Discrete Uncertainty Representation for CSP-based Planning and Scheduling and Application to Control Command Systems

Philippe Morignot
ASPertise
pmorignot@aspertise.net

Christophe Guettier
Safran Electronics & Defense
christophe.guettier@safrangroup.com

Abstract

Intelligent and Autonomous Unmanned Ground Vehicles (AUGV), both for civil security and warfare domains, are subject of growing interest from the Intelligent Transportation Systems community and from the Planning & Scheduling (P&S) one. This paper presents an approach to cope with discrete uncertainty representations of P&S problems. The model enables statement of coordination constraints within multiple agents. This representation is based on a discrete confidence interval, denoting bounds around an exact (certain) value provided at planning time. Search algorithms are also proposed, solving P&S problems of realistic size. An implementation, inside the CSP-based P&S system known as ORTAC, demonstrates that the computation time, due to this additional uncertainty representation, is not significantly degraded.

Introduction

Since their introduction in the early 90s, intelligent and autonomous vehicles aim at progressively replacing manually-driven vehicles by computer-driven automated ones, in order to prevent traffic accidents and injuries/fatalities. In general, Intelligent Transportation Systems (ITS) *cooperate* to handle free space and more generally road access.

But in many applications such as search and rescue, natural disaster response, or defense and security missions, several vehicles, manned or unmanned, have to *collaborate* to achieve a common goal. For those applications, Autonomous Unmanned Ground Vehicles (AUGV) are of a particular interest for several dangerous or fastidious missions.

AUGV involve several software modules such as simultaneous localization and mapping (SLAM), perception, data fusion, path planning and then control of the robotic platform. Each functionality has to deal with some form of temporal and spatial uncertainty while representing the environment. In our work, we consider multiple agents (with manned or unmanned vehicles) which traverse a topological map and which must eventually coordinate their respective actions via a control/command (C2) system. While manned coordination with voice communications can be very efficient within a trained team of first responders, interacting with AUGV can be a challenge, especially considering temporal uncertainty.

The paper focuses on the Planning and Scheduling (P&S) software modules and, more specifically, on the temporal uncertainty resulting from the environment or from the various processing stages. These modules will be integrated in C2 systems that coordinate the different vehicles and enable interactions between human and AUGV. The P&S environment is named ORTAC, standing for Optimal Resource and Technical Action Control, and has been developped for both defense and civil security domains, even if other applications are investigated.

The problem also involves technical actions (e.g., observations, measurements, communications) to perform on some waypoints and must consider specific metrics such as security, travelling distances and durations. That is, given a graph where vertices are locations and edges are routes, the P&S problem is to find for all agents a sequence of vertices (or route segments) with pass-by dates on waypoints, optimizing a mission cost. Once the plan defined and communicated, AUGVs must automatically manage their own trajectory and follow their navigation waypoints using control algorithms and time sequence. Drivers of manned platform follow a path of timed waypoints provided by a C2 interface.

A classical candidate approach for this problem is, for example, the A* algorithm (Hart, Nilsson, and Raphael 1968) considered as a best-first search in a space of paths. Even if A* can handle several metrics, including timing, it assumes that there exists only one agent which traverses edges and vertices to reach a final location. Our approach considers constraint programming, that has been identified since the 70s (Montanari 1974) (Laurière 1978) as a powerful tool to represent and solve combinatorial problems, or Constraint Satisfaction Problems (CSP). Its real-world applications are numerous, refer to (Simonin et al. 2015) to only mention a spectacular one.

In both C2 systems and ITS, path planning for multiple agents in a topological map can be modeled and solved using constraint programming (Guettier 2007). A CSP is composed of a set of variables, their domains and algebraic constraints (together compsing a *model*), which are based on abstracting some problem. However, due to many sources of uncertainty, the passing date of an agent on a given vertex/location in the topological map might not be precisely known at planning time: Some form of uncertainty has to be considered, in order to represent a lack of knowledge at planning time.

In this paper, we propose a new representation of uncertainty, based on confidence intervals, within a CSP model representing path planning in a topological map for multiple agents with coordination. The paper is organized as follows: The first section describes the system environment and the application; The second section presents the basic CSP model, representing path planning for multiple agents in a topological map, and then a new uncertainty model; The third section provides the implemented search strategies, and experimental results are reported in a fourth section; The last section relates our work to existing approaches, sums up our contribution and gives hints for further work.

Application domain

For each agent, the P&S module must find navigation plans and estimate passing dates, while satisfying coordination constraints with other agents. Each mission plan is composed of a set of totally ordered waypoints, for which a passing date must also be estimated. Agents plans and schedules must meet an objective and obey to terrain constraints. Global coordination between agents can be inforced by satisfying logical synchronisations on waypoints. In addition, the global mission plan should optimize a primary cost function, for instance mission duration, safety, security or observability. Without loss of generality, only mission duration is considered in this paper, that is, minimizing the maximal mission completion date for all agents.

Example

In Figure 1, both rescue AUGV and manned vehicules must perform a maximal exploration of villages (red circle) in a flooded area, looking for refugees and estimating damages. However vehicles must progress in a synchronized way for several operational reasons:

- Observations must be synchronized to avoid missing refugees;
- Operators in manned vehicles would want to see AUGV time to time in order to be able to switch to a teleoperation mode if needed;
- Communications between vehicles have to be maintained during mission progression.

In this scenario, search and rescue vehicles will start from location 1, and gather in the vicinity of node 20. All nodes circled in red have to be visited, where refugees and casualties are likely to be found. However, there are strong uncertainties concerning the time of traversal: On one hand, the manned vehicule need to master the AUGV execution in spite of uncertainty; On the other hand, the AUGV must adapt to the manned vehicule pace.

Architecture

Figure (2) presents a simplified C2 system architecture for an AUGV, with autonomous (robotic) capabilities. Building-up the situation awareness is based on various sensory data,

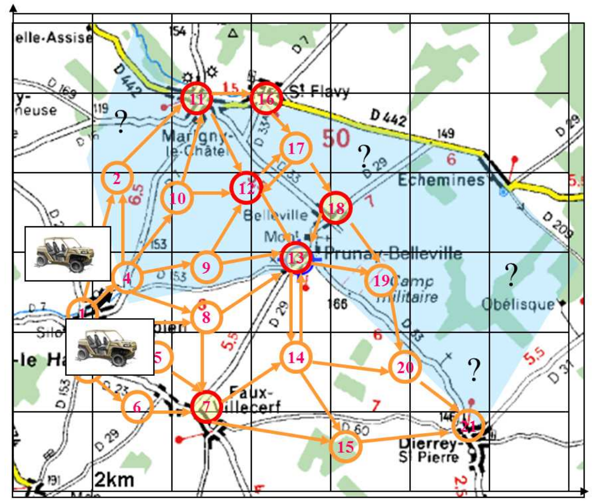


Figure 1: Search and rescue mission. Topological map for a manned vehicle and a AUGV throughout a flooded area between the Seine and the Vanne rivers in the area of Troyes, France. These specific vehicules, developped by SAFRAN E&D, can be either piloted or turned instantaneously into an AUGV.

local map and SLAM processing. These subsystems generate multiple sources of temporal and spatial uncertainty: drift, errors and bias. Moreover, time of traversal for some parts of the terrain is difficult to predict and is uncertain while planning the mission (in the example, moving in shallow water). Vehicular communications enable exchanges of mission plan as well as situation awareness (platform state such as position, velocity, time and list of observed objects). The vehicule can optionally interact with an operator or the pilot, but interactions definition are out of the scope of this paper. In addition to the vehicular P&S, we need a temporal uncertainy resolver module for several reasons:

- provide to the command staff go/no go decisions on whether the mission will continue;
- dynamically adapt the mission or trigger a replanning event;
- provide timing worst / nominal cases to the pilot or to the local operator;
- provide delays and arrival date estimates to other manned vehicules or AUGVs;
- adapt the execution controler to cope with potential delays and to maintain coordination with other vehicles.

A larger description of the tool is described in (Guettier 2007) and has been widely experimented. Example of detailed C2 system integration is described in (Guettier et al. 2011), while some fielded experiments are reported in (Guettier et al. 2009) (Guettier et al. 2015). The search algorithm baseline for a single agent is presented in (Guettier and Lucas 2016).

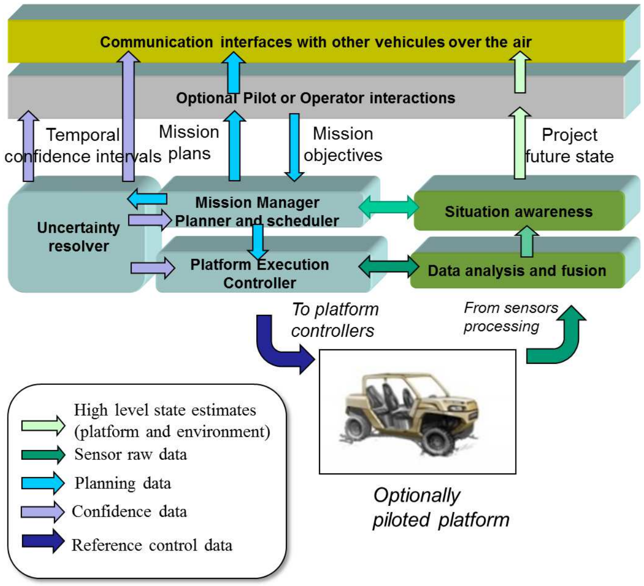


Figure 2: C2 system for optionally piloted vehicule (which can be piloted, remotely piloted, remotely operated or fully autonomous), with main components including P&S, communication, interactions and situation awareness. In spite of many sophisticated filtering techniques used to improve accuracy (location, observations) and reduce uncertainty, situation awareness is till an active area of research.

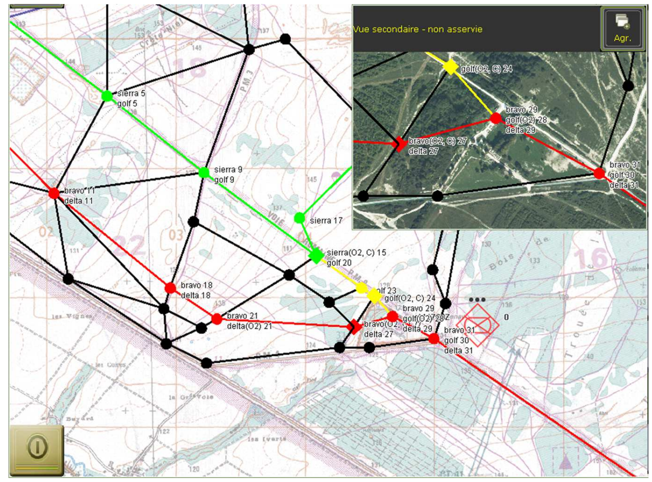


Figure 3: Operational vehicular C2 system integrating the ORTAC planner and displaying solutions. Agents paths are displayed in different colours to reach the central target. Plans are georeferenced on both satellite views and maps. The C2 systems is interfaced with inertial measurement units in order to provide locations with minimal uncertainty, even in GPS denied environments.

Model-based P&S with Uncertainty

In this section, we present a CSP-based P&S system and describe an extension to represent uncertainty.

The ORTAC P&S System

The system known as ORTAC is a model for a constraint satisfaction problem to compute paths of several agents using a directed graph, where vertices represent locations and edges represent routes (referred to as a topological map). With the given applications, graphs are defined during mission preparation, by terrain analysis and situation assessment. A path of an agent starts at a fixed initial vertex (starting location) and ends at a fixed final vertex (ending location), and is composed of a sequence of routes (i.e., chain of edges). The graph is maintained on-line during mission execution, by using fusion of sensory data (e.g., LIDAR, optronics). The P&S system also models duration and waypoint timing sequences, according to selected paths. Both manned and AUGV systems respond to an operator's (or mission commander's) demand by finding a route from a starting point to a destination, while visiting some mandatory waypoints.

In our approach, solving the P&S problem is achieved using Constraint Programming (CP) techniques, under a model-based development approach. CP is a competitive approach to solve such problems, providing completeness and optimality guarantees. With CP, a declarative formulation of the constraints to satisfy is provided which is decoupled from the search algorithms, so that both of them can be worked out independently. Both CSP formulation and search algorithms are implemented with the CLP(FD) domain of SICStus Prolog library (Carlsson 2015). It uses the state-of-the-art in discrete constrained optimisation techniques and Arc Consistency-5 (such as AC-5) for constraint propagation, managed by CLP(FD) predicates, as well as global constraints implementation.

Since more than one agent can be represented in ORTAC, the model also represents coordination among agents at given vertices of their respective paths. This is performed by expressing constraints relating two different agents on two different vertices: For instance, an agent must pass at a location *before* another agent passes at another location. Forced inclusion/exclusion of vertices/edges in a path of an agent can also be represented by additional constraints.

For each agent, the basic model relies on a graph where edges and nodes represent respectively ground mobility and accessible waypoints:

- a $\{0,1\}$ variable T_v on each vertex v, representing the fact that this vertex is included into agent path, i.e., the agent transits via vertex v;
- a $\{0,1\}$ variable $\varphi_{v,v'}$ on each edge v, v' representing the fact that the edge also belongs to the path, i.e., the agent transits from vertex v to vertex v';
- a flow constraint stating that an agent arriving at a vertex departs from it (with specific cases for the start and end vertices).

Constraint-based model for P&S More formally, a graph is a pair (V, U) where V is a set of vertices (or nodes) and U is a set of edges. Variables $\varphi_u \in \{0, 1\}$ represent a possible path from $start \in V$ to $end \in V$, where an edge $u \in U$ belongs to the navigation plan iff $\varphi_u = 1$ (and 0 otherwise).

The resulting navigation plan Φ can be represented as $\Phi = \{u |\ u \in U,\ \varphi_u = 1\}$.

Path consistency from an initial position to a final one is enforced by flow conservation equations, where $\omega^+(v) \subset U$ (resp., $\omega^-(v) \subset U$) represents the outgoing (resp., incoming) edges from (resp., to) vertex $v \in V$.

$$\sum_{u\ \in\ \omega^+(start)} \varphi_u \ = \ 1, \quad \sum_{u\ \in\ \omega^-(end)} \varphi_u \ = \ 1, \tag{1}$$

$$T_v = \sum_{u\ \in\ \omega^+(v)} \varphi_u = \sum_{u\ \in\ \omega^-(v)} \varphi_u \leq 1 \tag{2}$$

Since flow variables φ_u are $\{0, 1\}$, equation (2) ensures path connectivity and uniqueness, while equation (1) imposes limit conditions for starting and ending the path. This constraint produces a linear chain of pass-by waypoints in a graph — waypoints are vertices of a topological map which are present in a path of a navigation plan Φ.

These waypoints v are labeled by passing time D_v depending on variables $V_{(v,v')}$ denoting the average velocity on edge (v, v') — this variable is within realistic ranges, depending on the physical minimum and maximum speeds of the robotic AUGV. The value $\|v, v'\|$ is the constant distance between two vertices v and v'. Variable $d_{(v,v')}$ represents the duration of traversal of an edge (v, v'), therefore the last 3 variables are related by equation (3) — the remaining variable $r_{v',v}$ is ignored, since it represents non-integer values of variables $V_{(v,v')}$ and $d_{(v,v')}$.

Given equation (3), passing times on waypoints are propagated via equation (4), which cumulates edge traversal duration along waypoints.

$$\|v, v'\| = V_{(v,v')}.d_{(v,v')} + r_{v,v'} \tag{3}$$

$$D_v = \sum_{(v',v)\ \in\ \omega^-(v)} \varphi_{(v',v)}(d_{(v',v)} + D_{v'}) \tag{4}$$

Resource consumption, security and observability can also be modelled by constraints.

Representing Uncertainty

Regarding agents transiting on a topological map, uncertainty can be temporal (uncertainty on the time at which an agent arrives to a vertex) or spatial (an agent can be located with latitude/longitude coordinates, not necessarily those of any vertex). This paper focuses on temporal uncertainty among locations, planned by agents.

For this, confidence intervals, a well-known representation of temporal uncertainty over the previous (certainty) variables, are introduced: A *confidence interval* of a finite-domain variable X in the above CSP model is an interval over integers $[X^{min}; X^{max}]$, represented by two finite-domain variables X^{min} and X^{max} associated to X. A *realization* of X is an instanciation of X which conforms to its confidence interval.

For example, a realization of the above duration variables $d_{(v,v')}$ on edges is an instanciation of all variables $d_(v, v')$ along the path, which conform to their confidence intervals $[d^{min}_{(v,v')}; d^{max}_{(v,v')}]$.

Velocity Model for Uncertainty Uncertainty is represented in the previous velocity model (recall equation (3) in the previous paragraph) by turning both the velocity and the duration variables into confidence intervals $[V^{min}_{(v,v')}; V^{max}_{(v,v')}]$ (refer to equation (5)) and $[d^{min}_{(v,v')}; d^{max}_{(v,v')}]$ (refer to equation (10)), and smilarly relating their lower bounds (refer to equation (6)) and upper bounds (refer to equation (7)) — as for the previous certainty model, the rounding variables $r_{v,v'}$ are ignored, since they correspond to non-integer values of confidence intervales on velocities, distances and durations.

A constant relative confidence interval $[\Delta V^{min}_{(v,v')}; \Delta V^{max}_{(v,v')}]$, where $\Delta V^{min}_{(v,v')}$ is a negative integer and $\Delta V^{max}_{(v,v')}$ is a positive integer, limits the possible expansion of the confidence interval for velocity $[V^{max}_{(v,v')}; V^{max}_{(v,v')}]$ (refer to equations (8) and (9)).

$$V^{min}_{(v,v')} \leq V_{(v,v')} \leq V^{max}_{(v,v')} \tag{5}$$

$$\|v, v'\| = V^{min}_{(v,v')}.d^{min}_{(v,v')} + r^{min}_{v,v'} \tag{6}$$

$$\|v, v'\| = V^{max}_{(v,v')}.d^{max}_{(v,v')} + r^{max}_{v,v'} \tag{7}$$

$$V_{(v,v')} + \Delta V^{max}_{(v,v')} \leq V^{max}_{(v,v')} \tag{8}$$

$$V_{(v,v')} + \Delta V^{min}_{(v,v')} \geq V^{min}_{(v,v')} \tag{9}$$

$$d^{min}_{(v,v')} \leq d_{(v,v')} \leq d^{max}_{(v,v')} \tag{10}$$

The ORTAC model also represents the duration S_v of an action performed by an agent at a waypoint v, in addition to its passing time D_v — agents not only passes at waypoints but perform durative actions there. Since duration and velocity on an edge are uncertain, representing uncertainty also involves turning variable D_v into a confidence interval $[D^{min}_v; D^{max}_v]$ on each vertex v. The realization of variable D_v in its confidence interval is represented by equation (11).

Representing uncertainty on vertices relates the arrival time D_v at waypoint v to the time of arrival $D_{succ(v)}$ at the next waypoint $succ(v)$ of the same path, using the confidence interval bounds (12 and 13) — uncertainty over confidence interval never decreases along path following, hence the direction of the two inequalities.

$$D^{min}_v \leq D_v \leq D^{max}_v \tag{11}$$

$$D^{min}_{succ(c)}) \geq D^{min}_v + d^{min}_{(v,succ(v))} + S_v \tag{12}$$

$$D^{max}_v + d^{max}_{(v,succ(v))} + S_v \leq D^{max}_{succ(v)} \tag{13}$$

Uncertain Coordination among Agents The ORTAC model can represent not only one agent traversing a topological map, but several agents: this is performed by iteratively defining the finite-domain variables representing each agent behavior and constraint postings, all indexed by each agent — a loop over variable definitions and constraint postings, indexed by agent (that is, two different missions on the same scenario and the same agents, but with different coordinations, imply slightly different CSP models). As such, it seems natural to represent coordination among these agents, using additional constraints. For example, in a disaster recovery scenario involving a damaged village (as in Figure

1), one or more agents look for refugees in the surrounding outside while another agent searches for casualties inside.

For this, ORTAC represents temporal coordination constraints between two agents on two vertices, for which the semantics is informally defined as follows (refer to (Guettier 2007) for the logical semantic definition):

- *before*: agent A performs its action on its vertex before agent B performs its action on its other vertex, within a time window;
- *after*: agent A performs its action on its vertex after agent B performs its action on its other vertex iff agent B before agent A;
- *simultaneous*: agents A and B perform their respective actions on their respective vertices during the same period of time;
- *disjunct* : agent A is disjunct from agent B on vertex v iff agent A is passing before agent B or agent B is passing before agent A on vertex v.

In order to represent temporal uncertainty into these coordination constraints, the confidence intervals of the previous section must also be considered into the above coordination formulation. Let A and B be two different agents transiting in a topological map:

- Agent A is *uncertainly simultaneous* to agent B iff their respective confidence intervals exactly overlap (see equation (14)).
- Agent A is *uncertainly disjunct* from agent B iff the upper bound of the confidence interval of agent A is less than the lower bound of the confidence interval of agent B (including the duration of the action performed by agent A on the vertex), or the opposite by switching A and B (see equation (15)).

Formally, given two confidence intervals $[D_v^{min}(A); D_v^{max}(A)]$ denoting the passing time of agent A at waypoint v, and $[D_{v'}^{min}(B); D_{v'}^{max}(B)]$ similarly for agent B at waypoint v', the following uncertain coordination formulations can be written:

$$D_v^{min}(A) = D_{v'}^{min}(B) \wedge D_v^{max}(A) = D_{v'}^{max}(B) \tag{14}$$

$$\begin{aligned} D_v^{max}(A) + S_v(A) \leq D_{v'}^{min}(B) \vee \\ D_{v'}^{max}(B) + S_{v'}(B) \leq D_v^{min}(A) \end{aligned} \tag{15}$$

$$D_v^{min}(A) \leq D_{v'}^{min}(B) \wedge D_v^{max}(A) \leq D_{v'}^{max}(B) \tag{16}$$

$$D_v^{max}(A) + S_v(A) \leq D_{v'}^{min}(B) \tag{17}$$

Since temporal intervals are represented in addition to time points, the certain *before* coordination constraint is turned into *weak* (see equation (16)) or *strong* (see equation (17)) *uncertainly before* coordination constraints, depending on the existential (see equation (18)) or universal (see equation (19)) quantifier used for the realization of the variables $D_v(A)$ and $D_{v'}(B)$ for the synchronization between agents A and B. The difference between the weak and strong uncertainly before constraints can also be considered as enabling/forbidding overlaps between the two confidence intervals, which can be formalized with temporal intervals (see relations "before" and "overlaps" (Allen 1983)).

$$\begin{aligned} & weak\_synchronisation\_before(A, B) \Leftrightarrow \\ & \forall v \in V, \exists (D_v(A), D_{v'}(B)) \\ & \in [D_v^{min}(A); D_v^{max}(A)] \times [D_{v'}^{min}(B); D_{v'}^{max}(B)] \\ & s.t. D_v(A) \leq D_{v'}(B) \end{aligned} \tag{18}$$

$$\begin{aligned} & strong\_synchronisation\_before(A, B) \Leftrightarrow \\ & \forall v \in V, \forall (D_v(A), D_{v'}(A)) \\ & \in [D_v^{min}(A); D_v^{max}(A)] \times [D_{v'}^{min}(B); D_{v'}^{max}(B)] \\ & s.t. D_v(A) \leq D_{v'}(B) \end{aligned} \tag{19}$$

Finaly, the same formal model and constraints are defined for the *after* uncertain coordination constraint, by switching agents A and B in the previous *before* uncertain coordination model.

Compound search algorithms

Two search algorithms are considered, one to solve the initial coordinated P&S problems, and then one to solve the temporal uncertainty resulting from the coordinated paths.

Solving the P&S problem with ORTAC

The global search technique under consideration guarantees completeness, solution optimality and proof of optimality. It relies on three main algorithmic components:

- Variable filtering with correct values, using specific labeling predicates to instantiate problem domain variables. the constraint propagator being incomplete, value filtering guarantees the search completeness.
- Tree search with standard backtracking when variable instantiation fails.
- Branch and Bound (B&B) for cost optimisation, using minimise predicate.

Designing a good search technique consists in finding the right variables ordering and value filtering, accelerated by domain or generic heuristics. A static probes provides an initial variable selection ordering, computed before running the global branch and bound search (Guettier and Lucas 2016). In the approach, the variable selection order provided by the probe can still be iteratively updated by the labeling strategy that makes use of other variable selection heuristics. In general, dynamic probing techniques use solutions to some relaxations of the original problem and consider these 'partial' solutions as tentative values, see for example (Sakkout and Wallace 2000) and (Ruml 2001). In ORTAC, the search strategy uses a static prober which orders problem variables before the search. This ordering is based on the relations between problem structure and the partial solution found. Then, the solving relies a standard CP branch and bound search strategy, combining variable filtering, AC-5, generic heuristic and B&B. The probing technique proceeds in three steps:

- Establish the relaxed problem, abstracting away mandatory waypoints and coordination constraints.

- Compute a shortest reference path between starting and ending vertices, using Dijkstra or A*.
- Establish a minimal distance between any problem variable and the solution to the relaxed problem.

The last step considers the following distance between partial solution values X_s and all original problem variables X.:

$$\forall x \in X, \delta(x) = \min_{x' \in X_s} ||(x, x')|| \tag{20}$$

where $||.||$ is the distance metric, corresponding to the number of vertices between x and x'. The last step uses the resulting partial order to sort problem variables in ascending order, using $\delta(x)$. Problem variables are explored following that order in the global search. The probe construction is polynomial and does not change completeness nor optimality properties of the global branch and bound loop.

Search with Uncertainty

As explained above, confidence intervals are intervals over integers representing temporal uncertainty at planning time around an exact (certain) planned integer value. That is, involving confidence interval $[D_v^{min}(X), D_v^{max}(X)]$ of passing time $D_v(X)$ of agent X on vertex v, in which $D_v^{min}(X) \leq D_v(X) \leq D_v^{max}(X)$.

We follow this definition by using a labeling search on uncertainty after the labeling search on the exact (certain) value of the passing time $D_v(X)$ of agent X at each vertex v. Hence, paths and passing times on vertices are known (i.e., $D_v(X)$ finite-domain variables are instantiated) before search on uncertainty is performed.

In order to increase performances, a static heuristic on variables is used: If a path of length n is composed of waypoints $v_1, v_2 \dots, v_i, \dots v_n$, this heuristic reorders variables $D_{v_i}^{min}(X)$ and $D_{v_i}^{max}(X)$ according to the path from start to end in the forward direction, by increasing i. This heuristic on variables considers the uncertainty variables $D_{v_i}^{min}(X)$ and $D_{v_i}^{max}(X)$ for agent X in the following order (21):

$$\begin{aligned} D_{v_1}^{min}(X) \prec D_{v_1}^{max}(X) \prec D_{v_2}^{min}(X) \prec D_{v_2}^{max}(X) \\ \dots D_{v_i}^{min}(X) \prec D_{v_i}^{max}(X) \dots \prec D_{v_n}^{min}(X), D_{v_n}^{max}(X)) \end{aligned} \tag{21}$$

Since the confidence interval $[D_{v_1}^{min}(X); D_{v_1}^{max}(X)]$ at the starting location v_1 of agent X is known, labeling is sufficient to instantiate these confidence intervals along the path. If a coordination constraint creates an empty domain of any finite-domain variable $D_{v_i}^{min}(X)$ or $D_{v_i}^{max}(X)]$ for any agent X on any vertex v_i, CP backtracking occurs inside the search on uncertainty (i.e., on confidence intervals) and then possibly inside the search on certainty (i.e., on exact variables) — finding other confidence intervals for the same realization of $D_{v_i}(X)$, or finding another realization.

Experimental results

Benchmarks

Experiments on four benchmarks are presented, which are representative of peace keeping missions or disaster relief.

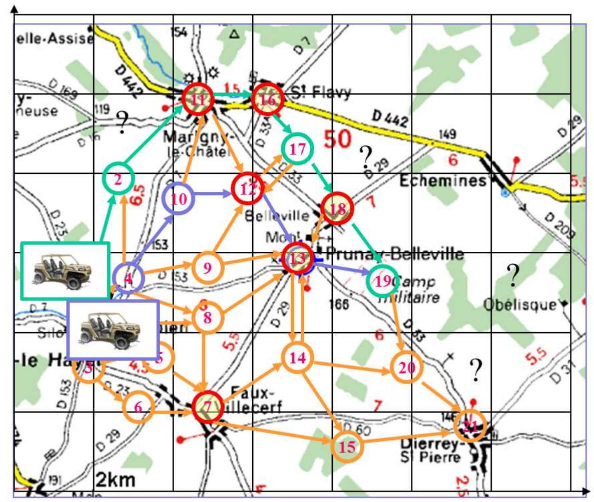


Figure 4: Paths solution for coordinated manned vehicle and the AUGV

Missions must be executed in less than 30 minutes. Areas range from 5x5 kms to 20x20 kms.

1. Recon villages: Observing different villages after a major water flooding event, described as a running example in Fig. (1) and for which a solution to a 2 agents problem is given in Figure (4);
2. Reinforce UN: Bring support to a United Nations mission by deploying observers in an unsecure town;
3. Sites inspections: Observing different parts of a town during inspection of suspect sites;
4. Secure humanitarian area: Observing different threats before securing refugees, over a large area.

On the first benchmark, Figure 4 shows the two paths found by the first P&S algorithm. Resolving uncertainty then provides the confidence intervals given in Figure 5 for the two coordinated agents.

Performances of the Uncertainty Model

In order to measure the additional computational cost of the solving process due to the uncertainty model, ORTAC has been run on 4 topological maps, composed of 22 / 33 / 23 / 22 vertices and, respectively, 74 / 113 / 76 / 68 edges. Each example involves 2 to 8 agents. The experiments were carried out on a computer with processor i7 at 2GHz with 4Gb RAM on a virtual machine. The computation time is measured for the certainty search and for the uncertainty one — see Figure 6. Further experiments have been carried out with a topological map representing the streets and intersections of Paris: solving time takes more than 2 hours under the same experimental conditions.

Related work and Discussion

First, Nilsson et al. (Nilsson, Kvarnström, and Doherty 2015) define Simple Temporal Networks with Uncertainty (STNUs) as an extension of Simple Temporal Networks

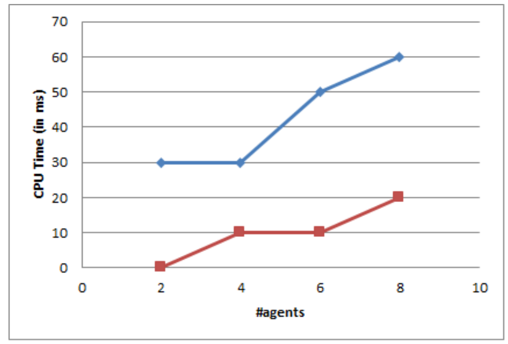
```
- Uncertain coordination simultaneous between:
unit1 on node 11 and unit2 on node 12
--- Agent : unit1
 Absolute uncertainty on node 2 : -2 =< 0 =< 3
 Absolute uncertainty on node 11 : 10 =< 32 =< 35
 Absolute uncertainty on node 16 : 55 =< 77 =< 80
 Absolute uncertainty on node 17 : 59 =< 81 =< 84
 Absolute uncertainty on node 18 : 65 =< 87 =< 90
 Absolute uncertainty on node 19 : 71 =< 93 =< 96

--- Agent : unit2
 Absolute uncertainty on node 1 : -2 =< 0 =< 3
 Absolute uncertainty on node 4 : 2 =< 4 =< 7
 Absolute uncertainty on node 10 : 5 =< 7 =< 10
 Absolute uncertainty on node 12 : 10 =< 12 =< 35
 Absolute uncertainty on node 13 : 16 =< 18 =< 41
 Absolute uncertainty on node 19 : 82 =< 84 =< 107
 Absolute uncertainty on node 20 : 88 =< 90 =< 113
```

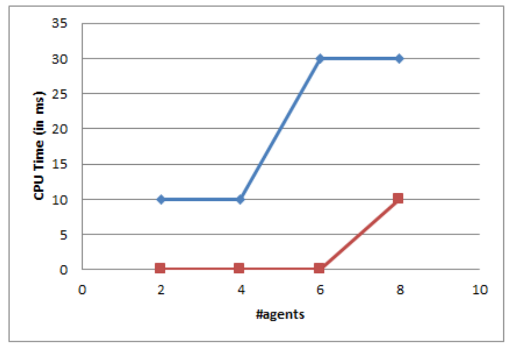
Figure 5: Excerpt of output of ORTAC for 2 units ”unit1” and ”unit2” with the coordination constraint ”simultaneous” between vertices 11 for unit1 and 12 for unit2. Each line shows the lower bound of the confidence interval $D_v^{min}(X)$, the exact (certain) passing time $D_v(X)$ on each vertex v, and the upper bound of the same confidence interval $D_v^{max}(X)$. Times are given in minutes and progression time in search and rescue is expressed in meters per minute.

(STNs) (Dechter, Meiri, and Pearl 1991) towards representing uncertainty — this has been extended towards continuous uncertainty with Probabilistic STNUs (Santana et al. 2016). A temporal action in a STNU is represented as start and end times, with a bounded duration: for every temporal action A, $duration(A) = end(A) - start(A) \in [min(A), max(A)]$. These authors propose an algorithm with $O(n^3)$ complexity to incrementally verify that there always exists a solution for the start and end times of each action (dynamic controllability), regardless of what happens at execution time — these start and end times are constrained by uncontrollable/contingent phenomena (e.g., wind, weather). In contrast, our approach does not consider one agent only, as with STNUs, but several, which is modelled by a flow constraint (refer to equation (2)). As such, our model can represent coordination constraints among agents (crucial for our application on AUGVs), which cannot be represented by STNUs' binary constraints. A common ground between STNUs and our approach would be to define a CP global constraint, called dynamic controllability verification, to ensure consistency of a subset of our CP constraints model.

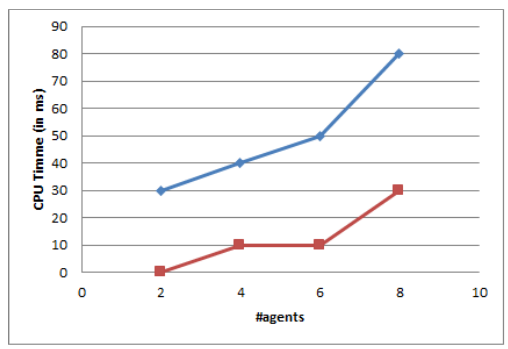
Second, Fargier et al. (Fargier, Lang, and Schiex 1996) extend the CSP framework to deal with reasoning under incomplete knowlege: they propose an anytime algorithm (implemented in (Guettier and Yorke-Smith 2005) for an application in the aerospace domain) based on a set X of uncontrollable variables and on another set Y of controllable variables — hence its name *mixed*-CSP. The algorithm proposed by these authors covers realizations of variables of X, one by one, with CSP resolution over variables of Y and iterates on realizations until they are all covered. This algorithm exhibits an anytime property, since uncontrollable variables are considered first one by one: interrupting this algorithm leaves covered a subset of X. In contrast, our approach is based on uncertainty by extending a certainty reasoning, as



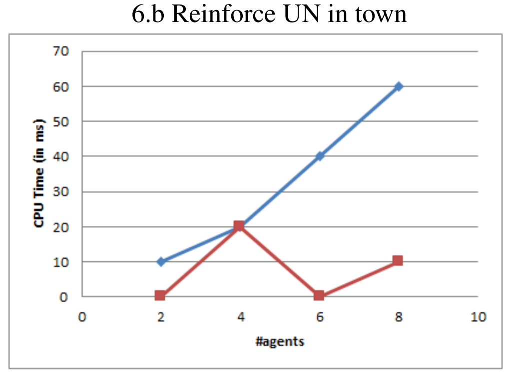
6.a Recon village after flooding



6.c Suspect sites inspection



6.b Reinforce UN in town



6.d Secure humanitarian area

Figure 6: Performance on benchmarks according to the number of agents and one coordination constraint per run: execution time in blue and red, respectively for the reference P&S problem, and the scheduling under uncertainty

STNU extends STN, whereas mixed-CSP considers uncontrollability first and then responds to it by controllability — an approach which suffers from severe algorithmic complexity.

Third, one could argue that mixed-integer programming (MIP), instead of CP, could be used to solve our model. That is, equation (2) would be interpreted as an integrity equation, common in MIP, whereas the rest of the model would be turned into linear inequalities among variables on integer or real values. Unfortunately, our velocity model is not linear but quadratic (refer to equation (3)).

However, following this idea anyway, our model is based on finite-domain variables (i.e., on variables over integers), as in every CSP, and it would be interesting to mix integers and real numbers, as in MIP. For example, for representing continuous values of temporal variables in our model, such as passing time D_v at waypoint v or duration S_v. Indeed, the implementation language, Sicstus Prolog, includes a continuous solver (Carlsson 2015), but that latter solver and the CSP solver hardly cooperate. A more interesting approach towards mixing discreteness and continuity in CP is the CSP solver CHOCO (Prud'homme, Fages, and Lorca 2017), harmoniously integrated to the continuous solver IBEX (Chabert and Jaulin 2009). But porting ORTAC onto these two solvers would entail large software engineering work.

Finally, the incremental property of STNU's verification algorithm and the anytime property of mixed-CSP are interesting, which would lead in our context to what could be called *anytime CSP*, meaning interrupting a CSP solver before completion and having a partial solution where some quality would increase over the alloted time. But that would be another story — after all, time that passes can also be considered as an uncontrollable continuous variable.

Conclusion

A discrete representation of temporal uncertainty based on confidence intervals in a CSP-based planning and scheduling system has been presented. This extends a system known as ORTAC (Guettier 2007) which finds paths in a topological map for multiple agents with coordination constraints — its applications include planning paths of tactical units in a wargame, finding routes in a road network while minimizing consumed energy and planning medical visits of patients. Early experiments show that adding an uncertainty model to a certainty one does not significantly degrade the solving performances of the whole system.

Future work includes: Considering a higher level language inspired by ANML (Smith, Franck, and Cushing 2008), which seems more appropriate than PDDL (McDermott et al. 1998) for P&S robotic applications (Dvorak et al. 2014); And connecting ORTAC to a wargame simulating AUGVs, before porting the system to AUGVs for real.

Acknowledgments

The authors thank Jean-Francois Tilman (SAFRAN E&D) for numerous fruitful discussions and anonymous reviewers for helpful suggestions. This work has been sponsored by contract CAMPUS for ADEME.

References

Allen, J. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 832–843.

Carlsson, M. 2015. *SICSTUS Prolog users manual*.

Chabert, G., and Jaulin, L. 2009. Contractor programming. *Artificial Intelligence* 173:1079–1100.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.

Dvorak, F.; Bit-Monnot, A.; Ingrand, F.; and Ghallab, M. 2014. Plan space hierarchical planning with the action notation modelling language. In *ICTAI*.

Fargier, H.; Lang, J.; and Schiex, T. 1996. Mixed constraint satisfaction: a framework for decision problems under incomplete knowledge. In *AAAI*. AAAI Press.

Guettier, C., and Lucas, F. 2016. A constraint-based approach for planning unmanned aerial vehicle activities. *The Knowledge Engineering Review* 31(5):486497.

Guettier, C., and Yorke-Smith, N. 2005. Enhancing the anytime behavior of mixed csp-based planning. In *ICAPS Workshop on planning under uncertainty for autonomous systems*.

Guettier, C.; Sechaud, P.; Yelloz, J.; Allard, G.; Lefebvre, I.; Peteuil, P.; Ponthereau, P.; Cuisinier, F.; and Martinet, J. 2009. Improving tactical capabilities with net-centric systems: the phoenix'08 experimentation. In *Proceedings of IEEE MILCOM Military Communications Conference*.

Guettier, C.; Yelloz, J.; Cherrier, O.; Mayk, I.; and Lamal, W. 2011. Interoperable joint planning and execution web service with titan. In *MILCOM 2011 Military Communications Conference*, 20252030.

Guettier, C.; Lamal, W.; Mayk, I.; and Yelloz, J. 2015. Design and experiment of a collaborative planning service for netcentric international brigade command. In *IAAI*.

Guettier, C. 2007. Solving planning and scheduling problems in network based operations. In *Proceedings of Constraint Programming (CP)*.

Hart, P.; Nilsson, N.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science and Cybernetics* 4(2):100–107.

Laurière, J.-L. 1978. A language and a program for stating and solving combinatorial problems. *Artificial Intelligence* 10:29–127.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ashwin, R.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. Pddl — the planning domain definition language. Technical report, Yale Center for Computational Vision and Control, New Haven, CT. Technical Report CVC TR98003/DCS TR1165.

Montanari, U. 1974. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences* 7:95–132.

Nilsson, M.; Kvarnström, J.; and Doherty, P. 2015. Efficient processing of simple temporal networks with uncertainty: Algorithms for dynamic controllability verification. *Acta Informatica*.

Prud'homme, C.; Fages, J.-G.; and Lorca, X. 2017. *Choco Documentation*. TASC - LS2N CNRS UMR 6241, COSLING S.A.S.

Ruml, W. 2001. Incomplete tree search using adaptive probing. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, volume 1, 235–241. Morgan Kaufmann Publishers Inc.

Sakkout, H. E., and Wallace, M. 2000. Probe backtrack search for minimal perturbations in dynamic scheduling. *Constraints Journal* 5(4):359–388.

Santana, P.; Vaquero, T.; Toledo, C.; Wang, A.; Fang, C.; and Williams, B. 2016. Paris: a polynomial-time, risk-sensitive scheduling algorithm for probabilistic simple temporal networks with uncertainty. In *ICAPS*.

Simonin, G.; Artigues, C.; Hebrard, E.; and Lopez, P. 2015. Scheduling scientific experiments for comet exploration. *Constraints* 20:77–99.

Smith, D. E.; Franck, J.; and Cushing, W. 2008. The anml language. In *ICAPS Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*.