A Demonstration of Dedicated Constraint-Based Planning within Agent-based Architectures for Autonomous Aircraft

Bertrand Allo[1], Christophe Guettier[2], Nelly Lécubin[1]

[1]Axlog Ingéniérie
19-21, rue du 8 Mai 1945, Arcueil, 94110, France
{bertrand.allo, nelly.lecubin}@axlog.fr
http://www.axlog.fr

[2]Xerox PARC
3333 Coyote Hill Road, Palo Alto, CA, 94304, USA
guettier@parc.xerox.com
http://www.parc.xerox.com

Abstract— **Autonomous agents are a challenging concept for future unmanned air operations in hostile environments. Previous aeronautic missions highlight the lack of on-board reasoning abilities to increase the decision making capability, to efficiently react to unexpected events or to adapt plans to unexpected situation changes. If many agent-based system approaches exhibit reasoning functionalities, the complexity of air missions prevents from using the underlying generic models onto realistic missions. By taking advantage of constraint programming techniques, this paper demonstrates how a dedicated planning method can manage unmanned air vehicles into a realistic mission.**

Keywords— **Autonomy, Mission Planning, Multi-Agent System, Constraint Programming**

I. Introduction

Mission planning requires to tackle globally the management of air operations, dealing simultaneously with several related system functionalities and operational needs. As an example, for each mission timeframe, tactical constraints such as aircraft coordination within the formation have to be compliant with system constraints, like resource usage (self-protection, kerosene,...) or aircraft performance.

Recent research carried out in space [JMM+00] and aeronautics [Yav94] domains emphasized the benefit of using Multi Agent Systems (MAS) [HJ96] as a constructive approach to tackle coordination and collaboration problems. MAS allows the design of global intelligent behaviors modeled through symbolic and logical representations [HJ96], [WJ94]. For instance, it is possible to formally specify how several agents can collaborate to perform a global "goal oriented" mission or to perform specific actions.

Those combinatorial problems have been widely investigated in the Constraint Programming (CP) community. Stemming from logic programming, integer and mathematical programming, Constraint Logic Programming (CLP) languages are recognized as powerful tools to cope with difficult and large combinatorial problems [DHS90], [GH99]. Replacing variable unification by constraint satisfaction, it offers higher compositionality to express and solve complex NP-Hard problems requiring mathematical structures.

This work has been led for the Western European Armement Group within the Misure project.

This paper first introduces the specific issues of air missions involving autonomous aircraft (§ II) before representing flying formations as a MAS (§ III). The three following parts (§ IV,V,VI) detail the different variables and constraints used to model the planning problem. We will then express the advantages of the CLP approach in term of solving capabilities in section § III-C. At last, we will present (§ VII) a set of experimentations led on a realistic scenario with their subsequent results.

II. Air missions using autonomous aircraft

A mission is composed of several formations, and is directed by a mission leader. Each formation is in turn decomposed in a wing commander and several wingmen, respectively denoted formation leader and followers in the sequel. The following roles are generally assigned for a given mission:

mission leader commands the set of formations by constructing a global long-term plan with timing directives during mission preparation or cruise flights;
formation leader commands and controls its formation by providing a local medium term plan, striving to respect mission leader directives;
followers control their attitude according to the leader one. In the following, their temporal and spatial representation is assumed to be equivalent to the leader one.

Aircraft within a same formation have far more opportunities for communication and coordination than between formations. Therefore, it is possible to replan within a formation more frequently than for the whole mission. Replacing the pilot also removes its capability to locally plan a subpart of the mission or to control the aircraft in a complex situation. Being a formation leader or a follower, those skills are required when considering reactions to the opponent behavior, and more critically when an unexpected threat occurs. Thus, on-line planning ability becomes necessary at different scales of the mission.

A. Navigation into a hostile environment

Finding a route for each formation within the set of possible navigation points to achieve the whole mission is a difficult matter. Nowadays, this planning problem is solved well in advance (it corresponds to Air Task Order and Air Command Order of NATO procedure, for example) and cannot easily be updated during air operations. The planning problem must consider simultaneously several feasibility conditions:

- *collaborative constraints:* the planning must take into account formations interoperability (for example, when jamming while allocating weapon frequencies;
- *opponent threats:* some flyby areas can be highly risky or may necessitate a specific formation (to perform Suppression of Enemy Air Defense (SEAD), Battle Damage Assessment (BDA), ...);
- *aircraft performances:* a given aircraft must cross a navigation point respecting its own performance such as maximal acceleration and turning rate.
- *available resources:* such as kerosene, self-protection devices (like decoys, jamming pods) or weaponry. Those resource constraints will be represented at the formation level.

B. Optimizing behaviors

Retrieving a feasible solution may not be enough for evaluating a mission. At every level of the mission, the commanding and control problem would also take into account many optimization criteria extracted from a set of assessing parameters such as air operations performance, aircraft survivability and safety as well as mission flexibility. Most of those parameters would necessitate more than a simple optimization criteria within a static planning process. Planning on the fly according to mission and environment updates would certainly tend to increase these parameters and to make the mission more robust to opponent strategy.

III. Flying formations as a Multi-Agent System

A MAS is mapped to the flying formation by associating an agent to each aircraft [BSD+99]. During the mission, formation (resp. mission) leaders solve medium (resp. long) term goals. They correspond to deliberative agents whereas the followers behave like reactive agents, leading to a hybrid deliberative/reactive architecture [HJ96].

A. Constraint model-based planning

The solving efficiency relies on the planning abilities of the proactive agent. In our approach, it consists in solving a set of combinatorial problems expressed as constraint-based models (navigation path or aircraft dynamics). Each addressed problem is modeled separately, but can be solved either independentely or commonly. The modeling method [Jou95], [Fro95], [GP00] extracts invariant from each problem and simplifies them until a tractable expression is found. The models can then be specialized by adding constraints corresponding to real-life assumptions. This approach has yet proven efficient on task scheduling or resource allocation [DHS90], [VSD95], [GH99]. By enabling compositional, generic and flexible way to separate modeling from search strategy, Constraint Logic Programming (CLP) efficiently sustains the approach used. Logical predicates correspond to constraints interpreted over finite domains expressible as $\{U, +, -, *, >, =\}$, $U \in \mathcal{P}(R)$ [VSD95]. Predicates composition is then converted into logical expressions. This leads to a more understandable and modular problem representation.

B. Architecture integration for on-board planning

In our approach, a formation leader must react efficiently to any update. This involves adapting the former plan or computing a new one according to the change importance and the available time. Layered architectures, involving a high-level planning and low-level execution are well suited to combine both behaviors, according to situation awareness [HV97], as successfully experimented during the Deep Space One mission [JMM+00].

B.1 Representing the global problem using multiple models

The modeling and solving phases of the generic addressed problem rely on a multi-model approach. As shown on fig. [1], each model owns internally a set of variables and constraints and so can be solved independently of the others. We compose them by unifying part of the variables and adding inter-model constraints. A plan subsequently correspond to a partial or complete assignement of the variables, according to the goals.

In the addressed domain of on-board planning (see fig. [1]), a complete plan is a set of edges to fly by for each formation, constrained by feasible mechanical parameters (altitude, speed, etc.). In this context, the planning phase consists in solving all the models.

B.2 Goals specification

Schema [2] expresses the great range of possible problems specifiable and tractable goals for defining MAS functionalities. Once the environment is given, any subpart of the variables can be assigned and any subpart of the models can be solved according to the goal. Additional constraints are added by assigning a set of variables or adding a cost function such as cumulated time or global kerozene consumption (§ II-B).

This high level of modularity enables to design several functionalities and integrate them into the multi-agent architecture. For instance, our experimental architecture considers a long term planning functionality assigned to the mission commander, and a medium term control one assigned to each formation leader.

The long term planner (assigned to mission leader) solves all the models and binds all the variables setting mandatory meeting points with temporal synchronization. The mission commander ensures this functionality in time-windows when communication is possible (before the mission or during cruise flight), and delivers its solution to each formation leader. Section § VII-B presents a global planning experiment on a realistic scenario.

The short term controller (assigned to formation leader) adapts solution according to the preliminary instantiation of the models. Then, models used to repair the local plan depends on the formation and accidents such as a tank

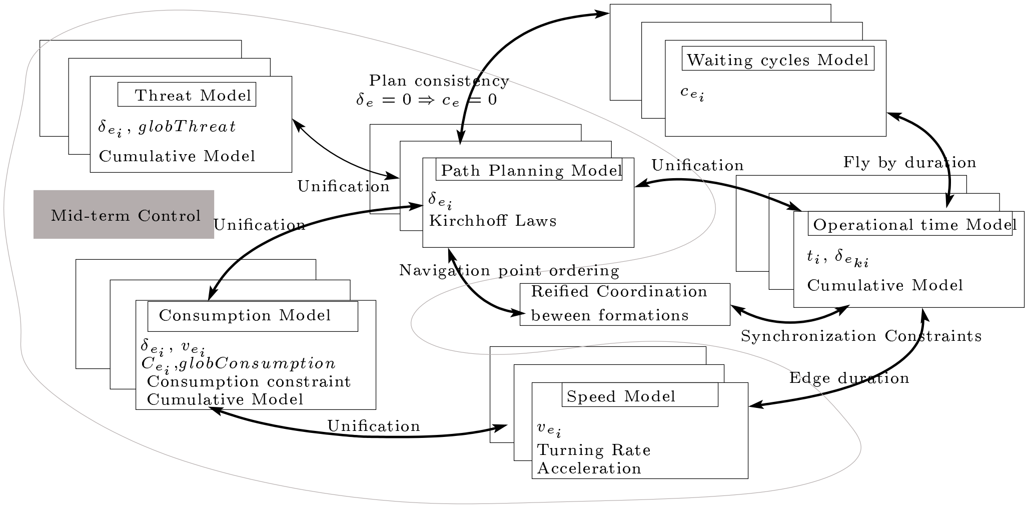

Fig. 1. The multi-model approach of on-board planning

```
For each Formation
  find solutions for {set of variables}
  such that optimize an objective function
    involving a subset of variables
  subject to additional constraints on variables
```

Fig. 2. Specification of a practical problem

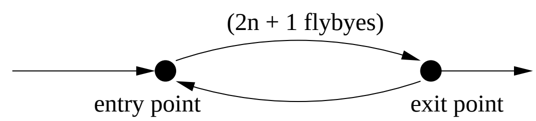

Fig. 3. Flyby of an area

loss or a new threat detection. We detail a significant subproblem instance and two interesting situations in section § VII-C.

C. Using CP solving capabilities

The presented models are purely declarative and can be used in different ways. This section presents how they can deal with slightly heterogeneous problems efficiently at different levels of granularity. Models and search techniques have been implemented using the Sicstus Prolog CLP(FD) library [VSD95]. In order to solve efficiently this global problem, mathematical composition of models is transformed into a concurrent search. Each model is associated with a solving process that explores a local solution space to the corresponding sub-problem. Thus, processes can exchange partial solutions by satisfying relations between models. All solving processes can run simultaneously in order to find a global solution that satisfies all the constraints of the problem.

IV. Environment Model

This model takes into account two physical aspects of the air mission. The spatial representation aims at modeling the static geographical map of the area in which the mission takes place. The threat model matches the enemy's positions.

A. Spatial representation

For each formation, the mission environment is modeled by a set of vertices which are the representation of navigation points. Each vertex has physical coordinates, including altitude. Vertices are linked by oriented edges which are the representation of the *area* the formation must fly by to reach a navigation point from another. Linked navigation points are entry and exit points of the defined area. The formation can wait on an area by flying back to the entry point after the exit navigation point has been reached (see fig. [3]). However, the formation must leave the area through the exit navigation point. Thus the formation can fly by the area $2n + 1$ times, n being the count of *waiting cycles*.

The graph G is denoted by $G = (X, U)$, where X is the set of vertices (*navigation points*), and U is the set of edges (*areas*). It can be dynamically updated by other on-board avionic and positioning systems.

B. Threat Model

The different formations are threatened by a group of radars distributed along the way to the target. Each aircraft can protect itself from the enemy by a limited ability to hide. To represent this fact, we constrain the problem by saying that the self-protection used during the mission must not overcome an available amount.

Each edge is weighted by the threat it represents for a formation to fly along. This threat depends on the altitude of flight, on the minimal distance of the edge from the axis of the radar and on the edge length. The threat is a static characteristic of an edge.

The notations used in the constraints are the same as those represented on fig. [4]. Let P be the nearest point from the radar axis, H be its projection on the axis, M be

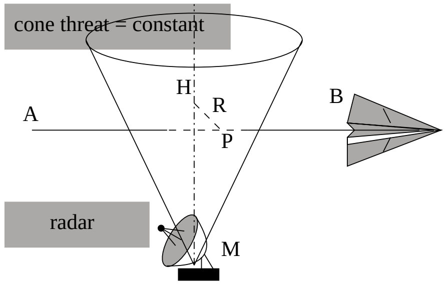

Fig. 4. Threat Model

the radar itself, and P(M) be its power. The edge $e_{A,B}$ is then weighted by the term $T_{e_{A,B}}$:

$$T_{e_{A,B}} = \sum_{M\ radar} \frac{P(M).\|MH\|}{\|MP\|}.\|e_{A,B}\|$$

The global constraint over the graph follows the cumulative model described in the next section, leading to equation (1):

$$\sum_{e\ edge} T_e.\delta_e \leq \sum_{p\ plane} SP_p \tag{1}$$

where SP_p is the amount of self-protection available for aircraft p. This constraint is easy to propagate and very useful to prune the domains of path variables, cutting the complexity of path planning.

V. Single formation planning and control

Different flight models are used to take into account a group of flight parameters : speed, pitch angle, time at different nav points. For each model, specific constraints allow to control efficiently the flight parameters considered for the pilot's safety and the aircraft integrity. Non linear equations extracted from the dynamics of flight [Hal84] are simplified around typical flight values to lead to efficient linear or quadratic constraints.

In the following, let $e_{x,y}$ be the oriented edge linking navigation point x to navigation point y, $\|e_{x,y}\|$ be its length, c_e be the number of cycles around edge e and $v_{e_{x,y}}$ be the average speed flying by it. Let $\delta_{e_{x,y}}$ be 1 if the patrol follows the edge (otherwise 0). Finally, let C_e be the kerosene consumption on edge e. To insure an internal coherence of the models, several constraints will be used to bind the values of related variables.

A. Path navigation model

Path consistency is asserted by the following constraints (2), where $\omega^+(v)$ and $\omega^-(v)$ are respectively the set of edges outgoing from v and incoming into v:

$$\forall v \in X \setminus \{Start\}, \quad \sum_{e\ \in\ \omega^+(v)} \delta_e \leq \sum_{e\ \in\ \omega^-(v)} \delta_e \leq 1 \tag{2}$$

$$\sum_{e\ \in\ \omega^+(Start)} \delta_e \ = \ 1 \tag{3}$$

$$\forall e \in U,\ \delta_e = 0 \Rightarrow c_e = 0 \tag{4}$$

The first inequality in (2) stands for the limit conditions of end of path. Limit condition for the starting navigation point $Start$ is modeled by imposing (3). Finally, equation (4) ensures consistency between path and waiting cycles.

B. Cumulative model for resources and timing constraints

This model is useful for various discrete cumulative constraints, such as timing on navigation points as well as resource consumption (kerosene, self protection). The cumulative models are recursively defined with the following generic formulation, well-known in Operation Research as path algebra formulations [GM95]. $t(v)$ is the intermediate cumulative value when reaching navigation point v, and w_e the local weight associated to area e. We obtain equation (5), where $t(Start) = 0$:

$$\forall v \in X, t(v) = \sum_{e_{u,v}\ \in\ \omega^-(v)} \delta_{e_{u,v}}(w_{e_{u,v}}(2c_{e_{u,v}}+1)+t(u)) \tag{5}$$

C. Dynamics model

The dynamics model manages aircraft attitude using velocity, pitch angle and acceleration. Pairs of possible incoming/outgoing edges are propagated. Physical constraints implied by the aircraft limits are appropriate to prune the domains of the different variables and solve the global problem. The maximum pitch angle ϕ_{max} and the maximum thrust (inducing a maximum acceleration γ_{max}) are taken into account not to deteriorate the cell structure of the aircraft and the pilot's safety. Speed is constrained statically to take values in the domain of flight $[mach\ 0.7, mach\ 1.3]$.

C.1 Speed Variable and Constraints

Speed value is strongly linked by timing to flyby dates τ_a of formation a on the different navigation points, which is modeled by equation (6):

$$\forall e_{x,y} \in U, \tau_a(y) = \tau_a(x) + \frac{\|e_{x,y}\|}{v_{e_{x,y}}} \tag{6}$$

C.2 Turning Rate Constraint

This constraint links the average speed of the aircraft with the pitch angle during turns. In order to simplify the equations, we assume a nominal behavior characterized by:

1. no sideslip during the turn;
2. all the turn is in a same horizontal plane.

In the following feasibility condition, let R be the maximal distance from navigation point y to begin to turn ($R = 138\ kts$), g be the gravity constant, and V be the average speed for the whole turn. The angle between edges e and f is denoted $\alpha_{e,f}$. Beyond $\alpha_{max} = 0.24\ rad$, the turn is always feasible.

$$arctan\left(V^2.\frac{1}{2.R.g}.\alpha_{e,f}\right) \leq \phi_{max}$$

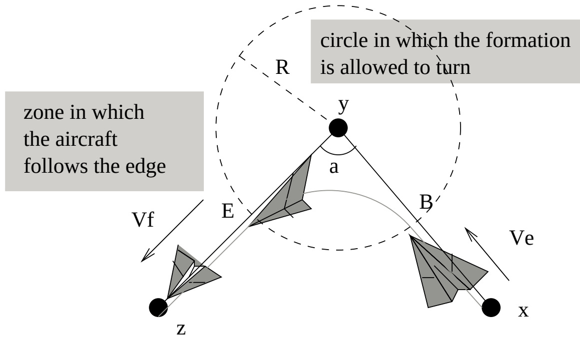

Fig. 5. Model of the turning rate

On the angle domain $[0, \alpha_{max}]$, it is possible to reason on $tan\,(\phi_{max})$. Lastly, by replacing V by the approximation $V = \left(\frac{V_e+V_f}{2}\right)$, the feasibility condition becomes linear according to problem variables, since the right-hand term does not contain any constrained variable and can be statically pre-calculated. This results in inequality (7):

$$V_e + V_f \leq 2.\sqrt{tg(\phi_{max}).\frac{2.R.g}{\alpha_{e,f}}} \tag{7}$$

C.3 Acceleration

The acceleration is discretized on the edge BE. Let G_{max} be the maximum acceleration worth to the pilot or the cell structure. Ensuring safety leads to the condition $\frac{V}{\|BE\|}.(V_f - V_e) \leq G_{max}$. Using the minimal static over approximation with distance $\|BE\| = R.\alpha_{e,f}$, the condition is recasted into the following quadratic constraint (8):

$$\frac{1}{R.\alpha_{e,f}}.(V_f - V_e).(V_f + V_e) \leq 2.G_{max} \tag{8}$$

D. Consumption Constraints

The consumption is calculated on each edge taken by the formation, for a turboreactor plane. The constraints are based on thrust, drag and air density calculus. Let $\rho(h)$ be the air density at altitude h, $h(x)$ be the altitude of vertex x, and $\kappa_{1,1}$, $\kappa_{1,2}$ and $\kappa_{1,3}$ be three constants of the aircraft used, depending of its specific consumption, drag when $incidence = 0$, wing surface and aspect ratio.

The following equation contains a term for a plane flight and one for the overcost induced by the altitude changings.

$$C_{e_{x,y}} = \delta_{e_{x,y}}.\|e_{x,y}\|.\left[\kappa_{1,1}.v^2_{e_{x,y}}.\rho\,(p) + \kappa_{1,2}.\frac{1}{v^2_{e_{x,y}}.\rho\,(p)} + \kappa_{1,3}.\frac{h(y)-h(x)}{v_{e_{x,y}}.\|e_{x,y}\|}\right]$$
$$\text{where } p = \frac{h(x)+h(y)}{2}$$

In the domain of flight considered, we can linearize the equation around *mach* 1 with a limited loss of quality of the results obtained. Replacing $v_{e_{x,y}}$ by $a(1+u_{e_{x,y}})$ where

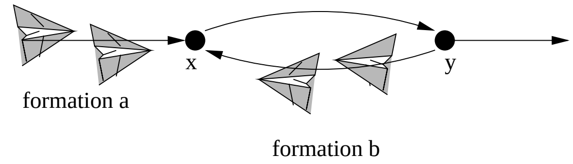

Fig. 6. Formation b covers Formation a while entering area $e_{x,y}$

a is the sound celerity and $u_{e_{x,y}}$ is in $[-0.3, 0.3]$, and then linearizing the equation we obtain (9):

$$C_{e_{x,y}} = \delta_{e_{x,y}}.\left[\kappa_{3,1} + \kappa_{3,2}.u_{e_{x,y}}\right] \tag{9}$$

where $\kappa_{3,1}$ and $\kappa_{3,2}$ are linear combinations of $\kappa_{1,1}$, $\kappa_{1,2}$ and $\kappa_{1,3}$ ponderated by terms containing $\rho(p)$ and a.

The global consumption is obtained for each aircraft by cumulating the consumption on each edge. It is urged not to overcome the initial amount of kerozene, which can reveal pruning-efficient at the end of the mission.

VI. Inter formation planning with collaborative models

Inter-formation coordination and collaboration can be defined by a new constraint set called *collaboration constraints*.

Let $t_a(x)$ be the flyby date of formation a on navigation point x, as defined by the cumulative model. As x is an exit point for an area and an entry point for another area, $t_a(x)$ is the date of a transition between two areas. A basic coordination constraint will be defined as (10):

$$t_a(x) + d_{min} \;\leq\; t_b(y) \;\leq\; t_a(x) + d_{max} \tag{10}$$

Thus, another transition date $t_b(y)$ can be constrained to a sliding time window of fixed width $d_{max} - d_{min}$, depending on $t_a(x)$.

This generic constraint scheme acts as a powerful basis for building any higher level collaboration constraint. We have actually implemented several such constraints like exclusive or joint flyby of an area, formation covering while entering, crossing or exiting an area, and successive operation in an area. In all these constraints, d_{min} and d_{max} remain useful for defining minimum and maximum delays, making constraints more or less flexible.

VII. Experimentation on a realistic scenario

Experimentation on the models described above (§ IV,V,VI) has been done on a realistic scenario depicted in fig. [9]. The problem is to find a feasible navigation plan for 4 flying formations among 28 nav points, connected by 65 edges, satisfying 12 coordination constraints and opposed to 14 sol-air sites threats. Realistic values have also been chosen for aircraft parameters, characterizing modern air fighters. The overall mission is depicted in fig. [7], where the expert solution is represented.

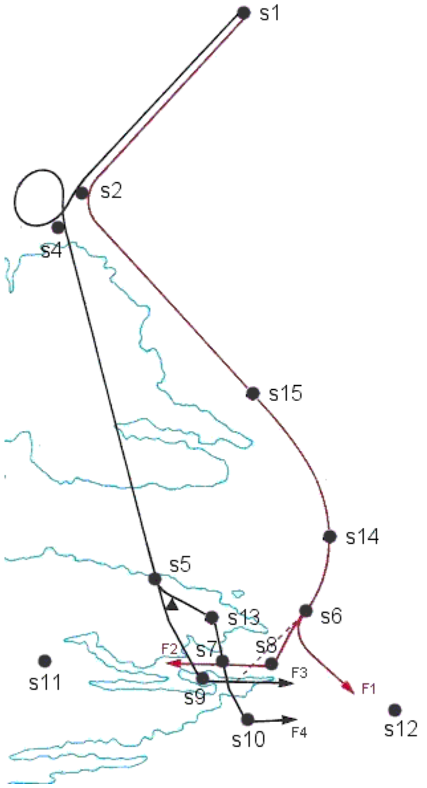

Fig. 7. Expert solution on the real map

```
For each Formation
  find solutions for {all models variables}
  such that {minimize the mission completion time}
  subject to coordination constraints between formations
```

Fig. 8. Specification of the long-term planning problem

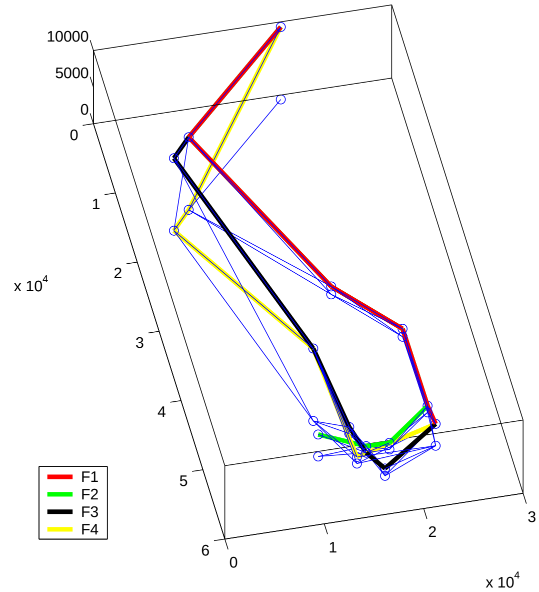

Fig. 9. Planner solution on the graph structure

A. *Mission inter-operability*

All formations are coming from the same nav point s_1. Formations F_1 and F_2 must cross simultaneously nav point s_6 to perform SEAD. Formations F_3 and F_4 must cross nav point s_5, and then take different routes (resp. by s_7, s_{10}, s_{12} and s_9, s_{12}). Threats are localized in the s_9, s_8 zone, where the mission objective is (imposed as a constraint).

Formation F_2 must cross nav point s_7 *before* formation F_3, to perform a BDA. In the same way, to satisfy system inter-operability, formation F_3 must cross nav point s_7 *before* formation F_4 crosses nav point s_9. Formations F_1, F_3 and F_4 must escape by nav point s_{12} and formation F_2 by nav point s_{11}. Those mission interoperability requirements have been represented using the coordination formalisms.

B. *The long-term planning problem instance*

In our approach, solving the global problem corresponds to a long-term planning function assigned to the mission leader. The problem under consideration involves solving a conjunction of all the models described in fig. [1] and minimizing the total duration of the mission as a cost objective, such as formulated in fig. [8]. Fig. [7] presents the expert solution, characterized by a loop between s_2 and s_4 for satisfying coordination constraints.

Fig. [9] pictures the graph that models the mission environment, structuring the input problem[1]. Aircraft potential trajectories have been interpolated and involve several altitudes for more accurate experimentations. The paths of the solution described in the sequel are highlighted.

[1]The experiments have been performed using the standard international units $(m, m.s^{-1}, m.s^{-2})$

B.1 On the problem complexity

Due to the number of variables and the hybrid nature of the global problem, it is difficult to give a good approximation of the complexity. The problem is to solve a feasible navigation plan under cumulative constraints. It differs from the class of scheduling problems with cumulative resources (known to be NP-hard) as we must construct a consistent set of tasks and not only assign a timeline to each possible task. Furthermore, compared to traditional planning approaches [W98] (also characterized as NP-hard), additional domain-specific constraints are taken into account. In the presented example, the problem instance is composed of 792 discrete variables and 15186 constraints. Optimization stands for minimizing the global mission duration, other optimization criteria closer to criteria presented in § II-B can also be formulated.

B.2 Experimenting the long-term planning functionality

A solution is retrieved by our implementation in 30 seconds on a Pentium II/500. For simplicity, the following tables summarize results and give main values. Timings are given in seconds, while the altitudes are denoted low ($< 3500\ ft$) and high ($> 3500\ ft$).

F_1 : $s_1 \rightarrow s_2 \rightarrow s_{15} \rightarrow s_{14} \rightarrow s_6 \rightarrow s_{12}$

area	t_{entry}	t_{exit}	speed	cycles	altitude
$s_1 \rightarrow s_2$	0	55	0.9	0	high
$s_2 \rightarrow s_{15}$	55	106	1.1	0	high
$s_{15} \rightarrow s_{14}$	106	125	1.3	0	high
$s_{14} \rightarrow s_6$	125	149	1.3	0	high
$s_6 \rightarrow s_{12}$	149	163	1.1	0	high

F_2 : $s_1 \rightarrow s_2 \rightarrow s_{15} \rightarrow s_{14} \rightarrow s_6 \rightarrow s_8 \rightarrow s_7 \rightarrow s_{11}$

area	t_{entry}	t_{exit}	speed	cycles	altitude
$s_1 \rightarrow s_2$	0	55	0.9	0	high
$s_2 \rightarrow s_{15}$	55	106	1.1	0	high
$s_{15} \rightarrow s_{14}$	106	125	1.3	0	high
$s_{14} \rightarrow s_6$	125	149	1.3	0	high
$s_6 \rightarrow s_8$	149	162	1.3	0	high
$s_8 \rightarrow s_7$	162	166	1.3	0	high
$s_7 \rightarrow s_{11}$	166	178	1.2	0	high

F_3 : $s_1 \rightarrow s_2 \rightarrow s_4 \rightarrow s_5 \rightarrow s_{13} \rightarrow s_7 \rightarrow s_{10} \rightarrow s_{12}$

area	t_{entry}	t_{exit}	speed	cycles	altitude
$s_1 \rightarrow s_2$	0	45	1.1	0	high
$s_2 \rightarrow s_4$	45	53	1.1	0	high
$s_4 \rightarrow s_5$	53	150	0.9	0	high $\rightarrow$ low
$s_5 \rightarrow s_{13}$	149	166	0.7	0	low
$s_{13} \rightarrow s_7$	166	172	1.3	0	low
$s_7 \rightarrow s_{10}$	172	181	1.3	0	low $\rightarrow$ high
$s_{10} \rightarrow s_{10}$ [a]	181	183	1.3	0	high $\rightarrow$ low
$s_{10} \rightarrow s_{12}$	183	202	1.0	0	low $\rightarrow$ high

[a] F_3 went into a dive because of a limited amount of protection

F_4 : $s_1 \rightarrow 2 \times (s_2 \rightarrow s_4) \rightarrow s_5 \rightarrow s_9 \rightarrow s_{12}$

area	t_{entry}	t_{exit}	speed	cycles	altitude
$s_1 \rightarrow s_2$	0	43	1.3	0	high $\rightarrow$ low
$s_2 \rightarrow s_4$	43	82	0.7	1	low
$s_4 \rightarrow s_5$	82	165	1.0	0	low
$s_5 \rightarrow s_9$	165	192	0.8	0	low $\rightarrow$ high
$s_9 \rightarrow s_{12}$	192	209	1.3	0	high

Compared to the solution given by military experts (vertices s_2, s_4, fig. [7]), the waiting points are located on similar edges. Furthermore, the formation speeds and altitudes are varying when necessary.

C. The formation command and control medium term planning problem

Once a first global plan has been delivered by the mission leader, each formation leader can refine and adapt its own plan according to a more accurate representation of its immediate environment. In this scenario, coordination constraints have been already solved by the mission leader and are relaxed in the cost function for each formation.

Starting from the initial global plan, each formation can solve incrementally its own plan by minimizing delays with pre-planed meeting dates. The global problem represented in fig. [1] can be decomposed into independent sub-problems specified in fig. [10]. Therefore, the problem is distributed over the set of formations, where the global quality of coordination is the common objective.

```
For each Formation
  find solutions for {path planning, resource, speed}
  such that {maximize the synchronization
            with the global long-term plan}
  subject to □
```

Fig. 10. Specification of the formation command and control problem

In each experiment, the planner gives a first solution in

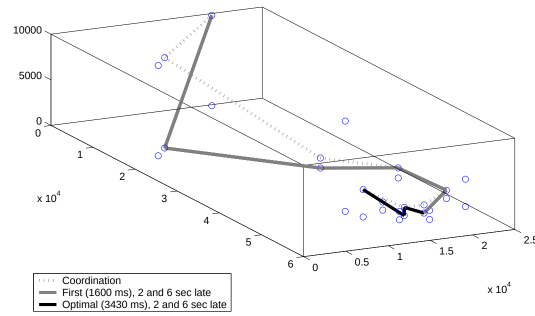

Fig. 11. Formation planning with over-constrained self-protection (5000)

a few seconds ($0.64\ s, 1.60\ s, 1.29\ s, 1.28\ s$)[2]. Although we may not guarantee an optimal solution within aircraft dynamics time frame, due to the incremental nature of the optimization, good solutions (represented as grey lines in fig. [11]) can be found in reasonable time.

C.1 Resource constraints experiments

As a first problem instance, the global planning level may have over-estimated the self protection resource of formation F_2. Thus the existing plan is no longer feasible, and should be locally recovered by the formation leader. The following experiments show various situations for this formation where the selfprotection resource is bounded[3] respectively by 9000, 7000, 5000 instead of 10000 as asserted by the mission leader. Satisfying the actual resource level generates safer trajectories characterized by an altitude lower than the initial one (represented in dashed lines in fig. [11]), but delays some dates of synchronization of the global plan. An optimal solution is found for each situation (represented as a dark line in fig. [11]), that characterizes a trade-off between safer altitudes and short delays. In the three examples, it took half a minute for the planner to find the optimal solution in the worst case (9000).

C.2 Unexpected threat experiments

In the second problem instance, an unexpected threat is discovered by the formation between s_2 *and* s_6 (represented as a cross in fig. [12]). This urges the formation leader to replan an escape path that maximize the ability for the mission leader to recover the whole mission. As shown in fig. [12], several potential paths are proposed by the onboard mission manager as an update of the input graph structure. The planner chooses a safer path, but delays the meeting date up to 13 s.

VIII. Conclusion

We have proposed a highly modular constraint based formulation for planning aircraft missions that involve autonomous behaviors. A better alternative has been given

[2]Those values correspond to a run performed by a code interpreter, and can be divided by three by optimized compilation

[3]The unit of this model is arbitrary, it represents the aircraft ability to jam a threat.

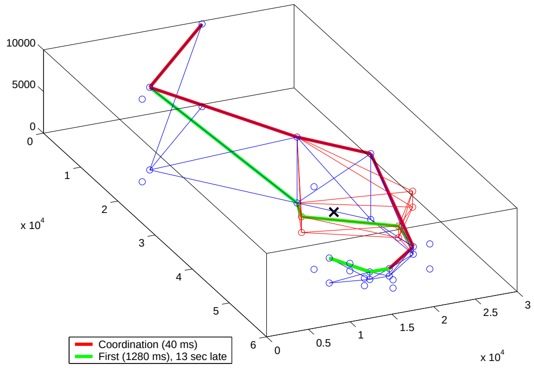

Fig. 12. Formation planning with an unexpected threat

to heuristic-based behavior of traditional multi-agent planners and a new way, sketched in [GP00], has been opened to tackle complex cooperative behaviors. We have demonstrated that the solving methods described in this paper can be integrated into dedicated multi-agent architectures at different levels, according to reactivity and locality trade-off. Furthermore, depending on the situation awareness, a multi-agent architecture can decide to solve over a same related model-based representation or to distribute the solving over a set of independent problem instances.

The pertinence of using Constraint Model Based Programming for specifying and solving the complex problem of a Multi-Agent plan has been shown. Generally studied independently, several models extracted from heterogeneous domains, such as the theory of dynamics of flight, tactics and operational research, have been expressed in a single formulation. The set of models is not exhaustive and many other domains may also be addressed. This work highlights the feasibility of the approach for tackling complex MAS problems by solving these different models concurrently.

This demonstration relies onto approximations, discrete representations that have been performed to implement finite domain constraints. Those models are interesting for medium and long term planning but should be refined to achieve short term control. In spite of those approximations, experiments onto realistic scenarios have exhibited interesting results, relevant to the operational expectations summarized in section § II-A and § II-B. Furthermore, with little work on solving strategies, the computation time remains in mission planning timeframe. Lastly, providing an optimized solution at any time is of a particular interest for embedded purposes.

Further works will investigate finer grain anytime search strategies for a better reactivity, when minor changes to the current plan are needed. Stronger search strategies, involving incremental concurrent optimization and branch and bound can also be studied for major planning updates. To be efficiently integrated into future MAS architectures, trade-offs between these different levels of reactivity have to be formalized. Lastly, the distribution of search over several agents will also be extended to weakly dependent sub-problems.

Acknowledgements

We would like to thank Vincent Legendre, our predoctorate student, who led the experimentations. We are grateful towards Eric Bornschlegl and Markus Fromherz for their constructive comments on the paper. Thanks to Markus for his reading and his precious advice.

References

[BSD+99] C.Barrouil, P.da Silva-Passos, Y.Davaine, R.Demolombe, P.Fabiani, B.Patin, N.Prego, C.Tessier, “Tandem: an Agent oriented approach for Mixed System Management in Air Operations”, *NATO-RTO Symposium on Advanced Mission Management and System Integration Technologies for Improved Tactical Operations*, Florence, Italy, Sept. 1999.

[DHS90] M.Dincbas, P.Van Hentenryck, H.Simonis, “Solving Large Combinatorial Problems in Logic Programming”, *Journal of Logic Programming*, Vol. 8, p.75-93, 1990.

[Fro95] M.Fromherz, V.Saraswat, “Model-Based Computing: Using Concurrent Constraint Programming for Modeling and Model Compilation”, *Proceedings of CP'95*, Industrial session, 1995.

[GH99] C.Guettier, J.-F.Hermant, “A Constraint Based Model for High Performance Real Time Computing”, *Proceedings of the International Conference on Parallel and Distributed Computing Systems*, Florida, 1999.

[GP00] C.Guettier, J.-C.Poncet, “Automatic Planning for Autonomous Spacecraft Constellations”, *Proceedings of the Int. NASA Workshop on Planning and Scheduling for Space*, San-Francisco, USA, 2000

[GM95] M.Gondran, M.Minoux, *Graphes et algorithmes*, ed. Eyrolles, 1995.

[Hal84] Francis J.Hale, *Introduction to Aircraft Performance, Selection and Design*, John Wiley and Sons, Inc., 1984, ISBN 0-471-07885-9

[HV97] K.Z.Haigh and M.M.Veloso. “High-Level Planning and Low Level Execution Towards a Complete Robotic Agent.”, *Proceedings of the First Int. Conf. On Autonomous Agents*, Feb. 1997, Menlo Park, CA.

[HJ96] J.M.P.O'Hare, N.R.Jennings, *Foundations of Distributed Artificial Intelligence*, Sixth-Generation Computer Technology series, 1996.

[JMM+00] A.Jónsson, P.Morris, M.Muscettola, K.Rajan, B.Smith, “Planning in Interplanetary Space: Theory and Practice”, *Proc. of the Second NASA International Workshop on Planning and Scheduling for Space*, March 2000.

[Jou95] J.Jourdan, *Concurrence et coopération de modèles multiples dans les langages de contraintes CLP et CC* PhD thesis, Université Denis Diderot, Paris VII, 1995.

[VSD95] P.Van Hentenryck, V.Saraswat, Y.Deville, “Design, Implementation and Evaluation of the Constraint Language CC(FD)”, *Constraint Programming: Basics and Trends*, A. Podelski Ed., pp. 68-90, 1995.

[W98] D.S.Weld, “Recent advances in AI Planning”, *Tech Report UW-CSE-98-10-01, Dept. of CSE, Univ. of Washington, 1998; also in AI Magazine*, 1999

[WJ94] M.Woodridge, N.R.Jennings, “Intelligent Agents: Theory and Practice”, *Knowledge Engeneering Review*, October 1994.

[Yav94] A.Yavnai. “Distributed Decentralized Architecture for Autonomous Cooperative Operation of Multiple Agent System”, *Proceedings of the IEEE 1994 Symp. on Autonomous Underwater Vehicle Technology.*