The Incompleteness of Planning with Volatile External Information

Tsz-Chiu Au and **Dana Nau**[1]

Abstract. In many real-world planning environments, some of the information about the world is both external (the planner must request it from external information sources) and volatile (it changes before the planning process completes). In such environments, a planner faces two challenges: how to generate plans despite changes in the external information during planning, and how to guarantee that a plan returned by the planner will remain valid for some period of time after the planning ends. Previous works on planning with volatile information have addressed the first challenge, but not the second one.

This paper provides a general model for planning with volatile external information in which the planner offers a guarantee of how long the solution will remain valid after it is returned, and an *incompleteness theorem* showing that there is no planner that can succeed in solving *all* solvable planning problems in which there is volatile external information.

1 INTRODUCTION

In many real-world planning environments, the planner must request information from external information sources (databases, CAD systems, web services, and the like), incurring a lag time for receiving the answers. In practical planning situations, the planning activity may take more time than is needed to execute the resulting plan [6, 8, 9]. In such cases, some of the information on which the plan depends will change while the planning process is going on. For instance:

- [8] describes a domain-specific system that uses AI planning to do web service composition. Information from web services can change before planning finishes, and the system modifies its plans in response to such changes.
- At our university, the web site for booking concert-hall tickets displays a countdown timer showing how much longer the system will hold the seats that the user has booked. If the countdown ever reaches 0 (e.g., if user spends too much time examining the rest of the concert schedule), the booking will expire and the user will have to select seats all over again.
- When a traveler tries to construct a travel plan, the information about an airline flight may expire while the traveler was trying to plan some other details of the trip.

In such environments, how to cope with changes of external information and at the same time generate a plan that can be executed correctly is a big challenge. Most existing planners will not do this correctly unless they are modified.

In some cases, the interleaving of planning and execution [3, 7] is a good strategy to deal with volatile information. But if the wrong choice of action can cause a failure that is irrecoverable (or recoverable only at a large cost), then it is necessary to reason, while the plan is being generated, about whether the assumptions that are being used to choose an action will still be true when the action is executed.

In [1] we described *query management strategies* that can adapt existing planners to deal with volatile external information by backtracking the planner to the first point in the planning process that makes use of outdated information, obtaining the up-to-date information, and resuming the execution from that that point. A primary limitation of that work was that it provided no guarantee of completeness, i.e., no guarantee that a planner using such a query management strategy could succeed in every solvable planning environment.

In this paper, we provide an *impossibility theorem* showing that it is *impossible* to have a planner that can successfully find a valid solution in every solvable environment. This resembles the famous impossibility theorems in distributed systems ⟨http://www.podc.org/influential/2001.html⟩, which says that some problems in distributed systems are fundamentally unsolvable [2]. Remedying this problem is far from trivial, because it requires a modification to the fundamental assumptions of the distributed system model people have been used. Many of those solutions are not ideal, but they are needed in order to able to solve certain problems.

2 INCOMPLETENESS IN VEI-PLANNING

Let L be a planning language such as PDDL [5]. We construct a new language $\hat{L}$ that contains all the symbols of L plus some additional symbols called *unknowns* as follows: given a set $\hat{U}$ of unknowns, and a set $dom(u_i)$ of terms (called the *values* of u_i) in L for each unknown u_i, $\hat{L}$ is a set of expressions produced by taking any expression e in L and replacing zero or more terms in e with corresponding unknowns.

For each unknown u, there is a piecewise-constant function $\mathbb{V}_u : \mathbb{R} \rightarrow dom(u)$, such that u's value at time t is $\mathbb{V}_u(t)$. To learn about the value of u, a planner $\tilde{\mathcal{A}}$ must issue *queries* to an *information source* I_u: if $\tilde{\mathcal{A}}$ sends a query q for an unknown u to I_u, I_u will return a value $v(q) = \mathbb{V}_u(t')$ for q at time $t' = t_{return}(q)$. If $\tilde{\mathcal{A}}$ knows nothing about how long $v(q)$ will remain true, it cannot provide any sort of guarantee about the correctness of the plan it generates: the information may change during plan execution, invalidating the plan. Thus, we assume that I_u's answer $ans(q)$ will include both $v(q)$ and an *expiration time* $t_{expire}(q)$. $v(q)$ is guaranteed to be valid until $t_{expire}(q)$, after which time it may change (see Fig. 1).

A *VEI-problem* $\tilde{P}$ ("VEI" stands for "volatile external information") is a triple $\langle \hat{P}, T, \mathbb{E} \rangle$, where $\hat{P}$ is a planning problem written in $\hat{L}$, T is a non-negative real number called a

[1] University of Maryland, College Park, U.S.A., email: {chiu,nau}@cs.umd.edu

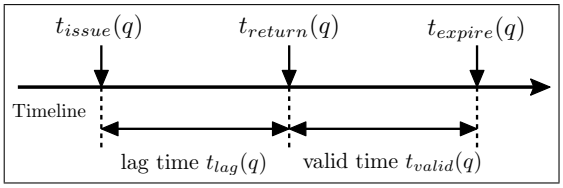

Figure 1. The lag time and the valid time of a query q. $v(q)$ is guaranteed valid only between $t_{return}(q)$ and $t_{expire}(q)$. After $t_{expire}(q)$, the value may possibly change.

validity guarantee, and $\mathbb{E}$ is a (finite or infinite) set of *VEI-environments*, each of which is a pair $\langle \mathcal{E}_{lag}, \mathcal{E}_{ans} \rangle$, where $\mathcal{E}_{lag} : \hat{U} \times \mathbb{R} \rightarrow \mathbb{R}$ and $\mathcal{E}_{ans} : \hat{U} \times \mathbb{R} \rightarrow \hat{V} \times \mathbb{R}$ are functions giving the lag time and the answer to a query q for u issued at time t, respectively. We assume that a planner $\tilde{\mathcal{A}}$ resides in one of the VEI-environments in $\mathbb{E}$ during its execution, but $\tilde{\mathcal{A}}$ has no prior knowledge about which one it is.

A value v of u is *confirmed* at time t if and only if $\tilde{\mathcal{A}}$ has issued a query q for u and has received an answer $ans(q) = (v, t_{expire}(q))$ such that $t_{return}(q) \leq t < t_{expire}(q)$. v is *T-confirmed* at time t if and only if v is confirmed at time t' for all $t \leq t' < t + T$. A solution π is *T-confirmed* at time t if all the values used by $\tilde{\mathcal{A}}$ for constructing π are T-confirmed at time t. A planner $\tilde{\mathcal{A}}$ *succeeds* in a VEI-problem $\langle \hat{P}, T, \mathbb{E} \rangle$ if and only if no matter which VEI-environment (among $\mathbb{E}$) $\tilde{\mathcal{A}}$ resides in, $\tilde{\mathcal{A}}$ returns a T-confirmed solution when $\tilde{\mathcal{A}}$ terminates. A VEI-problem $\tilde{P}$ is *solvable* if and only if there exists a planner that succeeds in $\tilde{P}$. A planner $\tilde{\mathcal{A}}$ is *complete* if and only if $\tilde{\mathcal{A}}$ succeeds in *all* solvable VEI-problems. [2]

Theorem 1 states the necessary condition under which there exists a successful planner for a VEI-problem.

Theorem 1 *A VEI-problem $\langle \hat{P}, T, \mathbb{E} \rangle$ is solvable only if for all $\mathcal{E} = \langle \mathcal{E}_{lag}, \mathcal{E}_{ans} \rangle \in \mathbb{E}$, (1) There exists a solution π for some instantiation of $\hat{P}$ with values $v_1, v_2, \ldots, v_m$ for unknowns u_1, u_2, ..., u_m, respectively; and (2) there exist times $t_1, t_2, \ldots, t_m$ such that the length of the time interval $\bigcap_{1 \leq j \leq m} \left[t_j + \mathcal{E}_{lag}(u_j, t_j),\ t_j + \mathcal{E}_{lag}(u_j, t_j) + t^j_{expire} \right) \geq T$, where where $\mathcal{E}_{ans}(u_j, t_j) = (v_j, t^j_{expire})$.*

For the above condition to be sufficient for solvability, the planner would need to be able to know instantaneously whenever a value expires. But that would require a communication channel with an infinite band bandwidth and information sources with infininte amount of computational power—an impossible requirement. In practice, there is usually an unavoidable delay between two consecutive queries. For example, a database can only process one query at a time, and there is a delay between the processing of two consecutive queries. Therefore, it is realistic to make the following assumption:

Assumption 1 (Limited Accessibility) *For each query q for an unknown u issued at time t, there is a period of time after t such that the planner cannot issue a query for u in the time period $[t,\ t + t')$, where t' is the length of the period.*

Using this assumption, we can find two VEI-environments such that no planner can succeed in them at the same time.

[2] Our definition of completeness is the usual one for search algorithms [10, p. 71] and planning algorithms [4, p. 544]: an algorithm is complete if it finds a solution whenever a solution exists.

Lemma 1 *Under the limited accessibility assumption, there exist two VEI-problems $\tilde{P}_1 = \langle \hat{P}, T, \{\mathcal{E}_1\} \rangle$ and $\tilde{P}_2 = \langle \hat{P}, T, \{\mathcal{E}_2\} \rangle$ that are solvable, but the combined VEI-problem $\langle \hat{P}, T, \{\mathcal{E}_1, \mathcal{E}_2\} \rangle$ is unsolvable.*

This follows that there is no complete planner; otherwise, this lemma contradicts the fact that if both $\langle \hat{P}, T, \{\mathcal{E}_1\} \rangle$ and $\langle \hat{P}, T, \{\mathcal{E}_2\} \rangle$ can be solved by the same planner $\tilde{\mathcal{A}}$, the combined VEI-problem $\langle \hat{P}, T, \{\mathcal{E}_1, \mathcal{E}_2\} \rangle$ can also be solved by $\tilde{\mathcal{A}}$.

Theorem 2 (Incompleteness) *Under the limited accessibility assumption, there is no complete planner.*

3 CONCLUSIONS

Our incompleteness theorem shows that even if we restrict our attention to solvable VEI-environments, there is no planner that can solve every one of these environments. A consequence of this is that no planner dominates all other planners in terms of its coverage, i.e., there is no planner that can solve all of the environments that are solvable by other planners.

In future, we would like to study how to get around the incompleteness, so that one can create planners that are complete if certain restrictions are satisfied. One possible solution is to provide information about the timings of queries to planners, so that planners can schedule its queries in advance. Another solution is to look for randomized query strategies that have a high probability of success.

ACKNOWLEDGEMENTS

This work was supported in part by ISLE subcontract 0508268818 to DARPA's Transfer Learning program, UC Berkeley subcontract SA451832441 to DARPA's REAL program, and NSF grant IIS0412812.

REFERENCES

[1] Tsz-Chiu Au, Dana Nau, and V.S. Subrahmanian, 'Utilizing volatile external information during planning', in *ECAI*, pp. 647–651, (2004).
[2] Michael J. Fischer and Nancy A. Lynch, 'Impossibility of distributed consensus with one faulty', *JACM*, **32**(2), 374–382, (1985).
[3] Michael R. Genesereth and Illah R. Nourbakhsh, 'Time-saving tips for problem solving with incomplete information', in *AAAI*, pp. 724–730, (1993).
[4] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*, Morgan Kaufmann, May 2004.
[5] Malik Ghallab, Adele Howe, Craig Knoblock, Drew McDermott, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins, 'PDDL—the planning domain definition language', Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, (1998).
[6] i2 Technologies. Reducing planning cycle time at altera corporation. http://www.i2.com/assets/pdf/96FDF2C7-71C7-43B5-906A01BAE2F0AE76.pdf, 2002.
[7] Craig A. Knoblock, 'Planning, executing, sensing, and replanning for information gathering', in *IJCAI*, (1995).
[8] Ugur Kuter, Evren Sirin, Dana Nau, Bijan Parsia, and James Hendler, 'Information gathering during planning for web services composition', *Journal of Web Semantics*, (2005).
[9] W. H. McRaven, *Spec Ops : Case Studies in Special Operations Warfare: Theory and Practice*, Presidio Press, 1996.
[10] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach (Second Edition)*, Prentice Hall, 2003.