Formalizing and Solving Information Collection Problems with Autonomous Sensor Systems

M. Godichaud*, E. Chanthery*, O. Buffet, M. Contat*****

*CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse, France
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse France
(e-mail:matthieu.godichaud; elodie.chanthery @laas.fr)
**INRIA, Nancy Université, CNRS;LORIA – Campus Scientifique, Nancy, France
(e-mail: olivier.buffet@loria.fr)
*** Cassidian, 1 Bd Jean Moulin, CS 40001, 78996 Elancourt cedex, France
(e-mail : marc.contat@cassidian.com)

Abstract: This paper addresses the problem of planning data collection missions for a set of Information Collection Systems (ICS) to respond to a set of information requests. This problem goes from the formalization of information needs to the optimization of ICS actions. After having formalized requests and decomposed them into elementary requests, the problem can be modeled with a graph characterizing the various aspects: coordination and assignment of ICSs, request satisfaction and ICS use optimization. Based on this graph, the problem can be solved with an A*-like search algorithm.

Keywords: Information systems, autonomous vehicles, search methods, self-optimizing systems, hierarchical decision making.

1. INTRODUCTION

The purpose of data collection missions is to obtain information to make decisions about important issues. It finds an echo in many applicative domains like defense, civil or industrial domain. The goal is to provide enough information to efficiently guide choices. Consequently, the quality and the amount of information in the collected data will have a direct impact on future decisions. Whatever the situation, information is produced by a finite set of heterogeneous means. This implies rationalizing and optimizing the use of these means to efficiently respond to different information requests formalized by an operator who has to take decisions. The objective of this work[1] is to provide appropriate tools to operators who define the use of autonomous sensor systems called Information Collection Systems (ICSs, i.e., humans, UAVs, radars …), in order to answer to a set of Information Requests (IRs).

Data collection optimization in geographical observation missions is essential but remains poorly developed (Janez (2007)). Our work aims at approaching these problems with an overall view of the information management process, i.e., from the formalization of input data, characterized by information requests in the form of free text, to the optimization of ICS mission plans.

A data collection mission for a given ICS is carried out by realizing actions (like motion actions, actions on the environment or data collection). ICS mission planning consist in selecting information to be collected by each ICS and scheduling the actions to realize taking into account the environment and ICS characteristic. Quality and quantity of collected information, action realization cost and various constraints have to be taken into account in optimization of data collection. Most of the time data collection missions involve the use of various ICSs. It is thus necessary to determine the action plans of multiple ICS and to enable cooperation. The problem of planning for sensor systems is discussed by Janez (2007), in a military context. The author proposes a planning model based on a vehicle routing problem. IRs are characterized by points on the geographical area of interest and ICSs have to visit these points. The heterogeneity of ICSs is taken into account through the points they can visit and their travelling times. This approach determines a first "coarse" plan by assigning points (IRs) to ICSs. To refine the plan, planning models specific to each ICS can be used. When multiple ICSs of different types are available, the resolution consists in assigning targets to ICSs and determining trajectories. In the case of UAVs, target assignment is addressed by Schumacher et al. (2002) and Rasmussen et al. (2003) for instance. Models are proposed by Richards et al. (2002) for selection and assignment of targets and determination of trajectories. The cooperation between systems allows balancing the work and, usually, improves information collection. Cooperative approaches are proposed for ground vehicles by Cook et al. (1996) and for aerial drones by Chandler et al. (2001). Satellites can also collect information and models have been developed in this way to select and schedule goals during a mission (Vasquez and Hao (2001), Lemaître et al. (2002)). An approach to build a plan for airborne observation is developed by Frank and Kurklu (2003). These works show that various ICS types can be used in an observation mission. A formalization of mission planning problems, generic enough to be applied to different types of ICSs, has been proposed by Chanthery et al. (2005). Furthermore, this model allows at a low level representing motion complexity and practical constraints (duration of

[1]funded by the French Defence Agency

action, consumption …). It has been compared to other works on task planning by Chanthery (2005). This formalization, however, applies only to missions with a single ICS. We suggest extending it to a multi-ICSs context and present how to use it for coordination of ICSs.

In this article, we present a formalization of the ICS optimization problem. In Section 2, the problem statement is presented with a description of various concepts that have to be taken into account. Section 3 gives a global view of the resolution process. According to this view, a model and a resolution algorithm are presented in Section 4. An illustrative example which gives a proof of concept is presented in Section 5. Section 6 gives conclusions and perspectives.

2. MODELLING REQUESTS

The goal of Sections 2 and 3 is to clearly express the data collection problem and to identify constraints to meet and the criterion to optimize for this type of problem. The problem consists in finding an appropriate timetable for a set of ICSs in order to answer to a set of IRs. This Constrained Optimization Problem (COP) is extremely complex because IRs are not formalized at the beginning of the process and the number of ICSs that can deal with these IRs is huge.

2.1. IR, FIR, EIR

Information Requests (IR) are formulated by an operator, with constraints (start date, end date, interest zone or target) and priority level. A Formalized Information Request (FIR) is an IR which is sufficiently detailed to be decomposed into Elementary Information Requests (EIR). The minimum information required to define an FIR is a temporal interval of validity, an interest zone or target and a priority level. The set of EIRs is finite and composed by “atomic” elements, that is to say elements that cannot be decomposed anymore. An EIR inherits from its parent FIR the temporal interval of validity, the interest zone or target and the priority level. In addition, an EIR is defined by its type (detection, recognition, on site intervention, evaluation reports…), inputs parameters and output parameters (desired information). EIRs related to the same FIR are linked by logical and temporal constraints.

Let sat(EIR_i) be the variable indicating that EIR_i is satisfied. Then for a given FIR, logical constraints that linked its EIR are described by a unique logic formula that includes the variable sat(EIR_i) associated with each EIR. For example, given two EIRs (EIR_1 and EIR_2), it is possible to express that at least one EIR is satisfied: “sat(EIR_1) OR sat(EIR_2)”, or that both are satisfied: “sat(EIR_1) AND sat(EIR_2). In the first case, a solution may satisfy only one of the two EIRs – for example if costs are too important – or it may satisfy both EIRs – for example if the priority level of the information request is high.

Once logical constraints are defined, it is possible to add temporal constraints to EIRs. Temporal constraints may be very expressive if the reasoning implies start and end dates of EIRs, respectively denoted start(EIR_i) and end(EIR_i) for EIR_i. For example: « end(EIR_1) < start(EIR_2) » means that EIR_1 has to end before the beginning of EIR_2; « (start(EIR_1) < start(EIR_2)) AND (end(EIR_1) > end(EIR_2)) » means that EIR_2 has to be realized during EIR_1; « (start(EIR_1) = start(EIR_2)) AND (end(EIR_1) = end(EIR_2)) » means that EIR_1 and EIR_2 are realized simultaneaously; « end(EIR_1) < *d* » means that EIR_1 has to end before the date *d*.

2.2. Information Collection Systems (ICS)

The set of ICSs includes all the possible means of collecting information. For example, in the case of fire detection, they can be on-site fire detectors, satellite images, human observations, etc. Each ICS is characterized by one carrier, a set of on-board sensors abilities to exploit data like transmission and processing abilities. The operator has to select one ICS for each EIR, meeting logical and temporal constraints, and optimizing the sequence of actions to achieve the EIR. Each EIR is associated with a set of actions achieved by the chosen ICS. Obviously, the set of actions to achieve an EIR depends on the ICS capabilities. Here are some generic properties for ICSs that are taken into account by our COP: (1) the type of ICS from a finite list of types indicates a set of specific constraints and parameters that have to be considered (for example, it is useful to constrain the altitude of a UAV but not for a ground vehicle); (2) IR types that the ICS can accept; (3) on-board sensors, which induces a set of specific constraints and parameters that have to be considered and a set of constraints that restrict the choice of ICSs for a set of given IRs; (4) autonomy and resources consumption; (5) memory capacity that limits the amount of information that can be stored by the ICS during its mission; (6) the range of on-board sensors that constraints the allocation of the ICSs to some IRs if some others are out of range for example.

Moreover, at a given interest point, the ICS may eventually achieve several IRs if its range allows it to. It is required to take into account the action range of the ICS, its transmission type which can constraint the trajectory, if an ICS has to pass on transmission points for achieving IRs, analysis type (on-board/ off-board), costs parameters (use, buying price) that are used as performance indicators.

2.3. Targets and environment

Targets are associated with EIRs. They are described by a type and characteristics (size, speed, visibility…). One ICS will be more or less efficient than another according to its capacity and the known characteristics of the targets. For example, if the target is moving, the ICS has to be able to follow it; if the target is hostile, the ICS has to be furtive …

The environment of the IRs induces constraints and performance indicators. The geographical environment is decomposed into zones which must be avoided – a solution in which no ICS crosses any of these zones is preferred (this may be formalized by a cost, for example) –; forbidden zones (formalized as constraints); waypoints with temporal validity intervals (constraints) and transmission waypoints. Different possible obstacles also induce intervisibility constraints. Weather conditions induce performance constraints and sometimes degradation of performances for moving ICSs.

3. DECISION PROCESS

Section 2 shows that the problem is complex to solve. The most appropriate solution to tackle it is to structure the problem into a hierarchy, which means to decompose its resolution in several sub-problems in order to solve only tractable size problems. This solution remains generic because hierarchical levels are not compartmentalized: if the resolution fails at one level, it is possible to redo the work at an upper level.

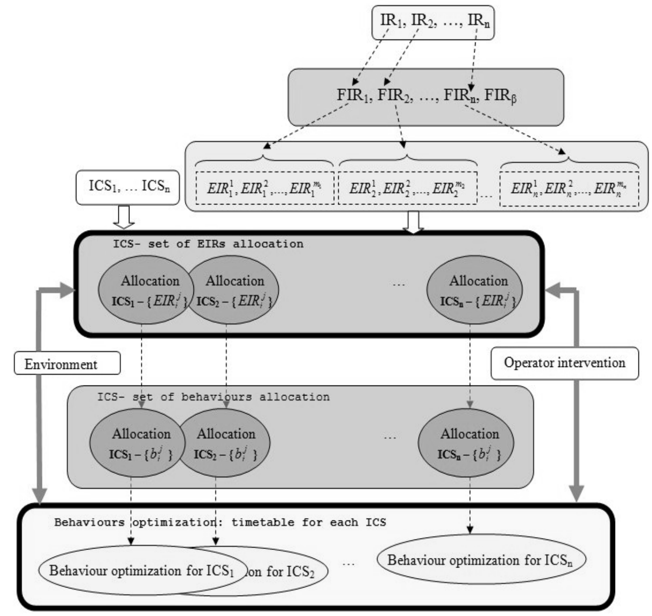

Fig. 1. Global view of the decision process

Fig. 1 summarizes the approach. The top of the figure corresponds to the problem input i.e. the IRs. They give a set of objectives in terms of observation goals that have to be realized by ICSs. A first step which is not in the scope of this paper is the IRs transformation into FIRs that can be automatically translated and treated by a computer or an automatic process. The FIRs are decomposed into EIRs which can be interpreted as a set of logical sentence between ICS actions. The lower level of the figure corresponds to the output of the optimization problem: each chosen ICS has an appropriate timetable in order to respond to a maximum number of FIRs.

The optimization problem is decomposed into two decision levels in bold on the figure: the first level (COP_1) corresponds to an allocation of EIRs to each ICS; the second level (COP_2) corresponds to an optimization of a set of elementary actions of each ICS in order to respond to a set of EIRs.

4. MISSION PLANNING PROBLEM

In the previous sections, the problem has been described thanks to the identification of various concepts which have to be modeled. A graph representation of all potential problem solutions from requests allocation to the planning of the ICS mission is adopted.

4.1. Model structure

The model is structured into two layers in order to take advantage of the abstraction principle as in Fig. 2. The top layer allows managing COP_1 and the bottom layer allows managing COP_2. The bottom layer represents all the available ICSs with EIRs to which they can respond. A planning model presented by Chanthery et al. (2005) is used to represent each ICS. It is a graph-based model structured in two levels.

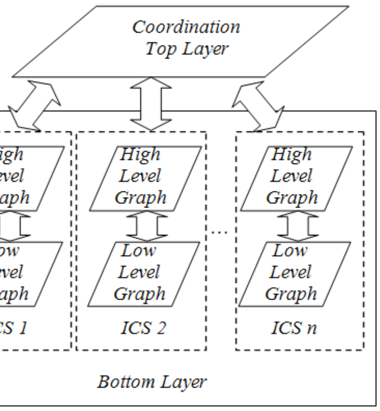

Fig. 2. Hierarchical model

For a given ICS, a low-level graph consists of a set of nodes, denoted *N*. n_1 is the origin node: it corresponds to the position of the ICS when the plan is applied. Given two successive nodes n_k and n_{k+1} a decision made between n_k and n_{k+1} will be applied at n_{k+1}. From a physical point of view, n_{k+1} corresponds to the action that the ICS is performing. In this approach, such a graph is developed for each ICS given all the EIRs to which it can respond. The high level is constructed from a partition of *N* noted *W* in order to describe the response to EIRs by the ICS and manage the satisfaction of these EIRs. For $W = \{W_1,...,W_e\}$, we define the relation S as follows: for a subset W_i, S(W_i) is a subset of *W* and is defined as the subset of W_i successors. A finite state machine is defined with *W* and the relation *S*. Without loss of generality, W_1 corresponds to the initial state of the ICS and W_e to the end of an ICS plan.

Given two nodes *n* and *m* from N, there exists one or more arcs between *n* and *m* if and only if there exists *i* and *j* with *n* in W_i, *m* in W_j and W_j in $S(W_i)$. This principle is applied to each graph corresponding to an ICS. This model allows furthermore more elements to be taken into account: the data and functions describing ICS dynamics; resources are function of time, actions and paths chosen in the graph and the environment; the time constraints on each node n, and constraints on resources. At the high level, selection and schedule of information are managed. At the low level, selection and optimization of action to be realized to collect the information are managed. Example of actions are presented by Chanthery (2005).

4.2. Coordination layer

The aim of the optimization is to find, for each ICS c, a sequence Q_c of state $W_{\pi_c(1)},\ldots, W_{\pi_c(q_c)}$ such that π_c is a function from $\{1,\ldots,q_c\}$ to $\{1,\ldots,e\}$ with : $\pi_c(1)=1$; $\pi_c(q_c) = e$; $W_{\pi_c(i+1)} \in S(W_{\pi_c(i)})$. Satisfying a set of EIRs linked by various relationships, generates a reward. The set of sequences must minimize the difference between realization costs and rewards while satisfying constraints. The coordination layer allows building sequences with respect to relationships between EIRs. A global plan for all ICSs can be defined in several steps. Each step corresponds to the state change of an ICS. The global plan is represented by a set of supernodes forming a high level graph.

A plan step is characterized by a supernode that represents a set of states corresponding to a high level node on the bottom layer. If we consider for instance two ICSs for a mission, the plan will consist of supernodes, each of which is characterized by a set of two high level nodes for each ICS. Possible transitions from a supernode V_x characterized by state W_i for ICS1 and state W_j for ICS2 are shown on Fig.3**Erreur ! Source du renvoi introuvable.**.

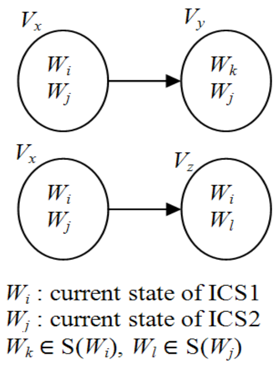

Fig. 3. Possible transitions between supernodes.

A supernode allows determining rewards to be generated at each step of the plan depending on the states reached by each state machine associated to the ICSs. Indeed, reward generations can only be established at the top layer given that they are associated to each FIR satisfaction characterized by the satisfaction of an EIR set (managed at the bottom layer). The satisfaction of EIR can be established and the relationship between them can be verified depending on the states reached at a supernode. Furthermore, authorized transitions in each ICS state machine can be updated according to the states which all ICS state machines have already reached.

Let P be the set of EIRs that can be realized by an ICS. A response to EIR o of P is defined by a triplet $(W_{s(o)} ; W_{e(o)} ; W_{r(o)})$ in W^3 with $s(o)$, $e(o)$ and $r(o)$ in $\{1,\ldots,e\}$. Physically $W_{s(o)}$, $W_{e(o)}$ and $W_{r(o)}$ correspond respectively to the beginning of the EIR, the end of the EIR and finally the transmission and study of the request informations. An EIR is realized when a transition from $W_{s(o)}$ to $W_{e(o)}$ is triggered. An EIR is satisfied when it is realized and a state machine is in state $W_{r(o)}$. For each EIR o, the state machine can be in state $W_{e(o)}$ only one time. Once an EIR is satisfied, it is taken into account by the top layer which evaluates a reward according to the states of the various state machines associated with each ICS.

In order to verify an FIR's satisfaction and to determine if the related reward can be generated, the satisfaction of the set of EIRs related to the FIR has to be verified. In the case of an OR relationship (a disjunction) between several EIRs, the related FIR reward is generated when one of the EIRs is satisfied: one of the state machines is in corresponding state $W_{r(o)}$ (transition from $W_{s(o)}$ to $W_{e(o)}$ has been fired) and the reward is generated at the supernode including this state. In the case of an AND relationship (a conjunction) between several EIRs, the related FIR reward is generated when all the EIRs are satisfied: all the state machines must have reached each EIR's $W_{r(o)}$ state. The reward is generated at the supernode including one $W_{r(o)}$ state and the others $W_{r(o)}$ states are included in the set of the supernode predecessors. A date t_k is evaluated for each state W_k which allows verifying time relationships. Indeed, the set of dates of a supernode sequence can be compared. For instance, to verify if an EIR $o1$ starts after an EIR $o2$, the dates associated to states $W_{s(o1)}$ and $W_{s(o2)}$ can be compared (Fig. 4) ; to verify if $o2$ begins only when $o1$ is finished, the dates associated to states $W_{e(o1)}$ and $W_{s(o2)}$ can be compared.

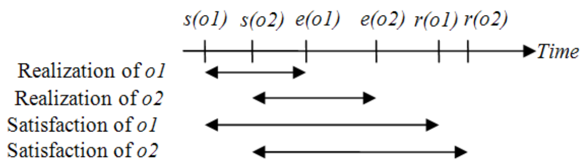

Fig. 4. Temporal relationships between EIR

4.3. Optimization scheme

The previous sub-section presented a graph based model to model the planning problem. A search algorithm for this graph is presented in this sub-section. It is an A*-like algorithm, which has been adapted to the mission planning problem. It allows for managing the coordination between ICSs by exploiting the concept of supernode. The algorithm is given on Fig. 5. It gives an overview of the algorithm with the main steps which have to develop to use it for a given application. Indeed, each method of the algorithm has to be specialized given the context. V_1 corresponds to the beginning of the plan; it contains node n_1 for each ICS. F is the criterion to be optimized. *ListN(V)* is the node list of a supernode V. n_i is a node and $S(n_i)$ is the set of its successors. *ListP* is the list of pending supernodes. $\hat{U}$ is the first node of *ListP*. I is a sub-criterion corresponding to the evaluation of one ICS plan. *ListT* is a node list and *ListS* is a supernode list.

The origin supernode V_I is defined with each origin node n_I. V_I is initially put in the pending supernode list *ListP* (line 1). The iterative process consists in developping the first element of the pending supernode list. This supernode is noted $\hat{U}$. Node development (line 5) consists in identifying its successors and evaluating the plan of the corresponding ICS going from n_I to the successor. The obtained value is noted I (line 9). This evaluation may integrate various realization costs (resource consumption, duration, risk …) of one ICS plan. Futhermore, this evaluation allows pruning the exploration tree if violations of ressources and temporal constraints are detected. The global criterion value F is evaluated for the plan going from V_I to V_j (line 17). The global criterion corresponds to the sum of sub-criteria related to each node of V_j plus value of the partial reward given the relationships between EIRs. Evaluation of the global criterion at this level allows pruning the exploration tree if relationships between EIRs are not validated. Different pruning function can be used at line 23 depending on the application context (Chanthery et al. (2005)). Different sorting functions can also be used at line 24.

4. ILLUSTRATIVE SCENARIO

The applicative context of this work is the management of a fire by a team of autonomous and collaborating robots. There are several types of data collections that have to be managed. They correspond to different stages of the management of a fire. The first step is *fire Detection*. Fire evidence may be acquired by different means: on-site forest detection, satellite images, human observations and others. Observations and alarms are gathered into a control center where they are analyzed. The second step is *Recognition*. Once a fire has been detected, the situation must be assessed. This step needs to collect enough information in order to decide on the appropriate response. The third step is *on site Intervention*. It includes localization of people in possible danger, advice and guidance for moving to a safe location, location of injured persons, providing primary help and transportation to safe place. It also includes localization of the area of the fire to be extinguished, monitoring and localization of team members, monitoring of the status of the fire and estimation of the progress of the fire. *Evaluation reports* should then be elaborated to summarize the conducted activities, the challenges, strengths and weaknesses of plans, procedures, and protocols, efficiency on resource utilization, and assessment of mission results. These reports have to be taken into account for improving achievement of future missions. In order to produce these reports, data collection has to be carried out. An example of a *Recognition* step is presented on Fig. 6 which is used as a proof of concept of our developments. A 2D view of the activity area is presented. The proposed planning algorithm allows however for working in 3D but we focus in this paper on multi-ICS

Begin

1 <u>Initialization</u> : Put V_1 in *ListP*

2 **While** *ListP is not empty*

3 Process $\hat{U}$:

4 **For each** $\hat{u}$ in $ListN(\hat{U})$ **Do**

5 **For each** n_i in $S(\hat{u})$ **Do**

6 $listeT = \varnothing$

7 **For each allowed action** a **Do**

8 Build the sequence from n_1 to n_i^a

9 Calculate I by choosing t for each node within the constraints

10 **If there is a solution then**

11 Calculate $h_{n_i^a}$ from n_i^a to an end node

12 Store in n_i^a : I and $h_{n_i^a}$

13 Put n_i^a in $ListT$

14 **For each** n_i in $ListT$ **Do**

15 $ListS = \varnothing$

16 Create a supernode V_j from $\hat{U}$ and n_i

17 Calculate F by checking the logical and temporal contraints between EIR

18 **If there is a solution then**

19 Calculate H from V_j to an end supernode

20 Store in V_j : F and H

21 Put V_j in $ListS$

22 **If** V_j is an end supernode and $F < BOUND$ **then** $BOUND \leftarrow F$

23 Prune exploration tree

24 Sort successors of $\hat{U}$ in $ListP$: Put elements from $ListS$ to $ListP$

25 Put $\hat{U}$ in $ListQ$

26 Remove $\hat{U}$ from $ListP$

End

Fig. 5. Search algorithm

management and the relationship between FIRs. Shaded areas correspond to the FIR zones. A time window, priorities (translated in reward or gain) and a mission type are associated with each of these FIRs. In this scenario EIRs correspond to “images” (photograph, radar, sound) of a zone. Notation $EIRq_{tx}$ states that the requested image in the q^{th} EIR is of type ”tx”. EIR relationships translating validation of FIRs (and reward generation) are shown on the bottom of Fig 6. Three ICSs are available. Dark squares indicate their starting/arrival bases noted respectively Po/Pf (starting and arrival base could be different in other scenarios except for ICS1). ICS1 is an aerial type and must follow a direct line trajectory and goes out activity zone. Its sensor is of type t1. If its trajectory goes over an objective zone, it can be seen entirely (sufficient height and field width). ICS1 can pass through the points represented by circles to meet t1 type EIRs (Px61 and Px71). ICS2 is an UAV that does scanning to cover a given zone. Its sensor is t2 type and can pass through the points represented by crosses to meet t2 type EIRs (noted Pe, Ps or Px). ICS3 is a fixed system. Its trajectory to view different successive zones consists in changing its orientation. It can be orientated towards points represented by squares to meet t3 type EIRs (noted Px).

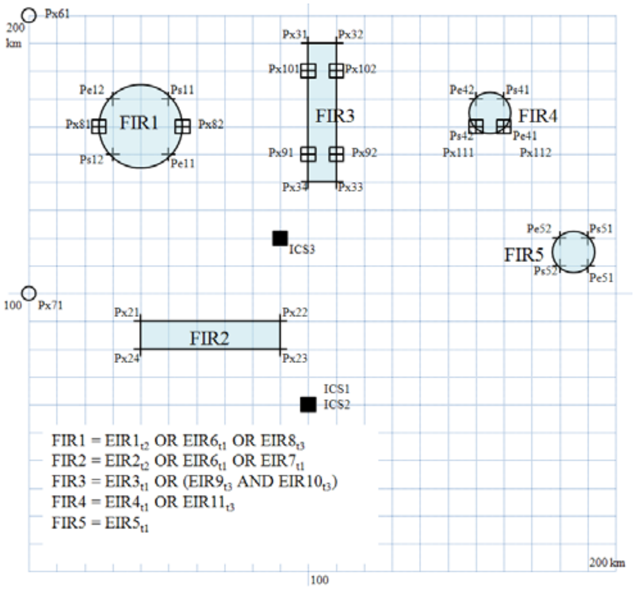

Fig. 6. 2D representation of a scenario.

The search algorithm can be adapted to different application contexts (different ICS type, data collection type …) by changing evaluation methods of nodes and supernodes. Different versions may also be defined according to the pruning tree method and sorting method of pending supernode list. In this example, the method for evaluating a node takes into account, according to the ICS associated to this node, autonomy, average speed, utilization costs, consumption costs and consumption function. The supernode evaluation includes rewards for satisfied FIRs and the sum of its node costs. Other physical models, more detailed and specific to each type of ICS, are currently under development. In the tested algorithm version, sorting method consists in classifying them by increasing F. The pruning tree method consists in summing F and G and compares it to the current bound. The estimation method of the path from a current supernode to an end supernode consists in summing rewards of non-satisfied FIRs. Other methods can however be defined to improve the search if possible.

A solution provided by the algorithm states a trajectory and its evaluation for each ICS. In this example with 5 FIRs and 3 ICSs, one solution found is: Po111-Px61 for ICS1, Po111-Pe51-Ps51-Pf111 for ICS2, Po311-Pe91-Ps92-Pe101-Ps102-Pe121-Ps122-Pf311 for ICS3. This solution is the best solution found for a given dataset. The time to find this solution is 0.4s and global execution time is 12 minutes. The number of developed nodes remains important but this aspect must be improved by developing others sorting and pruning heuristics. However, it is an anytime algorithm and a first good solution can be found rapidly.

5. CONCLUSION

This paper presents an approach to formalize and solve the problem of mission planning for Information Collection System (ICS). It is a broad problem that extends from the formalization of information needs, expressed as free text by an operator, to the optimization of ICS actions in order to respond to these needs knowing the various types of ICSs which can be used. A concept definition and global decision process have been proposed to state the utilization framework of the model presented after. It is a graph-based model structured into several levels. At the lowest level the trajectory and dynamics of each ICS is modeled. Satisfactions of elementary requests for each ICS are represented at an intermediate level. The highest level allows managing the coordination and assignment of ICSs. An A*- like search algorithm has been used to search the solution in this graph. The algorithm still needs to be improved in particular to limit the number of nodes explored. Various sorting and pruning algorithms should be tested. The problem resolution can also be decomposed into assignment algorithms at the coordination layer and single ICS algorithms at the bottom layer. Heuristics are being developed to be compared to the algorithm presented in this paper.

6. REFERENCES

Janez, F. (2007). Optimization for sensor planning, *Aerospace Science and Technology*, 11, 310–316.

Schumacher, C., Chandler, P.R., Rasmussen, S.R. (2002). Task Allocation For Wide Area Search Munitions, *Proc. of the 2002 American Control Conf.*, 3, 1917–1922.

Rasmussen, S.J., Chandler, P.R., Mitchell, J.W., Schumacher, C.J., Sparks, A.G. (2003). Optimal vs. Heuristic Assignment of Cooperative Autonomous Unmanned Air Vehicles. *Proc. of the AIAA Guidance, Navigation, and Control Conference, Austin, Texas 2003.*

Richards, A., Bellingham, J., Tillerson, M., How, J. (2002). Coordination and control of multiple UAVs. *AIAA Guidance, Navigation, and Control Conference and Exhibit, California, 2002*.

Cook, D.J., Gmytrasiewicz, P., Holder, L.B. (1996). Decision-theoretic cooperative sensor planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 18 (10), 1013–1023.

Chandler, P.R., Pachter, M., Rasmussen, S. (2001). UAV cooperative control. *Proc. of the American Control Conference,* 1, 50–55.

Vasquez, M., Hao, J.-K. (2001). A “Logic-Constrained” Knapsack Formulation and a Tabu Algorithm for the Daily Photograph Scheduling of an Earth Observation Satellite. *Computational Optimization and Applications*, 20, 137–157.

Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., Bataille, N. (2002). Selecting and scheduling observations of agile satellites, *Aerospace Science and Technology,* 6, 367–381.

Frank, J., and Kurklu, E. (2003). SOFIA’s choice: Scheduling observations for an airborne observatory. *In 13th Intern. Conf. on Automated Planning & Scheduling*.

Chanthery, E., Barbier M., Farges, J-L. (2005). Planning Algorithms for Autonomous Aerial Vehicle. *6th IFAC World Congress*

Chanthery, E. (2005). Mission planning for an unmanned aerial vehicle, PhD Thesis, Institut Supérieur de l’Aéronautique et l’Espace.