Strategic Plannning for Autonomous Systems over Long Horizons

Michael Cashmore, Maria Fox, Derek Long, Daniele Magazzeni, and Bram Ridder
Kings College London, London, WC2R 2LS
firstname.lastname@kcl.ac.uk, bram.ridder@gmail.com

Abstract

Planning plays a role in achieving long-term behaviour (persistent autonomy) without human intervention. Such behaviour engenders plans which are expected to last over many hours, or even days. Such a problem is too large for current planners to solve as a single planning problem, but is well-suited to decomposition and abstraction planning techniques. We present a novel approach to bottom-up decomposition into a two-layer hierarchical structure, which dynamically constructs planning problems at the abstract layer of the hierarchy using solution plans from the lower layer.

We evaluate this approach in the context of persistent autonomy in autonomous underwater vehicles, showing that compared to strictly top-down approaches the bottom-up approach leads to more robust solution plans of higher quality.

1 Introduction

This paper introduces a novel technique for planning in the context of persistent autonomous systems within tight deadlines and energy constraints. Persistent autonomy entails planning long-term behaviour for one or more autonomous vehicles achieving purposeful and directed activity over hours, days, or even weeks without human intervention. This includes many challenges, including robust execution, detection of errors and recovery (Faria et al. 2014; Cashmore et al. 2014). However, there is a challenge that precedes execution: generating plans for missions that extend over hundreds or thousands of actions, within hours or days of activity.

We show in Section 5 that such a problem is too large for current planners to solve as a single planning problem, but is well-suited to decomposition and abstraction planning techniques. Decomposition into a hierarchical structure is exploited by HTN planners (Erol, Hendler, and Nau 1994; Nau, Ghallab, and Traverso 2015), which rely on a top-down approach, exploiting pre-constructed plans to tackle separate component elements of the hierarchy. In contrast we propose dynamically decomposing the problem in two layers using a bottom-up approach. The tactical layer at which task plans are constructed from the original actions in the domain, and the strategic layer, which encapsulated actions that represent the completion of a task. The original problem is decomposed into disjunct tasks using a clustering algorithm, each task is planned for independently at the tactical layer and forms a pre-constructed plan. Then a problem is composed at the strategic layer, to find an execution order of the pre-constructed plans that satisfies the original planning problem.

We explore this approach in the context of the FP7 project PANDORA, managing a fleet of Autonomous Underwater Vehicles (AUVs). These vehicles are tasked with maintaining a seabed facility unsupervised. The structures on the seabed must be inspected on a regular basis. The AUVs must interact with control panels within set time windows to manage the site within time and resource constraints, refueling autonomously.

Decomposing a task, in general, is not trivial. Planners like SGPlan (Chen, Wah, and wei Hsu 2006) and REALPlan (Srivastava 2000) have explored this in the past with mixed success. The latter decomposes a plan based on the number of resources available, e.g. it crease a different plan for each available robot. In contrast, the decomposition approach used in the PANDORA project is based on "locality". The observation we exploit is that many of these long-term autonomy missions involve located executives interacting with their environment to achieve goals. These goals can be clustered, geographically and temporally, into a set of discrete tasks. A task can be associated with the area in which operations will be performed to accomplish its goals.

The key difference in our approach is in the construction of the strategic model. Similar to an HTN model, the strategic level contains macro actions that encapsulate plans at the tactical level. However, these plans are not constructed top-down, but automatically generated by a planner, bottom-up. A planner is used to construct plans for each task. The expected time and resource requirements to complete the task are taken automatically from the plan. These values are used as costs for the corresponding strategic action encapsulating this task. The strategic problem can then be constructed using these abstracted task-actions.

The paper is organized as follows. In Section 2 we give an overview of the relevant background of persistent autonomy and planning for long horizons. In Section 3 we describe our decomposition and abstraction technique in more detail. Then, in Section 4 we describe how a strategic mission plan can be executed, and some efficiency that can be gained. Finally, in Section 5 we describe the evaluation method for testing our framework and include the results of our evalua-

tion.

2 Related Work

Integrating planning systems with robotic systems for onboard planning in long-term missions is not a new concept, for example NASA's EUROPA framework was used for the EO-1 mission (Sherwood et al. 2006). Related to AUVs specifically a planning system was developed in cooperation with MBari to track algae blooms (Fox, Long, and Magazzeni 2012), within T-REX (Teleo-Reactive EXecutive) for reasoning onboard AUVs (McGann et al. 2008b), and using ROSPlan to integrate a planner with the COLA2 control architecture for AUVs in subsea intervention tasks (Cashmore et al. 2015; Palomeras et al. 2012).

We take the ideas introduced in these works and approach the challenge of *persistent autonomy*: missions that require robust planning and execution, with horizons of days, or even weeks. Alternative strategies for long term autonomy typically focus on execution monitoring or onboard replanning (eg (Smith, Rajan, and Muscettola 1997; McGann et al. 2008b; 2008a; Cashmore et al. 2015)). Nevertheless, these approaches are founded on the same ambition for long term autonomous behaviour and recognise the role of planning in achieving it. The contribution of this paper is in the formulation of a decomposed planning problem, through a bottom-up decomposition approach.

3 Bottom-Up Top-Down Strategic Missions

In this section we formalise the planning problem and the decomposition of it. We use PDDL2.1 (Fox and Long 2003) to describe our example domain and problems. In general the approach is not tied to choice of description language.

Definition 1. *PDDL2.1 Planning Problem. A PDDL2.1 planning problem is the tuple* $\Pi := \{P, V, A, T_p, T_v, I, G\}$*, where* P *is a set of propositions;* V *is a vector of real variables, called fluents; both are manipulated by* A*, a set of durative and instantaneous actions.* $I(P, V)$ *is a fuction over* $P \cup V$ *which describes the initial state of the problem. Similarly* $G(P, V)$ *is a function that describes the goal condition.* T_p *is the set of time initial literals (TILs). A TIL* (t, p) *describes that proposition* p *becomes true at time* t*.* T_v *is the set of time initial fluents (TIFs). A TIF* (t, v, x) *describes that fluent* v *is assigned the value* $x \in \mathbb{R}$ *at time* t*.*

A durative action a is described as a tuple: $a := \{pre_a, eff_a, dur_a\}$ where pre_a represents the action's preconditions – conditions that must hold for the action to be applied – eff_a represents the action's effects, and dur_a is a duration constraint, a conjunction of numeric constraints corresponding to the duration of the action a.

A single condition is either a single proposition $p \in P$, or a numeric constraint over V. A precondition is a conjunction of zero or more conditions. Each durative action A has three subsets of preconditions: $pre_{\vdash a}, pre_{\leftrightarrow a}, pre_{\dashv a} \in pre_a$. These represent the conditions that must hold at its start, throughout its execution, and at the end of the action, respectively.

Action effects are described by $eff^+_{\vdash a}$, $eff^-_{\vdash a}$, effects which add and remove propositions at the start of the action, respectively. Similarly $eff^+_{\dashv a}$, $eff^-_{\dashv a}$ add and remove propositions at the end of the action. Numeric effects $eff^{num}_{\vdash a}$, $eff^{num}_{\dashv a}$ assign values to fluents at the start and end of the action. Finally, continuous numeric effects $eff_{\leftrightarrow a}$ describe continuous change throughout the action's duration. (Fox and Long 2003).

A solution to a planning problem is a sequence of actions and timestamps $\langle (a_0, t_0), (a_1, t_1) \ldots, (a_n, t_n) \rangle$ applicable in I for which the resultant state S' satisfies the goal: $S' := G$.

A planning problem can contain fluents which are *resources* (Coles et al. 2014), we define resources as follows:

Definition 2. *Resource. Given a planning problem* Π *a resource is defined as a numerical fluent* $v \in V$ *whose value can be altered by the effect of an action* $a \in A$ *and also appears in the precondition of an action or is contained by a goal. A special resource is time as it can be constrained by TILs in the domain.*

An example of a resource in a logistical domain might be fuel, which can altered by consuming it or produced by refuelling.

Decomposition into Tasks

Given a planning problem Π we search for a decomposition that separates the goal G into a set of *tasks* T where each task $task \in T$ is $task \subseteq G$. For each $task$ we construct new planning problem $\Pi' = \Pi$, where $\Pi'_G = task$. At the tactical planning stage we do not know how many resources are available for each task, so we remove any constraints on these. These constraints are imposed on the planner at the strategic level.

We define a task's initial state to be a subset $I' \subseteq P$ describing the requirements for beginning the task. These requirements for beginning each task must be carefully chosen, we need to make sure that these requirements are reachable after the execution of any other task. In our work we rely on a domain expert to assign an initial state to each task. For example, in the PANDORA domain we identify, for each task, a location that is close to the area within which that task will be performed, $L(task)$. We call this the *jump off* point for $task$. The initial state I' of each task is that the AUV is located at the jump off point. These jump off points are located above the seabed structures, which makes them easily reachable. The task's initial state is used in the generation of the strategic problem, discussed in the next section.

Each problem Π' is passed to a planner, and a plan found. The solution plan $plan(\Pi')$ and its resource requirements are saved.

Whilst in general many decompositions are possible, it is sensible to decompose a problem such that related goals are combined in a single task. Decomposition of goals can be computed using a clustering algorithm or can be hand coded based on expert information of the domain. In PANDORA we exploit the fact that the domain is already separated into multiple seabed structures. While part of the same seabed facility, the structures are separated spatially, and are a simple way to split the goals into sets based on location. Moreover, control panels can be interacted with only within certain

time windows. If we create a task containing a goal with an associated time window with any other goal, then we might impose artificial constraints on the strategic level, leading to an unsolvable strategic problem. For example, consider the goals g_1, g_2, and g_3 that have non-overlapping time windows, such that g_1 needs to be achieved before g_2 and g_2 needs to be achieved before g_3. If we create a task that combines g_1 and g_3 and another task that contains g_2 then we render the planning problem unsolvable. For this reason we put those goals that are dependent on deadlines in separate tasks.

Generating a Strategic Problem

In order to generate the strategic problem, first an estimate of the resource use for each task must be computed. These estimates are computed from the tactical plans $plan(\Pi')$. The strategic domain and problem describe the execution of a task in an abstracted way, similar to HTN planning, by encapsulating the task plan as a single action. Therefore, the $plan(\Pi')$ is encapsulated in an abstract action a_{task} for which:

- pre_a models the bounds on the resources used by $plan(\Pi')$, and the initial state I'.
- eff_a^{num} describes the change in resource over $plan(\Pi')$.
- $eff_a^+ \cup eff_a^-$ describes the change in distinct variables over $plan(\Pi')$.
- dur_a is the duration of $plan(\Pi')$.

The domain and an example problem file from PANDORA are shown in figures 1 and 2 respectively. The *complete_mission* actions correspond to the tactical plans for a single task. The duration of these actions is determined by the function *mission_duration*, which is defined in the initial state. These mission durations (along with any other possible resource requirements) are set in the initial state to be equal to the duration of the corresponding task plan. For example, in figure 2 the duration of each mission is assigned in the initial state, eg:

```
(= (mission_duration Mission10) 117.739)
```

As a precondition of the *complete_mission* action, the executive vehicle must be at the mission jump-off location $L(T)$ for the task t, and there must still be enough resource and time available to achieve the task completely.

4 Efficient Execution of Strategic Missions

There are some improvements we can add in the way that the strategic plan is executed. It is possible to simply replace the strategic *complete_mission* actions with the tactical plans they represent. However, a more efficient plan can be found by replanning the tactical problem during execution of the strategic plan. Example strategic and tactical plans are shown in figures 3 and 4.

There are several reasons to replan the tactical problem before its encapsulating *complete_mission* action is executed in the strategic plan.

1. depending on the execution of previous tactical plans, the current amount of available resource might differ from what was expected to be available;

```
(define (domain strategic)
(:requirements ...)
(:types waypoint mission vehicle)

(:predicates
  (connected ?wp1 ?wp2 - waypoint)
  (at ?v - vehicle ?wp - waypoint)
  (vehicle_free ?v - vehicle)
  (in ?m - mission ?wp - waypoint)
  (completed ?m - mission)
  (active ?m - mission)
  ...
)

(:functions
  (distance ?wp1 ?wp2 - waypoint)
  (mission_duration ?m - mission)
  (charge ?v - vehicle)
  (mission_total)
)

(:durative-action complete_mission
  :parameters (?v - vehicle ?m - mission ?wp - waypoint)
  :duration ( = ?duration (mission_duration ?m))
  :condition (and
    (over all (vehicle_free ?v))
    (over all (active ?m))
    (at start (in ?m ?wp))
    (at start (at ?v ?wp))
    (at start (>= (charge ?v) (mission_duration ?m)))
  )
  :effect (and
    (at start (not (at ?v ?wp)))
    (at end (increase (mission_total) 1))
    (at end (decrease (charge ?v) (mission_duration ?m)))
    (at end (completed ?m))
    (at end (at ?v ?wp))
  )
)

(:durative-action do_hover ...
(:durative-action dock_auv ...
(:durative-action recharge ...
(:durative-action undock_auv ...
))
```

Figure 1: A fragment of a strategic domain. The body of some domain-specific operators is ommitted for space. The *complete_mission* operator corresponds to the tactical plan of a task.

2. similarly, the *complete mission* (strategic) action might be dispatched earlier or later than was anticipated, which might have knock-on effects on deadlines in the tactical task;
3. depending on the direction of approach from the previous action of the (strategic) plan, the tactical plan might be planned differently to exploit better routes between elements of the task.

The execution of a tactical plan, including any tactical replanning, or rescheduling, is handled by an onboard executive, in our evaluation we use ROSPlan (Cashmore et al.

```
(define (problem strategic_mission)
(:domain strategic)
(:objects
  v - vehicle
  mission_site_start_point_0 wp_auv0
  ... - waypoint
  Mission0 Mission1 Mission10 Mission12
  Mission13 Mission14 Mission15
  ... - mission
)
(:init
  (vehicle_free v)
  (at auv wp_v0)
  (= (charge v) 1200)
  (= (mission_total) 0)

  (recharge_at mission_site_start_point_0)

  (active Mission0)
  (active Mission1)
  (active Mission10)
  (active Mission12)
  (active Mission13)
  (active Mission14)
  (active Mission15)
  ...
  (at 4100 (not (active Mission0)))
  (at 7100 (not (active Mission1)))
  (at 86400 (not (active Mission10)))
  (at 86400 (not (active Mission12)))
  (at 86400 (not (active Mission13)))
  (at 86400 (not (active Mission14)))
  (at 86400 (not (active Mission15)))
  ...
  (in Mission0 mission_site_start_point_1)
  (in Mission1 mission_site_start_point_1)
  (in Mission10 mission_site_start_point_1)
  (in Mission12 mission_site_start_point_1)
  (in Mission13 mission_site_start_point_1)
  (in Mission14 mission_site_start_point_1)
  (in Mission15 mission_site_start_point_1)
  ...
  (= (mission_duration Mission0) 261.868)
  (= (mission_duration Mission1) 242.065)
  (= (mission_duration Mission10) 117.739)
  (= (mission_duration Mission12) 154.668)
  (= (mission_duration Mission13) 157.892)
  (= (mission_duration Mission14) 151.502)
  (= (mission_duration Mission15) 135.29)
  ...
  (connected mission_site_start_point_0 wp_v0)
  (= (distance mission_site_start_point_0
      wp_v0) 56.7891)
  ...
)
(:metric maximize (mission_total))
(:goal (> (mission_total) 0))
)
```

Figure 2: A fragment of an example strategic problem.

2015). In our tactical domain, we use a conservative model of action duration and cost, so we expect that most tasks will

```
0.000: (do_hover auv wp_auv0 wp0)  [291.548]
291.548: (complete_mission auv mission9 wp0)  [194.639]
486.188: (complete_mission auv mission8 wp0)  [236.909]
723.098: (do_hover auv wp0 wp1)  [270.416]
993.516: (dock_auv auv wp1)  [20.000]
1013.516: (recharge auv wp1)  [1800.000]
2813.517: (undock_auv auv wp1)  [10.000]
2823.517: (do_hover auv wp1 wp0)  [270.416]
3093.934: (complete_mission auv mission11 wp0)  [284.545]
3378.481: (complete_mission auv mission10 wp0)  [293.488]
3671.970: (do_hover auv wp0 wp1)  [270.416]
3942.387: (dock_auv auv wp1)  [20.000]
3962.387: (recharge auv wp1)  [1800.000]
5762.388: (undock_auv auv wp1)  [10.000]
5772.388: (do_hover auv wp1 wp0)  [270.417]
6042.806: (complete_mission auv mission1 wp0)  [342.707]
6385.514: (do_hover auv wp0 wp1)  [270.417]
6655.931: (dock_auv auv wp1)  [20.000]
6675.931: (recharge auv wp1)  [1800.000]
8475.932: (undock_auv auv wp1)  [10.000]
8485.932: (do_hover auv wp1 wp0)  [270.416]
8756.350: (complete_mission auv mission0 wp0)  [381.766]
9138.117: (do_hover auv wp0 wp1)  [270.416]
9408.534: (dock_auv auv wp1)  [20.000]
9428.534: (recharge auv wp1)  [1800.000]
11228.535: (undock_auv auv wp1)  [10.000]
```

Figure 3: A strategic plan for the abstract level. The *complete_mission* action corresponds to the tactical plan of a task.

```
% 0.000: (do_hover auv wp0 wp1) [34.547]
% 34.548: (check_panel auv wp1 ip0) [10.000]
% 44.549: (correct_position auv wp1) [10.000]
% 55.208: (valve_state auv wp1 p0) [10.000]
% 70.000: (do_hover auv wp0 wp2) [9.921]
% 79.922: (turn_valve auv wp2 p0 v0) [30.000]
% 109.923: (correct_position auv wp2) [10.000]
% 149.924: (turn_valve auv wp2 p0 v1) [30.000]
% 179.925: (correct_position auv wp2) [10.000]
```

Figure 4: A tactical plan for a single task. This plan is generated to provide resource estimates in the construction of an abstract strategic problem, and encapsulated at that level in a *complete_mission* action.

be completed within the estimated time. However, when executing plans onboard a physical platform there is always the chance that actions might fail, take longer than expected, or be accomplished more quickly.

To take advantage of extra or diminished resource (1), or alteration in deadline (2), we replan a tactical task before it is executed. This process takes 10 seconds, and can be performed in parallel with other strategic actions.

When planning tactical tasks in the construction of the strategic problem, the task jump-off point $L(t)$ is used as the intial position of the executive. After the strategic plan has been generated, the initial position of the executive can be improved by considering the previous actions in the strategic plan. Thus, the executive no longer needs to visit the jump off point, but can move directly to the most convenient first

location in the execution of the task.

To ensure efficient linkage to the next task, the latest destination in the strategic plan can be used as the initial location for the executive in the task to be replanned. $L(t)$ will be ignored and replaced with the necessary connecting navigation actions to get between the latest destination in the strategic plan and the best entry point in the next task.

Finally, it is necessary to revalidate the strategic plan after the replanning or completion of any tactical task. It is possible that a tactical plan takes longer to complete than expected, and from the current time the strategic plan no longer respects the task deadlines. Similarly, a tactical plan might take more resource than accounted for by our conservative action model, and there is no longer enough resource to complete subsequent actions in the strategic plan.

To account for these points, we use the ROSPlan executive for dispatch of the strategic, as well as tactical plan. More generally, execution monitoring techniques developed for tactical plans can be used for the execution of plans at both levels in the hierarchy.

5 Results

We tested this approach in the context of the FP7 project PANDORA, managing a fleet of Autonomous Underwater Vehicles (AUVs). In order to do this, we have built an underwater environment simulation. The simulator possesses an in-built editor used to model missions with very long horizons, allowing us to experiment with multi-hour and multi-day missions, combining multiple tasks, such as inspection of a complex site (figure 5) and valve turning, under deadlines and resource constraints. The simulation provides uncertainty about action durations, creates unpredictable features such as marine life, simulates currents, and allows the AUV to discover new features in the environment.

We compare against a selection of top-down decompositions to show that the tactical information gathered by the bottom-up top-down (BUTD) approach leads to heigher quality, and more robust solutions. We compare against a purely tactical approach to demonstrate that the scenario is too large to solve as a single planning problem, and some heirarchical decomposition is a possible solution.

We used the BUTD decomposition approach to separate a set of missions into tasks, with results between 10 and 30 tasks. We then found strategic solutions to these problems. To compare, we used the same set of tasks in a top-down decomposition, in which the tactical missions had not been planned, and their estimated resource usage computed. As the top-down approach has not yet planned on the tactical level, another estimate of each task's resource requirements is required. We expect that some prior knowledge of resource use would be known, and so use the estimated resource usage computed from the BUTD decomposition as this a priori knowledge, and used it in four different estimates of task resource usage:

- *mean*: a naive strategy that takes the mean resource use over all tasks, and assigns this to all tasks;
- *conservative* a conservative strategy that assigns to all tasks the 80th percentile of resource use over all tasks;
- bucket-mean: the tasks were divided into sets of similar type – inspection, valve-turning, etc. – and size. Then, the mean resource use from each set was assigned as the estimated resource use for each task in that set;
- bucket-conservative: the tasks were divided in the same was as bucket-mean, but the 80th percentile from each set was used instead of the mean.

We used the planner POPF (Coles et al. 2010). POPF is a temporal and metric planner, which allows us to model the synchronisation aspects of our problems (including deadlines for interaction with valves), the constraints on energy over long missions, and optimise plans based on a metric function. The metric to be optimised was the number of tasks completed. POPF was given 1800 seconds and 8GB of RAM to find the best possible solution. In the BUTD approach POPF was given 10 seconds per tasks to perform tactical planning for each task. This reduced the amount of time given to solving the BUTD strategic problem.

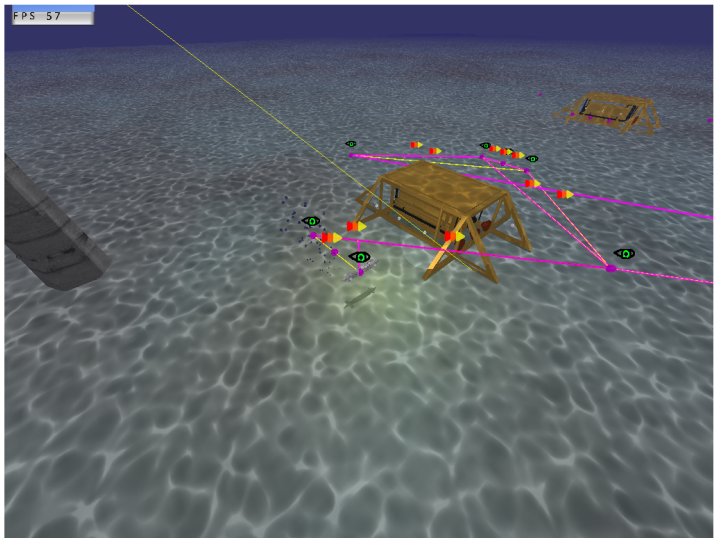
Due to the extra information from the tactical level, we expected that the strategic solutions generated using the BUTD approach would plan more efficient solutions in terms of number of tasks completed. Moreover, as the tasks had already been tactically planned, the resource usages of fewer tasks would have been underestimated, and the solutions are expected to be more robust.

Table 1 compares the number of tasks solved by BUTD and the various top-down strategies. As can be seen from the table, BUTD solves more tasks than any of the top-down strategies. Table 2 shows the number of tasks for which the resource usage is underestimated. In the strategic problems generated by the top-down strategies, these tasks have assigned resource estimates lower than that derived from planning a tactical solution. As a result, these tasks are very likely to run out of time and resource before successfully completing.

In the top-down approaches, the conservative approach performs the poorest, which is to be expected – the large uniform estimates for the resource usage of every task leads to a solution with many unnecessary refueling actions and missed deadlines. Both mean and bucket-mean strategies lead to higher quality solutions, but are not robust, underestimating the resource use of 32% and 27% of tasks respectively. The bucket-conservative approach performs best of the top-down appraches. The approach is the most robust, underestimating the resource requirements of only 14% of tasks, and generating the highest quality strategic solutions amongst all top-down approaches.

BUTD outperforms all top-down approaches, confriming our expectations. The variance in the resource usage between tasks of similar type and size is due to constraints that are only visible at the tactical level of the original domain, and as such are not available prior to planning the task itself. With this information, the BUTD strategic solutions are the most robust – none of the tasks are guaranteed to fail – and of the highest quality.

Table 3 shows the results for attempting to plan missions without decomposing into a tactical/strategic heirarchy. The number of tactical goals achieved by the best plan. It is clear



0.000: (correct_position auv0 wp_auv0) [3.000]
3.001: (do_hover_fast auv0 wp_auv0 strategic_location_7) [11.403]
14.405: (correct_position auv0 strategic_location_7) [3.000]
17.406: (observe_inspection_point auv0 strategic_location_7 inspection_point_22) [10.000]
27.407: (correct_position auv0 strategic_location_7) [3.000]
30.408: (do_hover_fast auv0 strategic_location_7 strategic_location_4) [11.673]
42.082: (correct_position auv0 strategic_location_4) [3.000]
45.083: (do_hover_controlled auv0 strategic_location_4 strategic_location_5) [4.000]
49.084: (observe_inspection_point auv0 strategic_location_5 inspection_point_20) [10.000]
59.085: (correct_position auv0 strategic_location_5) [3.000]
62.086: (do_hover_controlled auv0 strategic_location_5 strategic_location_6) [4.000]
66.087: (observe_inspection_point auv0 strategic_location_6 inspection_point_21) [10.000]
76.088: (correct_position auv0 strategic_location_6) [3.000]
79.089: (do_hover_fast auv0 strategic_location_6 strategic_location_4) [4.000]
83.090: (correct_position auv0 strategic_location_4) [3.000]
86.091: (observe_inspection_point auv0 strategic_location_4 inspection_point_19) [10.000]
96.092: (correct_position auv0 strategic_location_4) [3.000]
99.093: (do_hover_controlled auv0 strategic_location_4 strategic_location_5) [4.000]
103.094: (do_hover_fast auv0 strategic_location_5 strategic_location_8) [13.124]
116.219: (correct_position auv0 strategic_location_8) [3.000]
119.220: (observe_inspection_point auv0 strategic_location_8 inspection_point_23) [10.000]
129.221: (correct_position auv0 strategic_location_8) [3.000]
132.222: (do_hover_fast auv0 strategic_location_8 strategic_location_5) [13.124]
145.348: (correct_position auv0 strategic_location_5) [3.000]
148.349: (do_hover_controlled auv0 strategic_location_5 strategic_location_4) [4.000]
152.350: (do_hover_fast auv0 strategic_location_4 strategic_location_7) [11.673]
164.023: (correct_position auv0 strategic_location_7) [3.000]
167.024: (do_hover_fast auv0 strategic_location_7 wp_auv0) [11.403]
178.429: (correct_position auv0 wp_auv0) [3.000]
181.430: (do_hover_fast auv0 wp_auv0 mission_site_start_point_0) [40.307]

Figure 5: The simulation (left) and current plan (right) of a tactical mission for an inspection task.

that a purely tactical approach is insuffiecient for these sizes of problem.

6 Conclusion

We have presented an novel approach to the heirarchical decomposition a planning problem in the context of persistent autonomy. Our approach constructs an abstracted "strategic" layer of the heirarchy from solutions to the tactical tasks planned at the level of the original domain. The resulting problems can be solved and dispatched using a breadth of execution frameworks already available for executing plans onboard a robotic platform.

We briefly described the decomposition of a mission into a set of tasks, based on geographical and temporal clustering. This simple solution enables us to generate the problem heirarchy, but is not necessarily the best approach. In fact, any decomposition sacrifices the possibility or more optimal solutions that are formed from the interweaving of action in multiple tasks. Moreover, our spatial and temporal decomposition is specific to our domain. We propose to investigate the use of more general decompositions in future work.

In generating an strategic layer there is a trade-off when estimating the resource constraints of abstracted actions that encapsulate collections of lower-level behaviour. A more conservative approach is more robust, but over-estimation of resource use leads to less efficient plans that do not utilise all of the time and resource actually available. We show that it is possible to precompute much of this information in the automatic generation of the strategic layer, creating tighter bounds on resource use that remain robust.

We demonstrate these benefits, simulating long-term missions by autonomous underwater vehicles in a dynamic environment.

Tasks Completed					
number of tasks	BUTD	Top-down conservative	bucket-mean	mean	bucket-conservative
10	**10**	6	**10**	**10**	**10**
10	**10**	6	**10**	**10**	**10**
10	**10**	6	**10**	**10**	**10**
15	**15**	4	**15**	**15**	**15**
15	**15**	5	**15**	**15**	**15**
15	**15**	5	**15**	**15**	**15**
20	**20**	4	**20**	**20**	14
20	**20**	4	**20**	**20**	14
20	**20**	4	**20**	**20**	**20**
25	**24**	4	16	13	18
25	**23**	4	16	13	18
25	**25**	4	14	10	24
30	**25**	3	11	10	22
30	15	3	11	10	**22**
30	**25**	3	11	10	23

Table 1: Comparing BUTD and Top-down performance over long missions. The number of tasks completed in strategic missions of varying size.

Tasks Underestimated				
number of tasks	conser -vative	bucket mean	mean	bucket con -servative
10	1	5	4	2
10	0	3	2	1
10	2	3	2	3
15	1	7	6	2
15	0	3	3	1
15	4	5	4	5
20	1	9	8	2
20	0	5	4	1
20	4	5	4	5
25	1	11	10	2
25	0	6	5	1
25	6	7	6	7
30	1	13	12	2
30	0	7	6	1
30	6	7	6	7

Table 2: The quality of the different top-down representations, in terms of number of tasks given a duration that is shorter than the actual estimate of time required to complete the task. These tasks are likely to run out of time during execution.

Goals achieved		
goals	BUTD	Separate tasks
106	106	9
106	106	9
106	106	9
150	150	9
150	150	9
150	150	9
212	212	14
212	212	9
212	212	9
265	258	11
265	250	9
265	265	9
310	270	8
310	190	9
310	270	10

Table 3: Comparing the BUTD decomposition against planning for each goal separately. The number of goals achieved by the best plan.

References

Cashmore, M.; Fox, M.; Larkworthy, T.; Long, D.; and Magazzeni, D. 2014. Auv mission control via temporal planning. In *IEEE International Conference on Robotics and Automation (ICRA'14)*.

Cashmore, M.; Fox, M.; Long, D.; Magazzeni, D.; Ridder, B.; Carrera, A.; Palomeras, N.; Hurtos, N.; and Carreras, M. 2015. Rosplan: Planning in the robot operating system. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS'15)*.

Chen, Y.; Wah, B. W.; and wei Hsu, C. 2006. Temporal planning using subgoal partitioning and resolution in sgplan. *Journal of Artificial Intelligence Research* 26:369.

Coles, A.; Coles, A.; Fox, M.; and Long, D. 2010. Forward-chaining partial-order planning. In *Proceedings of the 20rd International Conference on Automated Planning and Scheduling (ICAPS'10)*, 42–49.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2014. A hybrid LP-RPG heuristic for modelling numeric resource flows in planning. *CoRR*.

Erol, K.; Hendler, J.; and Nau, D. S. 1994. Htn planning: Complexity and expressivity. In *AAAI*, volume 94, 1123–1128.

Faria, M.; Pinto, J.; Py, F.; Fortuna, J.; Dias, H.; Martins, R.; Leira, F.; Johansen, T. A.; de Sousa, J. B.; and Rajan, K. 2014. Coordinating uavs and auvs for oceanographic field experiments: Challenges and lessons learned. In *2014 IEEE International Conference on Robotics and Automation (ICRA'14)*.

Fox, M., and Long, D. 2003. PDDL2.1: An extension to pddl for expressing temporal planning domains. *Journal of Artificial Intelligence Res. (JAIR)* 20:61–124.

Fox, M.; Long, D.; and Magazzeni, D. 2012. Plan-based policy-learning for autonomous feature tracking. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS'12)*.

McGann, C.; Py, F.; Rajan, K.; Ryan, J. P.; and Henthorn, R. 2008a. Adaptive control for autonomous underwater vehicles. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence, (AAAI'08)*, 1319–1324.

McGann, C.; Py, F.; Rajan, K.; Thomas, H.; Henthorn, R.; and McEwen, R. S. 2008b. A deliberative architecture for AUV control. In *2008 IEEE International Conference on Robotics and Automation (ICRA'08)*, 1049–1054.

Nau, D. S.; Ghallab, M.; and Traverso, P. 2015. Blended planning and acting: Preliminary approach, research challenges. In *Proceedings 29th AAAI Conference on Artificial Intelligence (AAAI'15)*.

Palomeras, N.; El-Fakdi, A.; Carreras, M.; and Ridao, P. 2012. Cola2: A control architecture for auvs. *IEEE Journal of Oceanic Engineering* 37(4):695–716.

Sherwood, R.; Chien, S.; Tran, D.; Davies, A.; Castaño, R.; Rabideau, G.; Mandl, D.; Frye, S.; Shulman, S.; and Szwaczkowski, J. 2006. *Enhancing science and automating operations using onboard autonomy*. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration.

Smith, B. D.; Rajan, K.; and Muscettola, N. 1997. Knowledge acquisition for the onboard planner of an autonomous spacecraft. In *10th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW'97)*.

Srivastava, B. 2000. Realplan: Decoupling causal and resource reasoning in planning. In *AAAI/IAAI*, 812–818.