Controlability and Makespan Issues with Robot Action Planning and Execution

Matthieu Gallien
LAAS-CNRS*
7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4, France
{matthieu.gallien}@laas.fr

Introduction

Nowadays, many robotic applications need autonomous decision-making capabilities. Among them, some make intensive use of planning. Yet, planning is an activity whose algorithmic complexity is often incompatible with the reactivity requirement of an exploration rover or a space probe.

In past years, some planners have proven their ability to handle complex situations required by autonomous systems. Some of these systems (e.g. RAXPS [Jonsson *et al.* 2000], CASPER [Chien *et al.* 2005]) have been deployed.

The IxTeT planner[1] [Ghallab & Laruelle 1994] was developed to handle such robotic planning problems. It was extended to handle complex resources [Laborie & Ghallab 1995], continuous domains and constraints between both atemporal and temporal variables [Trinquart & Ghallab 2001]. Further work [Lemai 2004] added a temporal executive to IxTeT.

Reasoning about time is necessary to address these planning problems. The planner must be able to take into account strict deadlines, temporal windows for some tasks, durative actions, and durative goals. The STN[2] [Dechter, Meiri, & Pearl 1991] formalism is often used in temporal planning because the requests on these networks are solved very efficiently by polynomial algorithms. Nowadays, an extension to uncertain constraints has been studied and a polynomial algorithm [Morris, Muscettola, & Vidal 2001] has been proposed.

Actual robotic space exploration mission are very expensive, with a high requirement for quality scientific returns. During the MER mission, the use of MapGen has allowed a 25% increase of such returns [Rajan 2004]. In a fully autonomous planner, optimization can be made in two ways: finding directly one good plan or searching through the whole search space several plans to find the optimal one. Due to limited computational capacity, the second approach is often unreasonable. So we have to modify the planner to search for high quality solutions.

New issues were raised while experimenting with IxTeT new executive. Some are related to time aspects. We decided to experiment another time framework. Some are related to efficiency. We decided to make a different heuristic to solve this issue. We use a simulator of the robot to make intensive tests of solutions to the previous issues. During the tests, it became apparent that existing plan repair capabilities in IxTeT were in some cases unacceptably inefficient. In this paper, we describe a preliminary solution and some results commented. Future works will extend this work to try to make the plan repair mechanism complete and efficient.

Planning

IxTeT [Ghallab & Laruelle 1994] is a temporal constraint-based causal link planner using partially instantiated actions. Its planning algorithm is adapted from SNLP [McAllester & Rosenblitt 1991]. A time reified logic describes the evolution of state variables across the whole plan. IxTeT uses CSP techniques[3] to maintain the consistency of the plan's constraints. In particular, the planner uses a Simple Temporal Network [Dechter, Meiri, & Pearl 1991] to represent the temporal constraint.

Definition 1 *A* temporal assertion *on a state variable v is either an* event *or a* persistence condition *on v.*

Definition 2 *A plan $\mathcal{P}(S, \Phi, G, CA, F, T)$ is described by the state variables contained in S. Φ is a chronicle describing all the temporal assertions of the plan. F is the set of defaults in the plan. $CA \subset \Phi$ contains temporal assertions on variables of S describing the predicted evolution of contingent attributes. The goals are in $G \subset \Phi$, they are persistence conditions on state variables of S. T is the set of tasks in the plan.*

The planner begins with a plan describing the initial situation, the initial goals and the known predicted evolutions of contingent attributes such as visibility windows. The search is performed until the plan contains no default. These defaults are temporal assertions unexplained in the current plan[4], conflicts between two temporal assertions or possible resource conflicts. At each search step, a default is chosen according to a given heuristic. One of the resolvants of this

---

[*] Part of this work has been funded by a grant from the ESF (European Social Fund)

[1] IxTeT is a system used for chronicle recognition, planning and temporal execution.

[2] STN: Simple Temporal Network

[3] Constraint Satisfaction Problem [Mackworth 1977] (CSP)

[4] A temporal assertion is not explained by a plan if it is not an initial condition or if no causal link establishes the assertion.

default is then chosen and applied. The planner need only to backtrack on resolvant choices and not on default choices to be complete.

IxTeT uses a least commitment heuristic to evaluate a cost for each resolvant of each default. Then, a notation Opp(ρ) is computed for each one (see this work [Lemai 2004] for more details). The basic idea is to minimize the size of the search space and to ease the choice between the resolvants of one default.

In order to make plans with a shorter makespan, we make a new heuristic. We have implemented it by modifying two costs of the old heuristic. The first considers one single ordering resolvant. The new cost depends on the earlier date of the first timepoint instead of the commitment of the resolvant. It makes the planner produces plans with a shorter makespan. It is completed by the second modification. This cost evaluates one causal link. Instead of using the maximum duration for computing the commitment, we now use the minimum duration. The idea is that the planner makes shorter links and thus makes shorter plans. This heuristic is called makespan minimizing heuristic.

Underlying CSPs

IxTeT uses classical CSPs algorithms for managing constraints on atemporal variables. It uses an STN for managing all the temporal constraints, and a general arc-consistency filtering algorithm for managing symbolic and numeric constraints.

In some cases, we want to link the effects of a task to its duration. For example, if the consumption of a resource depends on the duration of the task, you need a mixed constraint between temporal and atemporal variables like a navigation duration depending on the navigation length and speed of the robot. IxTeT features a mechanism to propagate these constraints [Trinquart & Ghallab 2001].

On the STN, IxTeT needs the minimal graph to be computed. This comes from the number of requests that is much higher than constraint updates. It uses a path consistency algorithm like PC-2. An incremental version (only for constraint addition) is used during planning with a complexity of $O(n^2)$. For a constraint relaxation, the complete one is used in $O(n^3)$.

During execution, IxTeT updates the plan for example at each start or end of task. The CSP framework allows it to do this. A special care is taken to always kept the STN complete and minimal during execution. In fact the resource conflict detection, the plan repair mechanism and the propagation of mixed constraints need a complete graph. So the executive does not use a local temporal propagation like this work [Muscettola, Morris, & Tsamardinos 1998]. The atemporal CSP is only kept arc-consistent for computational reason and because the system can repair or replan.

Simple Temporal Network with Uncertainties

Definition 3 *An STNU [Vidal & Fargier 1999]* $\Theta = (V, D, C_{clb}, C_{ctg})$ *with* V *the set of variables,* D *the set of domains. All constraints are in the form* $lb \leq v_i - v_j \leq ub$*. The set* C_{clb} *is all the controllable constraints equivalent to STN constraints.* C_{ctg} *is a set of contingent constraints. The duration of these constraints can only be observed.*

The introduction of a new type of constraint changes the consistency notion inherited from the STN. Three main levels of controllability have been defined [Vidal & Fargier 1999]. In IxTeT, we use the dynamic controllability. An STNU is dynamically controllable if the execution controller must take decisions knowing only the past observations and timepoint instantiations. The 3DC+ algorithm [Morris, Muscettola, & Vidal 2001] is known to establish it in polynomial time. The result is similar to STN's result (i.e. the minimal network). It introduces a new ternary constraint type called “wait" necessary to safely execute the STNU.

We have made two little improvements to this algorithm. The first concerns the fact that the STNU in IxTeT are dynamic one (i.e. constraints and variables are added during planning). Before any constraint addition, we remove all existing “waits”. The second one replaces the complete algorithm uses to keep the STNU minimal during 3DC+ run by the same incremental one used on STN.

Execution

IxTeT's executive runs an execution cycle corresponding classically to a scheme “sense/plan/act". The executive begins with an initial plan produced by the planner.

All executable timepoints[5] are started as soon as possible except the end of some actions labelled as “late preemptible" or “not preemptible".

The executive receives task reports, new goals or resource capacity changes. It has to check the validity of the task reports considering the current plan. If the report is not nominal, the system integrates the report, thus partially invalidate the current plan and triggers a plan repair if possible. All causal links possibly in conflict with new inserted tasks are removed during the relaxation. The execution can continue interleaved with the plan repair. If the failed plan does not anymore support the running tasks, all tasks are interrupted and a complete replanning made. The new goals and resource capacity changes are integrated in the same manner.

Simulation and Results

IxTeT runs on the robot Dala and on a simulator of this robot. The simulator allows us to perform accurate tests of the different IxTeT strategies presented in the paper. The environment and the initial conditions can be exactly the same between runs.

We illustrate our contributions with an exploration rover like mission. The robot must acquire scientific data from several places. During its mission, it can communicate with an orbiter during visibility windows.

IxTeT now features two different planning heuristics and two different time management systems. This defines four IxTeT instantiations and we compare their performances using the simulator and the robot.

---

[5] IxTeT currently executes only a subset of the plan's timepoints: start and end of actions, goal and contingent timepoints.

During some missions with an STN, the system makes a very bad thing. Due to multiple faillures, the system repaired many times the plan. The makespan was the maximum allowed duration for the mission. In that case, the system must cancel unachievable goals according to their priority. The system does not make that but due to STN propagations decided to keep a low priority goal instead of cancelling it. The execution of the tasks to satisfy the low priority goal make impossible to satisfy other goals with higher priority. The system makes exactly what we do not want. Thanks to STNU, it may be impossible because uncertain durations are never reduced.

Using an STN and the makespan minimizing heuristic can make up to 30% shorter plan. With an STNU, the value is approximatively 15%. During execution of the mission and depending on the world, the mission duration can increase of 15% removing the advantage of the new heuristic.

Results show that the combination of an STN and the makespan minimizing heuristic makes plans very unstable and breakable in the most times and sometimes make a very good and shorter execution. In general, the correct execution of the mission highly depends on the uncertainties. The new heuristic gives good results for the initial plan with STN or STNU, but if some plan repairs are made during execution the quality decreases significantly. The STNU makes stable and robust plans. Thanks to this, the whole mission is executed in a more reliable way.

Improve the Plan Repair Mechanism

We identify a drawback of the current plan repair process of IxTeT during our tests. Sometimes, a repaired plan contains unnecessary tasks leading to a suboptimal plan. For example, during our tests, we add new “take picture" goals. The planner produces a plan resulting in navigation from an existing waypoint to a new goal location and back from the new goal to the old one. This may lead to a very low quality plan.

This situation arises when the planning decision taken to satisfy a new goal make the old tasks not supported by the plan. So new tasks are inserted to support these tasks. In fact the set of tasks added to restore the state variable to their values before the new goals may be unnecessary, for example the navigation tasks. A better way is to relax the existing tasks so that they may be adapted to the new plan.

A preliminary solution

The example is a mission with initially 5 “take picture" goals and 2 communication goals. One “take picture" goal is added during the first communication. The initial plan is found in 1.7s. The simulator runs on a Pentium4 at 3GHz.

The problem comes from a limited relaxation of the plan before the plan repair process. The plan repair solution described in the precedent sections of this paper, removes only causal links. A POCL planner using partially instantiated tasks, adds constraints on variables to make causal links valid. If these constraints remain after the removal of the link, the plan repair may produce a suboptimal plan. The solution is to remove the constraints at the same time than the link.

We integrated the algorithms described in [Surynek & Barták 2004]. We adapt it to continuous domains and use it to manage the filtering in the atemporal CSP. This permits to remove the atemporal constraints supporting a causal link. The current implementation does not remove temporal constraints because of the very small benefit. In fact, the number of temporal constraints added during resolution of conflicts between temporal assertions is much higher than constraints added with causal links. The relaxation of causal links' temporal constraints does not relax significantly the plan contrary to atemporal constraints.

Our proposition does not anymore remove all removable causal links. The ones added with new tasks, are no more removed to keep as much as possible satisfiability decisions.

The planner finds a solution containing only necessary tasks or navigations. This solution is yet limited to simple cases where actions partial order allows the planner to find a new solution. The duration is rather similar than for the initial planning. We have not yet made a complete comparison with replanning from scratch. From initial tests, the answer is that replanning may be faster to find a new plan but must interrupt all running tasks. In our test, the rover is navigating to its next goal and interleaving it with the plan repair resulting in zero delay for mission execution. The replanning, will introduce a delay before the navigation can be made. Yet a comparison of the duration of the plans produced by either repair either replannning has not been done.

Ongoing and Future Works

The recording of constraints associated to causal links permits to remove only some constraints before a plan repair. We will try to generalize this idea to record more explanations inside the plan. In fact, we want to be able to change task ordering when doing plan repair. By recording not only feasibility decisions but also the satisfiability decisions, we may be able to do that in the same way than the precedent work.

A promising way of research is to be able to explain why a task is in the plan and why it is in a specific time window. Using such explanation, one will be able to make local change on a plan in order to repair or improve it.

Any of this research way may invalidate some hypothesis of the executive and may need to review all them in order to be able to use new repair capabilities. Clearly if one want to use plan repair in an IxTeT like system, it must be globally interesting for the overall mission even if plan repair is longer to find a plan or if the duration of a repaired plan is greater than the one of a new plan.

We need also to improve the propagation of temporal constraint removals. In fact, this is the longest operation made during an execution cycle. The maximum duration of an execution cycle influences the task models and the reactivity to exogenous events. To safely execute the plan, the value must be enough to always be greater than the real duration.

Conclusion

We have describe a temporal planner and executive whose plan execution raises new issues. The first one is to deal with uncontrollable durations. We use a temporal framework with explicit uncertainties. The second one is the bad quality of the plans when compared with a duration optimal plan. We modify the search control of the planner to find better plans by modifying the planning heuristic.

A simulation architecture is used to evaluate the two solutions. During the test, the heuristic has shown a good robustness. Yet, an identified drawback limits the performance of this work. A solution using the plan repair ability is described in the last part of the paper.

The integration of an STNU shows that it is usable on a rover. It shows a better robustness of the mission execution. If one goal is achievable, with 3DC+, it is executed.

We see that ongoing work improves the plan repair mechanism but this work is limited to only some case and may take more time than a complete replanning. Ongoing work is made to evaluate the opportunity of using new relaxation methods before a plan repair and to extend the relaxation.

References

Chien, S.; Tran, D.; Rabideau, G.; Cichy, B.; Davies, A.; Sherwood, R.; Castano, R.; Mandl, D.; Frye, S.; Trout, B.; D'Agostino, J.; Shulman, S.; and Boyer, D. 2005. The autonomous sciencecraft on earth observing one. In *i-SAIRAS-2005*.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal Constraint Network. *Artificial Intelligence* 49(1-3):61–95.

Ghallab, M., and Laruelle, H. 1994. Representation and Control in Ixtet, a Temporal Planner. In *AIPS*, 61–67.

Jonsson, A. K.; Morris, P. H.; Muscettola, N.; Rajan, K.; and Smith, B. D. 2000. Planning in Interplanetary Space: Theory and Practice. In *Artificial Intelligence Planning Systems*, 177–186.

Laborie, P., and Ghallab, M. 1995. Planning with sharable resource constraints. In *IJCAI*.

Lemai, S. 2004. *IxTeT-eXeC : planning, plan repair and execution control with time and resource management*. Ph.D. Dissertation, LAAS-CNRS and Institut National Polytechnique de Toulouse, France.

Mackworth, A. 1977. Consistency in networks of relations. *Artificial Intelligence* 8:99–118.

McAllester, D. A., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *AAAI*, 634–639.

Morris, P. H.; Muscettola, N.; and Vidal, T. 2001. Dynamic control of plans with temporal uncertainty. In *IJCAI*.

Muscettola, N.; Morris, P.; and Tsamardinos, I. 1998. Reformulating temporal plans for efficient execution. In *Principles of Knowledge Representation and Reasoning*.

Rajan, K. 2004. Invited talk: Mapgen. In *IWPSS 2004, 4th International Workshop on Planning and Scheduling for Space, June 23 - 25*.

Surynek, P., and Barták, S. 2004. A New Algorithm for Maintaining Arc Consistency After Constraint Retraction. In *Principles and Practice of Constraint Programming*.

Trinquart, R., and Ghallab, M. 2001. An extended functional representation in temporal planning : towards continuous change. In *ECP*.

Vidal, T., and Fargier, H. 1999. Handling contingency in temporal constraint networks: from consistency to controlabilities. *JETAI* 11(1):23–45.