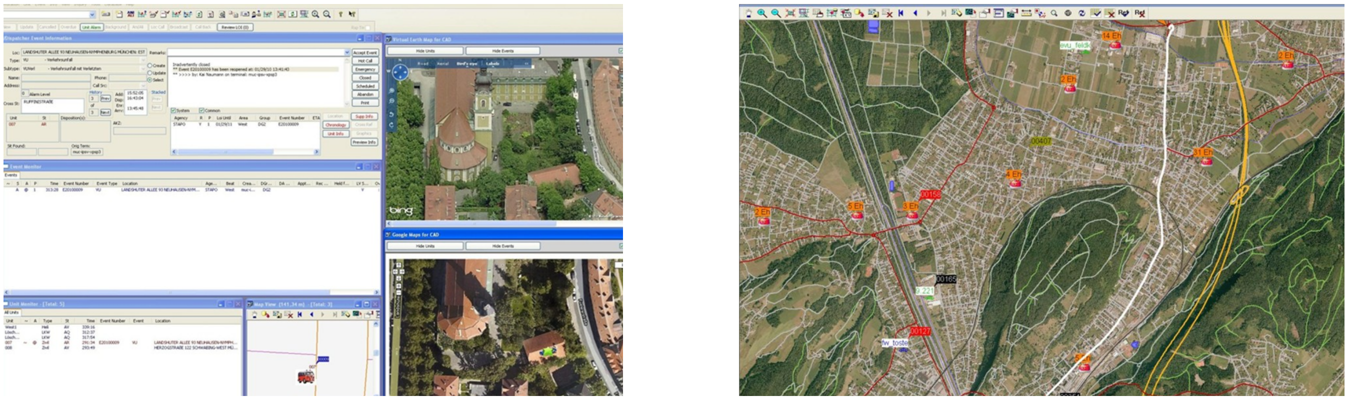

Fig. 1. Intergraph Computer Aided Dispatch (I/CAD) as example for a C2/C4I system. It shows a configuration for two monitors, additional integration of Bing and Google Maps (left) and a standard map (right). It is able to provide resource proposal. According to available information like vehicle equipment and type of incident it proposes how to dispatch forces in the most efficient way. (Courtesy of Intergraph SG&I Germany)

the staff command level becomes active. This level is able to reassign resources on a more global and abstract level and to activate additional resources like special units or the military. Moreover, it is also responsible for the cooperation with administration, politicians and the press.

Figure 2 shows an overview of the different modules active in the disaster mitigation process. On the command center level existing C2/C4I system like the example above are already used in the field. These systems ease and accelerate the task of allocating resources to incidents by providing context information, e.g., location of the caller, a blueprint of a building, and information on the available resources such as fire trucks in repair or maintenance. Usually the proposed resource allocation follows predefined rules stored in a database. These rules have a certain degree of granularity and do not consider task-related information. For instance, there exists a predefined platoon arrangement comprising particular vehicles that is deployed for building on fire.

Typically resources are *greedily* assigned to targets with respect to their spacial distance, i.e., the closer resources are located to an incident the more likely they are dispatched for it. It can easily be shown that this greedy strategy turns out to be suboptimal, particularly when several incidents occur at the same time. Our long-term goal is to increase system performance and flexibility by introducing an abstract interface for decision support. Through this interface highly optimized constraint solvers can propose resource allocation schemes, for example, based on task-related information such as the actual location of resources and the current situation of road traffic. In the subsequent sections we provide a formal problem definition of the underlying resource allocation problem, and a preliminary solution strategy based on CSP-techniques.

IV. Problem Formulation

In this section we present a general formulation that captures the problem of assigning a team of heterogenous rescue agents for disaster mitigation during an incident. The incident is composed of several locations of interest at which specific activities can be carried out by the agents. Each activity has the intention to support the disaster mitigation of an incident. Possible activities can be, for example, to extinguish

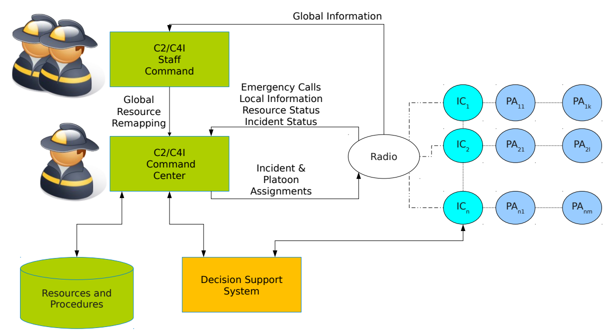

Fig. 2. System overview of the enhanced C2/C4I system showing incident commanders (ICs) and platoon agents (PAs).

burning buildings, to pump water from flooded buildings, but also to refill water tanks of fire trucks. Each activity can require specific equipment, such as hoses and ladders, and each member of a team can have individual capabilities too.

Formally, we denote the team of N heterogeneous agents by $\mathcal{R} = \{R_1, R_2, \dots, R_N\}$ and the set of M activities by $\mathcal{A} = \{A_1, A_2, \dots, A_M\}$. The heterogeneity of agents is expressed by the capability sets $\mathcal{C}_r$ with $r \in \{1 \dots N\}$, one attached to each agent, and the specific requirements of activities are expressed by requirement sets $\mathcal{Q}_i$ with $i \in \{1 \dots M\}$, one attached to each activity.

The individual C_r and $\mathcal{Q}_i$ can be seen as preconditions. An agent r is able to work on activity i if its capabilites match the activitie's requirements.

Typically, activities have to be carried out at different locations that have to be reached by the agents in advance. We denote the location where activity A_i can be carried out by L_i. Notice that we clearly distinguish here between the activities extinguish building A and extinguish building B in case $L_A \neq L_B$. Due to practical reasons we do not intend to model the path planning problem as part of our problem formulation. For path and motion planning there exist many efficient methods such as *Random Rapid Trees (RRTs)* [13] and *A* heuristic search* [19]. Thus, we assume the existence of a pre-computed distance matrix $\{d_{ij}\}$ denoting the real distance between locations L_i and L_j, and $ttime_{ij}$ as the constant travel time needed to travel between activities A_i and A_j assuming that agents travel at same speeds. This matrix can efficiently be computed in $O(N^3)$ by applying the Floyd-Warshal algorithm [7], where N is the number of activities.

An activity A_i may have a deadline T_i^{max} before it has to be completed.

V. Constraint Satisfaction Problem Formulation

Formally, a constraint satisfaction problem (CSP) is defined as a triple $\langle X, D, C\rangle$, where X is a set of variables, D is a domain of values, and C is a set of constraints. Every constraint is in turn a pair $\langle t, R\rangle$ (usually represented as a matrix), where t is a n-tuple of variables and R is a n-ary relation on D. An evaluation of the variables is a function from the set of variables to the domain of values, $v : X \to D$. An evaluation v satisfies a constraint $\langle (x_1, \ldots, x_n), R\rangle$ if $(v(x_1), \ldots, v(x_n)) \in R$. A solution is an evaluation that satisfies all constraints. Moreover, an objective function $O : X \to \mathbb{R}$ can be defined.

In order to formulize the general problem we need the following variables and constraints:

A. Variables

- We denote the assignment of agent n to activity m by the binary variable R_nA_m, i.e., R_nA_m is 1 if agent n is assigned to activity m and 0 otherwise. In order to model the fact that an agent has to travel to its first assigned activity form its initial position for each agent i an initial activity A_i is generated. Therefore, the set of activities $\mathcal{A}$ has $N + M$ members. The location of initial activities are set to the initial position of the related agent.
- The integer variable T_m_s denotes the absolute time when the work on activity A_m is started.
- The integer variable $R_nA_m\_s$ denotes the absolute time when the work on activity A_m by agent R_n is started. Notice that by this agents can join the work on a task at arbitrary times. $R_nA_m\_s$ is zero if the binary variable R_nA_m is zero.
- The integer variable $R_nA_m\_d$ denotes the amount of time (duration) agent R_n works on activity A_m. $R_nA_m\_d$ is zero if the binary variable R_nA_m is zero. Per default the duration of initial activities are zero as well.

B. Constraints

$$\forall_{t\in\{1\ldots N+M\}} \sum_{r\in\{1..N\}} R_rA_t = 1$$

Each activity A_m is assigned to exactly one agent R_n. A agent may perform more than one activity, but not in parallel.

$$\forall_{t\in\{1,\ldots,N\},r\in\{1,\ldots,N\}} R_rA_t = \begin{cases} 1 & \text{if } r = t \\ 0 & otherwise \end{cases}$$

$$\forall_{t\in\{1,\ldots,N\}} \forall_{r\in\{1,\ldots,N\}} R_rA_t\_s = 0$$

$$\forall_{t\in\{1,\ldots,N\}} \forall_{r\in\{1,\ldots,N\}} R_rA_t\_d = 0$$

The initial activities are assigned to their related agents. The start time and duration of the initial activities are zero.

$$\forall_{r\in\{1\ldots N\}} \sum_{t\in\{1..N+M\}} R_rA_t \cdot Q_t \le QR_r$$

Each activity A_t needs the resource Q_t to be executed. Each agent r cannot invest more than its own resource QR_r for all task allocated to it.

$$\forall_{t\in\{1,\ldots,N\},r\in\{1,\ldots,N\}} R_rA_t\_s \ge 0$$

All start times of activities have to be greater than zero. No activity can be started in the past.

$$\forall_{t\in\{1,\ldots,N\},r\in\{1,\ldots,N\}} R_rA_t\_d = d_t \cdot R_rA_t$$

The time needed to complete an activity t by an assigned agent r is given by d_t. The time is zero for unassigned agents.

$$\begin{gathered} \forall_{r\in\{1\ldots N\}} \forall_{i,j\in\{1..N+M\},i<j} \\ (R_rA_i\_s + R_rA_i\_d + d_{i,j}) \cdot a_{ij} \le R_rA_j\_s \cdot a_{ij} \lor \\ (R_rA_j\_s + R_rA_j\_d + d_{i,j}) \cdot a_{ij} \le R_rA_i\_s \cdot a_{ij} \\ \text{with } a_{ij} = R_rA_i \cdot R_rA_j \end{gathered}$$

This ensures that no two activities A_i and A_j assigned to an agent R_r overlap in time and that the agent needs at least the travel time between both activities $d_{i,j}$ in between.

$$\forall_{t\in\{1,\ldots,N+M\}} T_t\_s = \sum_{r\in\{1,\ldots,N\}} R_rA_t\_s \cdot R_rA_t$$

$$\forall_{t\in\{1,\ldots,N+M\}} T_t\_d = \sum_{r\in\{1,\ldots,N\}} R_rA_t\_d \cdot R_rA_t$$

The start time and duration of a activity A_t is determined by the time and duration of the agent assigned to it.

$$\sum_{t\in\{1,\ldots,N+M\}} \sum_{r\in\{1,\ldots,N\}} R_rA_t \cdot U_t \cdot \frac{T_t^{max}}{T_t^{max} + T_t\_s}$$

The objective function for the problem is the sum of the individual utilities of the activities discounted by a factor for later start. Please note that unassigned activities do not contribute to the function. In order mitigate the problems faster this function has to be maximized.

VI. Homogenous Agent Example

We consider the problem of assigning fire fighters for extinguishing fires as shown in Figure 3. In this problem the time for extinguishing a burning building but also the danger that fires spread to neighboring buildings increases the longer the fire exists. Hence, fire agents need to be assigned to fires as early as possible since at early stages both risks for fire spread and time needed for extinguishing are low. In our example we assume that each agent can reduce the amount of water needed for extinguishing a burning building during each discrete time step t by one unit. The amount of water W_b needed to extinguish a burning building b depends on the size and the number of storeys of the building.

Furthermore, there exists a deadline T_b^{max} after that the building will be destroyed by the fire, and a time dependent utility $U_b(t)$ for extinguishing the building. Since we assume here that every agent can extinguish the buildings in the same way (homogeneous agent team), we can reduce the capability sets $\mathcal{C}_i$ to a single integer representing the amount of water per agent. As an example we consider the case of four

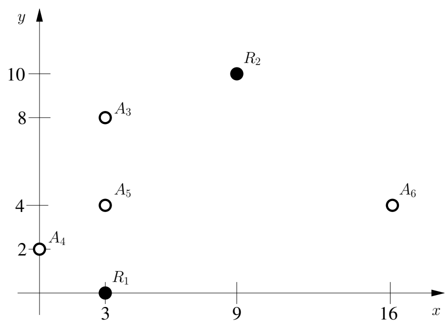

Fig. 3. Simple extinguishing example with 4 activities/fires (A_3, A_4, A_5, A_6) and two firefighting agents (R_1,R_2). The agents are located initially at position (3,0) and (9,10).

simultaneously burning buildings and two firefighting agents shown in Figure 3. The parameters of this problem are defined as follows:

- *Bldg. 1:* $L_1 = (3, 8)$, $W_1 = 100$, $T_1^{max} = 120$, $U_1 = 5$
- *Bldg. 2:* $L_2 = (0, 2)$, $W_2 = 10$, $T_2^{max} = 100$, $U_2 = 2$
- *Bldg. 3:* $L_3 = (3, 4)$, $W_3 = 5$, $T_3^{max} = 10$, $U_3 = 1$
- *Bldg. 4:* $L_4 = (16, 4)$, $W_4 = 20$, $T_4^{max} = 100$, $U_4 = 5$
- *Agent 1:* $I_1 = (3, 0)$, $QR_1 = 140$
- *Agent 2:* $I_2 = (9, 10)$, $QR_2 = 140$

Resulting in the activities:

- $A_1 = (0, (3, 0), 0, 0)$ ... initial activity R_1
- $A_2 = (0, (9, 10), 0, 0)$ ... initial activity R_2
- $A_3 = (100, (3, 8), 120, 100)$ ... building 1
- $A_4 = (10, (0, 2), 10, 10)$ ... building 2
- $A_5 = (5, (3, 4), 10, 5)$ ... building 3
- $A_6 = (20, (16, 4), 100, 20)$ ... building 4

The distances between tasks are represented by their Euclidean distance, e.g. $d_{3,5} = 4$. Using the CSP-based formulation of Section V, the *MiniZinc* modeling language [15] and the open-source constraint satisfaction solver *Gecode* [20] we were able to obtain the following optimal assignment:

- A_1 assigned to R_1 with $A_1\_s = 0$ and $A_1\_d = 0$
- A_2 assigned to R_2 with $A_2\_s = 0$ and $A_2\_d = 0$
- A_3 assigned to R_2 with $A_3\_s = 6$ and $A_3\_d = 100$
- A_4 assigned to R_1 with $A_4\_s = 50$ and $A_4\_d = 10$
- A_5 assigned to R_2 with $A_5\_s = 110$ and $A_5\_d = 5$
- A_6 assigned to R_1 with $A_6\_s = 14$ and $A_6\_d = 20$

Please note that the start times respect also the travel time between the activities. For instance, the start time of activity A_3 T_3_s is 6 since agent R_2 needs 6 time units to reach its first activity. Notice that the deadline for T_5 is violated as the deadlines are only modeled as soft-constraint within the objective function. Because of the low individual utility of $U_5 = 1$ a higher overall utility is achieved even if T_5 misses the deadline. Apparently, if this solution is undesirable for some kind of domain, one can also model deadlines as hard-constraints.

VII. Preliminary Results

In order to systematically evaluate the performance of the proposed CSP-based problem formulation we generated artificial problem instances with varying parameters. We generated three problem instances for each combination of the number of agents (1..7) and fires (1..10). Within the individual problem instances we set the remaining parameter randomly from a uniform distribution:

- initial location of agents: randomly on a 100 × 100 grid
- location of fires: randomly on a 100 × 100 grid
- time for extinguishing a building: randomly from the interval [0,10]
- deadline for a burning building: randomly from the interval [0,200]
- agent resources: randomly from the interval [10,20]

We set the minimum amount of resources to 10 in order to avoid useless agents with zero resources. Moreover, we assigned an uniform utility of 1 to all activities. Figure 4 shows the average computation time for the CSP-based assignment problem. Please note that the results for 1 and 2 agents are omitted. There is no feasible solution for the problem in most cases because of the limited number of agents and resources. For the other problem cases the CSP-solver is able to find an optimal solution with respect to the given objective function and constraints. The experiments were conducted using the *Gecode* CSP-solver [20] on a computer equipped with a quad-core Intel Xenon CPU running at 3.2 GHz and 6 GB memory. The used solver is single-threaded and therefore not able to benefit from multiple cores.

These preliminary results show that for a small number of agents (up to 5) and fires (up to 7) the computation of an optimal assignment can be calculated in a few seconds. This is already a promising result because considering travel distances between fires as well as resource constraints it is already impossible for a human operator to find the optimal assignment. For larger assignment problems the computation time is currently infeasible (several hours) using the first simple CSP-based problem formulation.

VIII. Conclusion and Future Work

The success of disaster mitigation depends on the responder's capability to allocate resources to activities efficiently. Modern C2/C4I system support responder in the resource allocation task. Usually, this task follows predefined operation rules that do not take into account task-related information such as travel efforts and synergetic effects. In this paper we gave a general formulation of this problem and show a first solution based on CSP-techniques.

Our preliminary results show that automated task allocation is feasible for smaller problem instances in the order of ten targets to be handled at the same time. This work can be considered as a starting point for the development of more efficient solvers that then can be integrated into real existing C2/C4I systems.

In future work we will aim at a decentralized version of the assignment formalization in order to allow to solve

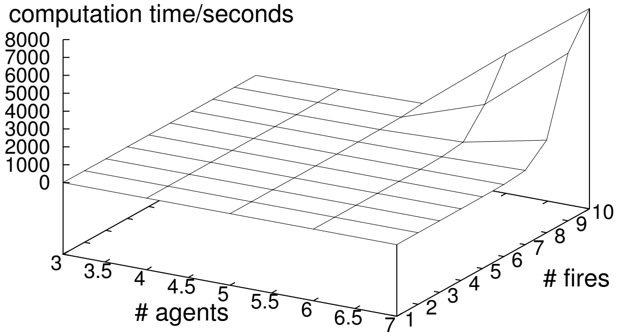

Fig. 4. Average computation time for assignment problems with different numbers of agent and fires.

Figure 5 shows the average computation time and its standard deviation for a fixed number of 5 agents. It clearly shows the exponential grow of time to find the optimal solution. Moreover, it shows a large variance for the computation times. For different problem instances with 5 agents and 10 fires the computation time varies between 7 and 464 seconds. This indicates that even if the number of agents and fires are fixed there is wide variety in the difficulty to find an optimal solution for the different problem instances.

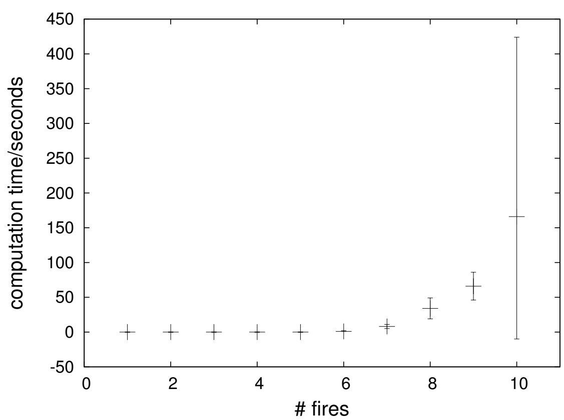

Fig. 5. Average computation time for assignment problems for a fixed number of 5 agents.

assignments with a minimum communication overhead as it is typically required for operational task assignment in the field.

Acknowledgment

The authors would like to thank Heimo Krajnz from the Professional Fire Brigade Graz for discussions on their operation rules and the C2/C4I systems used. Moreover, we like to thank Anette Breu and Jens Hartmann from Intergraph SG&I Germany for the access to a running installation of their I/CAD C2/C4I system. This work was partially supported through the Swedish Foundation for Strategic Research and the Excellence Center at Linköping and Lund in Information Technology (ELLIIT).

References

[1] Irène Abi-Zeid and John R. Frost. SARPlan: A decision support system for Canadian Search and Rescue Operations. *European Journal of Operational Research*, 162(3):630 – 653, 2005. Decision-Aid to Improve Organisational Performance.

[2] Krzysztof Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.

[3] T. Becker, B.-S. Lee, Q. Hamp, and M. Eitelberg. Efficient Decision Support for Crisis Management Based on Information Fusion and Modern SOP Algorithms. Technical report, Gesellschaft für Informatik, Leipzig, Germany, 2010.

[4] JG Blitch and R Maurer. KNOBSAR: A knowledge based system prototype for robot assisted urban search and rescue. *SIMULATION*, 66(6):375–391, JUN 1996.

[5] BMT Marine Information Systems Limited. SEARCH AND RESCUE INFORMATION SYSTEM (SARIS). Technical report, BMT, 2010.

[6] Brian Cheung, Samuel Davey, and Douglas Gray. Comparison of the PMHT Path Planning Algorithm with the Genetic Algorithm for Multiple Platforms. In *Information Fusion*, Edinburgh, UK, 2010. IET.

[7] T.H. Cormen. *Introduction to algorithms*. The MIT press, 2001.

[8] DHI-WASY. GeoFES. `http://www.dhi-wasy.de`, October 2010.

[9] EMS. SARMASTER. `http://www.emsglobaltracking.com`, October 2010.

[10] Ian P. Gent, Chris Jefferson, and Ian Miguel. Minion: A fast scalable constraint solver. In *European Conference on Artificial Intelligence (ECAI)*, pages 98–102, Riva del Garda, Italy, 2006. IOS Press.

[11] Quirin Hamp, Michael Eitelberg, Bo-Sik Lee, Tobias Becker, Dirk Wiebeck, and Leonhard Reindl. Information Fusion based on graph analysis during Urban Search and Rescue. In *Information Fusion*, Edinburgh, UK, 2010. IET.

[12] M. Koes, I. Nourbakhsh, and K. Sycara. Constraint optimization coordination architecture for search and rescue robotics. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3977–3982. IEEE, 2006.

[13] S.M. LaValle and J.J. Kuffner Jr. Rapidly-exploring random trees: Progress and prospects. 2000.

[14] R.T. Maheswaran, M. Tambe, E. Bowring, J.P. Pearce, and P. Varakantham. Taking DCOP to the real world: Efficient complete solutions for distributed multi-event scheduling. In *Proc. of the 3rd Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, pages 310–317. IEEE Computer Society, 2004.

[15] Kim Marriott, Nicholas Nethercote, Reza Rafeh, Peter J. Stuckey, Maria Garcia De La Banda, and Mark Wallace. The design of the zinc modelling language. *Constraints*, 13(3):229–267, September 2008.

[16] T. Matsui, M.C. Silaghi, K. Hirayama, M. Yokoo, and H. Matsuo. Directed soft arc consistency in pseudo trees. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1065–1072. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[17] P.J. Modi, W.M. Shen, M. Tambe, and M. Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1):149–180, 2005.

[18] Pro DV. DeNIS II^{plus}. `http://www.prodv.de/`, October 2010.

[19] S. Russell, P. Norvig, and A. Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 1995.

[20] Christian Schulte and Guido Tack. Implementing efficient propagation control. In *TRICS 2010, Third workshop on techniques for implementing constraint programming systems*, 2010.

[21] Christine Schweier and Michael Markus. ASSESSMENT OF THE SEARCH AND RESCUE DEMAND FOR INDIVIDUAL BUILDINGS. In *13th World Conference on Earthquake Engineering*, Vancouver, B.C., Canada, August 2004.

[22] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Distributed stochastic search and distributed breakout: properties, comparison and applications to constraint optimization problems in sensor networks. *Artificial Intelligence*, 161(1):55–87, 2005.