Planning and Acting in Deadline Situations

Sarit Kraus [2,1]
sarit@umiacs.umd.edu

Madhura Nirkhe [1]
madhura@cs.umd.edu,

Donald Perlis [1,2]
perlis@cs.umd.edu

Department of Computer Science[1]
and
Institute for Advanced Computer Studies [2]
University of Maryland
College Park, Maryland 20742

How to organize (plan) ones planning, especially in the presence of time constraints? Meta-planning, of course. But that takes time too! Maybe the time taken by meta-planning can be kept very short.[1] But what of highly novel settings in which one cannot a priori assign expected utilities to various conceivable options or refinements? Then the planner may have to decide on utilities and other factors in real time. In these cases it seems unlikely that such meta-planning will always have a modest time cost.

We have undertaken a project in combining declarative and procedural forms of real-time planning for novel situations. We began with the usual observation that in deadline [2] situations the time taken in reasoning toward a plan brings the deadline closer. Thus the planning mechanism should take account of the passage of time during that *same* reasoning. This general consideration is also the subject of other work[3] However, we are attempting to treat *all* facets of planning, for novel highly situations, as deadline-coupled; the problem then is how to take proper account of the approaching deadline when any such accounting itself simply takes more time and seemingly gets in the way of its own accuracy.

To elaborate, we present an illustrative domain, which we call the Nell and Dudley Scenario: Nell is tied to the railroad tracks as a train approaches. Dudley must formulate a plan to save her and carry it out before the train reaches her. If we suppose Dudley has never rescued anyone before, then he cannot rely on having any very useful assessment in advance, as to what is worth trying. He must deliberate (plan) in

[1] Indeed, this is stated in [?] (page 402): "Here we will not worry about the cost of meta-reasoning itself; in practice, we have been able to reduce it to an insignificant level", and in [?] (page 50): "The time required for deliberation scheduling will not be factored into the overall time allowed for deliberation. For the techniques we are concerned with, we will demonstrate that deliberation scheduling is simple, and, hence, if the number of predicted events is relatively small, the time required for deliberation can be considered negligible."

[2] We do not agree with the claim in [?] (page 401) that "the 'deadline' model of time pressures is overly restrictive, since in reality there is almost always a continuous increase in the cost of time." We think that many situations do involve relatively hard deadlines; e.g. getting to the airport in time, not to mention the more dramatic Nell and Dudley case below.

[3] In [?],[?] and [?], decision-theoretic approaches are used to optimize the value of computation under uncertain and varying resource limitations. In both works, dead-lines and the passage of time while reasoning are taken into consideration in computing the expected computational utility. Dean [?] proposed a computational approach to reasoning about events and their effects occurring over time. Dean, Firby and Miller [?] subsequently designed FORBIN, a planning architecture that supports hierarchical planning involving reasoning *about* deadlines, travel time and resources. Dean and Boddy [?] formulated an algorithmic approach to solution of time-dependent planning problems by introducing "anytime algorithms" which capture the notion that utility is a monotonic function of deliberation time. In [?] they demonstrated deliberation scheduling for a time-dependent planning problem involving tour and path planning for a mobile robot. We refer the reader to [?] for a survey of related work.

order to decide this, yet as he does so the train draws nearer to Nell. Thus he must also assess and adjust (meta-plan) his on-going deliberations vis-a-vis the passage of time. His total effort (plan, meta-plan, action) must stay within the deadline.

Drew McDermott [?] and Andrew Haas [?] have discussed the Nell and Dudley problem in terms of a surprising difficulty: if Dudley does not properly distinguish his planned actions from actual events then he may formulate a plan to save Nell and then conclude that his plan will save her, hence she is not in danger, hence he does not need the plan after all! This bizarre possibility can indeed arise in a highly limited representational setting, in which plans are not distinguished from actions.

However, although this is treated in our project, it is only a small part of the main thrust of our concern, which is *to find effective representational and inferential tools by which a reasoner can keep track of the passing of time as he makes (and enacts) his plan, thereby allowing him to adjust the plan so that neither the plan-formation that is in progress, nor its simultaneous or subsequent execution, will take him past the deadline.* In terms of our toy scenario, we want to prevent Dudley from spending so much time seeking a theoretically optimal plan to save Nell, that in the meantime the train has run Nell down. Moreover, we want Dudley to do this without much help in the form of expected utilities or other prior computation.

Our current project employs the formalism of "step logics" introduced by Elgot-Drapkin, Miller, and Perlis ([?], [?], [?]) in which inferences are parametrized by the time taken for their inference, and in which these time parameters themselves can play a role in the specification of the inference rules and axioms. In effect, step-logics are first-order logics modified to include a $Now(i)$ predicate, where the value of i changes at every execution of an inference rule.

Our progress on the project thus far: We have created a suitable representational language for a simple (e.g., we are not yet addressing interacting plans) version of the Nell and Dudley representative deadline problem. This version has been implemented. As Dudley develops a partial plan to save Nell, he continuously refines his estimate of the time to carry the plan to completion, making sure it will not overshoot the deadline.[4] Within the context of the plan, he makes predictions into the future and stretches his knowledge of the present as far as it will persist.[5] These two together form his dynamic set of projections which is used while extending and refining the plan. Dudley starts to act on the partially developed plan as soon as it is possible to perform a primitive action, not waiting for the plan to reach completion. At the same time he continues planning. Subsequent to performing a primitive action of the plan, he observes whether he succeeded or not, and this observation influences his continued execution of the plan. His predicted projections and observations are compared; conflicts resolved in favor of the latter. If he has no observations available, he acts on the basis of the former.

We are currently extending our implementation in various ways, to involve perceptual reasoning,[6] explicit representations for extended actions, revising plans when they are seen to be inadequate, and choosing between multiple plans.

To illustrate our efforts in a bit more detail, we present below portions of the output from our PROLOG program that implements the ideas we have been discussing. Here Nell is a distance of 35 'paces' from Dudley when he first realizes (step 0) the train will reach her in 50 time units (which we hereafter term 'seconds'). He begins to form a plan, seen below in step 1 as Ppl ('partial plan'), and refines the plan in subsequent steps. Ddl is the deadline time (50). d is Dudley, and n is Nell. The subscript obs indicates that the wff it is attached to is the result of an observation. Subscripted ls indicate locations and subscripted ts indicate times (step numbers). A colon between times (as in $At(0:\infty, d, l_1)$) represents a time interval during which the predicate is asserted to be true (in this case, that Dudley will be at location l_1 from 0 to infinity.

Wet stands for 'working estimate of time', and is Dudley's calculation of how long his partial plan (as he

[4]This "working estimate of time" is one of our concessions to procedural methods: we do not require Dudley to figure out how to do arithmetic but rather allow that he already knows. But we do require him to note the passage of time during the execution of the procedure.

[5]Thus he deduces supposed changes in the world, thereby revising some beliefs and retaining others: the familiar issues of the frame problem.

[6]This ties back to spatial reasoning, and to aspects of a plan that involve getting more information; for instance Dudley may have to move in order to see whether Nell is tied. This in turn relates to existing work ([?], [?]) on ignorance and perception.

has so far formed it) will take to execute. This he adds to the current time and compares the result to the deadline to make sure the plan is not hopeless. As long as it is not he declares it *Feasible*, and continues refining and/or putting it into execution. *Proj* gives Dudley's projections as to what will be true in the future, based on his partial plan and whatever *Facts* he has to work with. The word *save* that appears as argument to *Ppl*, *Proj* and *Feasible* in step 1, is simply a label naming the plan he is forming.

0. $\textbf{Facts}(\{At(0,d,l_1)_{obs}, Tied(0,n,l_2)_{obs}\})$, $\textbf{Deadline}(50)$, $\textbf{Goal}(out\_of\_danger(Ddl,n,l_2))$

1. $\textbf{Facts}(\{At(0,d,l_1), Tied(0,n,l_2)\})$, $\textbf{Deadline}(50)$,
   $\textbf{Unsolved}(Goal(out\_of\_danger(50,n,l_2)))$, $\textbf{Ppl}(save, 1, \{out\_of\_danger(50,n,l_2)))\})$,
   $\textbf{Proj}(save, \{At(0:\infty,d,l_1), Tied(0:\infty,n,l_2)\})$, $\textbf{Wet}(save,0)$, $\textbf{Feasible}(save,0)$

   In step 1 the *Ppl* simply records that Dudley plans to get Nell out of danger. In his *Proj* he still expects to remain where he is (l_1 for the indefinite future ('∞') since he has not yet realized in this first second that he must move to save Nell. Nor has he realized he must untie Nell, so he also projects that she will remain tied indefinitely.

2. $\textbf{Facts}(\{At(0:1,d,l_1), Tied(0:1,n,l_2)\})$, $\textbf{Deadline}(50)$, $\textbf{Unsolved}(Goal(out\_of\_danger(50,n,l_2)))$,

$$\textbf{Ppl}(save, 2, \left\{ \begin{bmatrix} Not\_tied(t_1,n,l_2) \\ Pull(t_1:t_2,d,n,l_2) \\ Out\_of\_danger(t_2,n,l_2) \end{bmatrix} \right\}, \{t_2 \leq 50, t_1 = t_2 - 1\},)$$

   $\textbf{Proj}(save, \{At(0:\infty,d,l_1), Tied(0:\infty,n,l_2)\})$, $\textbf{Wet}(save,0)$, $\textbf{Feasible}(save,1)$

   In step 2 Dudley has begun refining his plan, namely he determines that if Nell were untied then he could *Pull* her out of danger; this he infers from general world knowledge (axioms, not shown). The times t_1 and t_2 here are indefinite times that must satisfy only the conditions shown, so that the Wet is not too long. The column matrix indicates an action (*Pull*) with its enabling condition (Not_tied) and result (Out_of_danger). We skip the next three steps for the sake of brevity.

5. $\textbf{Facts}(\{At(0:4,d,l_1), Tied(0:4,n,l_2)\})$, $\textbf{Deadline}(50)$, $\textbf{Unsolved}(Goal(out\_of\_danger(50,n,l_2)))$,

$$\textbf{Ppl}(save, 5, \left\{ \begin{array}{cc} \begin{bmatrix} At(t_6,d,l_1) \\ Run(t_6:t_7,d,l_1:l_2) \\ At(t_7,d,l_2) \end{bmatrix}_1 & \begin{bmatrix} At(t_3,d,l_2) \\ Untie_1(t_3:t_9,d,n,l_2) \\ Succ\_u_1(t_9) \end{bmatrix}_{2\ldots} \\ \begin{bmatrix} At(t_5,d,l_2), Succ\_u_2(t_5) \\ Untie_3(t_5:t_4,d,n,l_2) \\ Succ\_u_3(t_4), Not\_tied(t_4,n,l_2) \end{bmatrix}_4 & \begin{bmatrix} Not\_tied(t_1,n,l_2) \\ Pull(t_1:t_2,d,n,l_2) \\ Out\_of\_danger(t_2,n,l_2) \end{bmatrix}_5 \end{array} \right\},$$

   $\{t_2 \leq 50, t_1 = t_2 - 1, t_4 \leq t_1, t_5 = t_4 - 1, t_3 = t_4 - 3, t_3 = t_9 - 1,$
   $t_7 \leq t_3, t_8 = t_7 - 1, t_6 < t_7\}),$

   $\textbf{Proj}(save, \{At(0:t_8,d,l_1), At(t_7:\infty,d,l_2), Tied(0:t_5,n,l_2), Not\_tied(t_4:\infty,n,l_2),$
   $Out\_of\_danger(t_2:\infty,n,l_2), Pull(t_1:t_2,d,n,l_2), Release(t_3:t_4,d,n,l_2), Run(t_6:t_7,d,L:l_2)\}),$
   $\textbf{Wet}(save,4), \textbf{Feasible}(save,4).$

   In step 5, Dudley has been able to infer (from axioms not shown) that he can refine his plan by running to Nell from l_1 (since he projects' from earlier steps that he will still be at l_1 at step t_6) to l_2 and releasing her (which will take him three untying actions). The numerical subscripts attached to column matrices show the order in which they are to be read (the list is linear but our formatting does not show this); also the subscripts show some portions have been omitted for ease of presentation. Note that the result Not_tied in the fourth matrix matches the enabling condition of the fifth matrix. Notice in *Proj* at last Dudley knows he must move to l_2 (by some as yet indefinite time t_7, where he then supposes he will remain.

6. $\mathbf{Facts}(\{At(0:5,d,l_1), Tied(0:5,n,l_2)\})$, $\mathbf{Deadline}(50)$, $\mathbf{Unsolved}(Goal(out\_of\_danger(50,n,l_2)))$,

$$\mathbf{Ppl}(save, 6, \left\{ \begin{bmatrix} At(t_6,d,l_1) \\ Pace(t_6:t_{10},d,l_1:l_1+v) \\ At(t_{10},d,l_1+v) \end{bmatrix}_{1\ldots} \quad \begin{bmatrix} At(t_8,d,l_1+34v) \\ Pace(t_8:t_7,d,l_1+34v:l_2) \\ At(t_7,d,l_2) \end{bmatrix}_{35\ldots} \right\})$$

$\{t_2 \le 50, t_1 = t_2 - 1, t_4 \le t_1, t_5 = t_4 - 1, t_3 = t_4 - 3, t_3 = t_9 - 1,$
$t_7 \le t_3, t_8 = t_7 - 1, t_6 = t_7 - 35, t_6 = t_{10} - 1\})$,

$\mathbf{Proj}(save, \{At(0:t_8,d,l_1), At(t_7:\infty,d,l_2), Tied(0:t_5,n,l_2), Not\_tied(t_4:\infty,n,l_2),$
$Out\_of\_danger(t_2:\infty,n,l_2), Pull(t_1:t_2,d,n,l_2), Release(t_3:t_4,d,n,l_2), Run(t_6:t_7,d,l_1:l_2),$
$Succ\_u_1(t_9:\infty), Succ\_u_2(t_5:\infty), Succ\_u_3(t_4:\infty), Untie_1(t_3:t_9,d,n,l_2),$
$Untie_2(t_9:t_5,d,n,l_2), Untie_3(t_5:t_4,d,n,l_2)\})$, $\mathbf{Wet}(save, 39)$, $\mathbf{Feasible}(save, 5)$

In step 6 Dudley has planned his run (35 paces) and is ready to start enacting his plan. This is seen by comparing step 6 and step 7; in the latter he no longer has the plan to do the first pace (from l_1 to $l_1 + v$ since he has moved this to his 'do' list of actions (not shown) since (in this case) the *Facts* list does not contradict his projected position of l_1. Here v is his velocity (i.e., one pace per second).

7. $\mathbf{Facts}(\{At(0:6,d,l_1), Tied(0:6,n,l_2)\})$, $\mathbf{Deadline}(50)$,
$\mathbf{Unsolved}(Goal(out\_of\_danger(50,n,l_2)))$,

$$\mathbf{Ppl}(save, 7, \left\{ \begin{matrix} \begin{bmatrix} At(8,d,l_1+v) \\ Pace(8:9,d,l_1+v:l_1+2v) \\ At(9,d,l_1+2v) \end{bmatrix}_{1\ldots} & \begin{bmatrix} At(41,d,l_1+34v) \\ Pace(41:42,d,l_1+34v:l_2) \\ At(42,d,l_2) \end{bmatrix}_{34} \\ \begin{bmatrix} At(t_3,d,l_2) \\ Untie_1(t_3:t_9,d,n,l_2) \\ Succ\_u_1(t_9) \end{bmatrix}_{35\ldots} & \begin{bmatrix} Not\_tied(t_1,n,l_2) \\ Pull(t_1:t_2,d,n,l_2) \\ Out\_of\_danger(t_2,n,l_2) \end{bmatrix}_{38} \end{matrix} \right\}$$

$\{t_2 \le 50, t_1 = t_2 - 1, t_4 \le t_1, t_5 = t_4 - 1, t_3 = t_4 - 3, t_3 = t_9 - 1,$
$t_6 = 7, t_{10} = 8, \ldots, t_7 = 42, t_7 \le t_3\})$,

$\mathbf{Proj}(save, \{At(0:t_6,d,l_1), At(t_{10},d,l_1+v), \ldots, At(t_7:\infty,d,l_2), Tied(0:t_5,n,l_2),$
$Not\_tied(t_4:\infty,n,l_2), Out\_of\_danger(t_2:\infty,n,l_2), Pull(t_1:t_2,d,n,l_2),$
$Release(t_3:t_4,d,n,l_2), Run(t_6:t_7,d,l_1:l_2), Succ\_u_1(t_9:\infty), Succ\_u_2(t_5:\infty), Succ\_u_3(t_4:\infty),$
$Untie_1(t_3:t_9,d,n,l_2), Untie_2(t_9:t_5,d,n,l_2), Untie_3(t_5:t_4,d,n,l_2),$
$Pace(t_6:t_{10},d,l_1,l_1+v), Pace(t_{10}:t_8,d,l_1+v,l_1+2v),$
$Pace(t_8:t_7,d,l_1+2v,l_2),\})$ $\mathbf{Wet}(save, 38), \mathbf{Feasible}(save, 6)$

We see in step 7 above that now Dudley believes he will be at $l_1 + v$ by time 8, having taken the first pace toward Nell during the one second between times 7 and 8. His actions continue, until by step 47 he has saved Nell.

References

[1] Mark Boddy and Thomas Dean. Solving time-dependent planning problems. In *Proceedings of IJCAI-89*, pages 979–984, Detroit, Michigan, 1989.

[2] D. E. Davis. Inferring ignorance from the locality of visual perception. In *Proceedings, AAAI88*, St. Paul, Minnesota, 1988.

[3] Thomas Dean. Planning and temporal reasoning under uncertainty. In *IEEE Workshop on Principles of Knowledge based Systems*, Denver, Colorado, 1984.

[4] Thomas Dean and Mark Boddy. An analysis of time-dependent planning. In *Proceedings, AAAI88*, pages 49–54, 1988.

[5] Thomas Dean, R. James Firby, and David Miller. Hierarchical planning involving deadlines, travel time and resources. *Computational Intelligence*, 4:381–389, 1988.

[6] J. Drapkin, M. Miller, and Donald Perlis. Life on a desert island. In *Proc. Workshop on The Frame Problem in Artificial Intelligence*, pages 349–357. American Association for Artificial Intelligence, 1987.

[7] J. Elgot-Drapkin. *Step-Logic: Reasoning situated in time*. PhD thesis, Univ. of Maryland, 1988.

[8] J. Elgot-Drapkin and D. Perlis. Reasoning situated in time: basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence*.

[9] A. Haas. Possible events, actual events, and robots. *Computational Intelligence*, 1, 1985.

[10] E. Horvitz, G. Cooper, and D. Heckerma. Reflection and action under scare resources: Theoretical principles and empirical study. In *Proceedings of IJCAI-89*, pages 1121–1127, Detroit, Michigan, 1989.

[11] E. J. Horvitz. Reasoning under varying and uncertain resource constraints. In *Proceedings, AAAI88*, pages 111–116, 1988.

[12] S. Kraus and D. Perlis. Assessing others' knowledge and ignorance. In *Proc. of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 220–225, 1989.

[13] D. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.

[14] S. Russell and E. Wefald. Principles of metareasoning. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 400–411. Morgan-Kaufman, 1989.

[15] Austin Tate, James Hendler, and Mark Drummond. A review of AI planning techniques. To appear.