Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling

The Complexity of Partial-Order Plan Viability Problems

Xing Tan
Telfer School of Management,
University of Ottawa,
Ottawa, Ontario,
K1N 6N5 Canada
tan@telfer.uottawa.ca

Michael Gruninger
Dept. of Mechanical & Industrial Engineering,
University of Toronto,
Toronto, Ontario,
M5S 3G8 Canada
gruninger@mie.utoronto.ca

Abstract

Estimating the distance from a current partial-order plan to the goal state of the plan task is a challenging problem, with past research achieving only limited success. In an effort to understand the reasons for this situation, we investigate the computational complexity of the partial-order plan viability problem. We define several boundaries between the tractable and intractable subclasses of the problem, from which we identify several constraints that contribute to the computational intractability of the problem. These results bring new insights into the design and the development of future partial-order planning heuristics.

1 Introduction

Planning is defined as the problem of finding at least one sequence of events (actions) that will achieve a goal from an initial state. A partial-order plan compactly defines a collection of potentially exponentially many total-order plans by specifying a set of partially ordered events. Each linearization of the plan thus corresponds to a total-order plan. With the strategy of least commitment and constraints posting, planning based on partial-order representations searches for solutions through repeatedly refining a current partial-order plan, which is initially empty.

The nature of planning through refinement enables a partial-order planner to offer a flexible and intuitive approach to domains where replanning is frequent due to the uncertainty of the plan execution environment, and a quick and clear understanding of the behavior of the planning algorithm by human-beings is critical (Bresina et al. 2005). Partial-order planning approaches have their unique strength in domains where plan executions are subject to environmental uncertainty and interactions between planners and humans are frequent. On the other hand, to perform competitively relative to the total-order planners such as the GraphPlan or the SATPlan, a partial-order planner has to be equipped with an effective and fast heuristic to estimate the viability of a current partially ordered plan w.r.t. the goal state. It becomes apparent, however, that the design and the development of partial-order planning heuristic is quite tricky (Weld 2011) and research on the domain in the past decade achieves only limited success (the REPOP planner introduced in (Nguyen and Kambhampati 2001) is one of the few examples). Nevertheless, there are recent efforts towards developing heuristics for partial-order planning, e.g., (Bercher, Geier, and Biundo 2013).

We believe that a comprehensive study on why determining the viability/validity of a partial-order plan[1] is computationally difficult would facilitate further practice of partial-order planning heuristic design and development. Accordingly, we investigate in this paper the source of the computational intractability of the partial-order plan viability problem, by identifying several different constraints, where each constraint defines a case that delineates the tractability boundary.

Study of the complexity of the partial-order plan validity problem was initiated by (Nebel and Bäckström 1994) in the context of event systems, which are a variant to the classical propositional STRIPS representation. The work that we present in this paper also uses event systems for problem definitions. It is worth noting that the plan viability problem in event systems is equivalent to the *admissible* possible truth problem, whereas the temporal projection problem (more precisely, the possible truth problem) as appeared in (Dean and Boddy 1988) and (Nebel and Bäckström 1994) allows *inadmissibility*. Consequently, an event whose preconditions are not satisfied in a state is allowed to occur in the state, legally but effectlessly.

In this paper, we show that the plan viability problem maintains its NP-completeness in an almost-simple event system, a simple event system, or even an almost-simple event system whose cause-and-effect graph is a directed acyclic graph (DAG). Nevertheless, the problem is tractable in a simple event system with a cause-and-effect graph that is a DAG, whereas the corresponding inadmissible possible truth problem remains NP-complete. This fact indicates that the role of *admissibility* is sometimes critical in bringing the problem into tractable zones. In addition, many other constraints, such as the size of the preconditions lists and effects lists, the size of the initial set, the topological structure of the

[1] Informally speaking, plan viability checking involves deciding whether there exists a linearization of the plan that achieves the given goal, whereas plan validity involves checking whether all linearizations of the plan achieve the given goal.

cause-and-effect graph, all contribute to the intractability of the plan viability problem.

The remainder of the paper is organized as follows. Section 2 provides background materials. Major complexity results are presented in Section 3 and the paper is summarized in Section 4.

2 Background

Definitions on event systems in this section adapt and extend the ones in (Nebel and Bäckström 1994).

Definition 1 (Causal Structure) *A Causal Structure is defined as a 3-tuple* $\Phi = \langle \mathcal{P}, \mathcal{T}, \mathcal{R} \rangle$ *where* $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ *is a set of propositional atoms (conditions);* $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ *is a set of event types;* $\mathcal{R} = \{r_1, r_2, \ldots, r_p\}$ *is a set of causal rules in the form* $r_i = \langle t_i, \varphi_i, \alpha_i, \delta_i \rangle$ *where* $t_i \in \mathcal{T}$ *is the event type that triggers the application of* r_i*,* $\varphi_i \subseteq \mathcal{P}$ *is the set of preconditions,* $\alpha_i \subseteq \mathcal{P}$ *is the set of added conditions,* $\delta_i \subseteq \mathcal{P}$ *is the set of deleted conditions.*

The notion of event systems is introduced to describe a set of actual events that are subject to temporal constraints in the form of partial orders:

Definition 2 (Event System) *An event system is a 6-tuple* $\Theta = \langle \mathcal{P}, \mathcal{T}, \mathcal{R}, \mathcal{E}, \mathcal{O}, \mathcal{I} \rangle$ *where* $\mathcal{P}, \mathcal{T}, \mathcal{R}$ *are the same as the ones defined in* Φ*; and* $\mathcal{E} = \{e_1, e_2, \ldots, e_p\}$ *is a set of actual events, such that for each* e_i*,* $type(e_i) \in \mathcal{T}$*,* $\mathcal{O}$ *is a partial order on* $\mathcal{E}$*, and* $\mathcal{I}$ *is the initial state, a subset of* $\mathcal{P}$*.*

Definition 3 (Almost-simple Event System) *An event system* Θ *is almost-simple iff it is unconditional (i.e. for each event type* $t \in \mathcal{T}$*, there exists only one causal rule for* t*);* $|\mathcal{I}| \geq 1$*; and for each causal rule in the form* $r = \langle t, \varphi, \alpha, \delta \rangle$*,* $|\varphi| = |\alpha| = |\delta| = 1$*, and* $\delta = \varphi$*.*

Note that the definition of almost-simple even systems provided here is stronger than the one provided in (Nebel and Bäckström 1994) (Definition 3.1) in the sense that we require that the size of the precondition is precisely 1.[2]

Definition 4 (Simple Event System) *A simple event system* Θ *is an almost-simple event system, with additional constraint* $|\mathcal{I}| = 1$*.*

Definition 5 (Cause-and-Effect Graph) *A cause-and-effect graph of an almost-simple event system* Θ *is a directed graph. Each node in the graph corresponds to a condition, whereas each directed edge in the graph corresponds to a causal rule in* Θ*.*

Informally, the concept of causal structure is introduced to specify how a given **state**, which is a subset of $\mathcal{P}$, evolves over actual occurrences of events. Given a causal structure Φ and a state *State*, an actual event e_i with event type $t_i \in \mathcal{T}$ is applicable in *State* if and only if there exists a rule $r_i = \langle t, \varphi, \alpha, \delta \rangle$, whereas $t = t_i$ and the preconditions in φ are satisfied in *State*. If applicable, e_i changes *State* by adding some conditions to *State* and removing some others from it, in the way specified by the set of causal rules $\mathcal{R}$. For almost-simple event systems, we require that no condition to be added by e_i should already be a member of *State*.

Formally, let $app(State, e)$ denote the set of all applicable rules for an event $e \in \mathcal{E}$ in state *State*, e is admissible in *State* iff $app(State, e) \neq \emptyset$. Also, let $\boldsymbol{f} = \langle f_1, \ldots, f_k, \ldots, f_m \rangle$ be an event sequence, $\langle f_1, \ldots, f_k \rangle$ is denoted by $\boldsymbol{f}/f_k$. Further, we write "$\boldsymbol{f}; g$" to denote $\langle f_1, \ldots, f_m, g \rangle$.

Definition 6 (Result) *The change of states (subset of* $\mathcal{P}$*) over event sequences* $\boldsymbol{f}$ *(initially empty, denoted by* $\langle\rangle$*) is defined recursively as follows*

- $Result(State, \langle\rangle) = State$*,*
- $Result(State, (f; g)) = Result(State, f) - \{\delta(r) | r \in app(Result(State, f), g)\} \cup \{\alpha(r) | r \in app(Result(State, f), g)\}$.

Definition 7 (Admissibility) *An event sequence* $\boldsymbol{f} = \langle f_1, \ldots, f_k, \ldots, f_m \rangle$ *is admissible in a state State iff each* f_i*, for* $1 \leq i \leq m$*, is admissible in* $Result(State, \boldsymbol{f} \backslash f_i)$*.*

Given an event system Θ, the set of all linearizations of $\mathcal{E}$ with respect to $\mathcal{O}$ in Θ is denoted by $\mathtt{CS}(\Theta)$. Meanwhile, the subset of $\mathtt{CS}(\Theta)$, which includes linearizations that are admissible in $\mathcal{I}$ with respect to Θ, is denoted by $\mathtt{ACS}(\Theta)$.

Definition 8 (Plan and Planning Task) *A planning task* Π *is a triple* $\langle \Phi, \mathcal{I}, \mathcal{G} \rangle$ *of a causal structure, an initial state and a goal state, where* $\mathcal{I}$ *or* $\mathcal{G}$ *is also a subset of* $\mathcal{P}$*. A plan* Δ_Φ *is a 3-tuple* $\langle \Phi, \mathcal{E}, \mathcal{O} \rangle$*.*

Definition 9 (Viability and Validity of Plans) *Given a planning task* $\Pi = \langle \Phi, \mathcal{I}, \mathcal{G} \rangle$*, a condition* $p \in \mathcal{G}$*, and a partial-order plan* $\Delta_\Phi = \langle \Phi, \mathcal{E}, \mathcal{O} \rangle$*, we construct an event system* $\Theta = \langle \mathcal{P}, \mathcal{T}, \mathcal{R}, \mathcal{E}, \mathcal{O}, \mathcal{I} \rangle = \langle \Delta_\Phi, \mathcal{I} \rangle$*.*

The plan Δ_Φ *is viable for* p *with respect to* Π*, written as* $p \in Viable(\Pi, \Delta_\Phi)$*, iff there exists a linearization* $\boldsymbol{f}$ *in* $\mathtt{ACS}(\Theta)$ *such that* $p \in Result(\mathcal{I}, \boldsymbol{f})$*.*

The plan Δ_Φ *is valid for* p *with respect to* Π*, written as* $p \in Valid(\Pi, \Delta_\Phi)$*, iff* $\mathtt{CS}(\Theta) = \mathtt{ACS}(\Theta)$*, and for each linearization* $\boldsymbol{f}$ *in* $\mathtt{ACS}(\Theta)$*, we have* $p \in Result(\mathcal{I}, \boldsymbol{f})$*.*

3 Complexity of Plan Viability

In the two sections below, five classes of plan viability problems are considered. Three of them are NP-complete (Theorem 1 to Theorem 3), whereas the other two are polynomial-time solvable (Theorem 4 and Theorem 5).

3.1 Intractability Results

The proofs for NP-hardness presented in this section are based on polytime transformations from the NP-complete DHP problem - more precisely, the variant *Directed Hamiltonian Path* problem where both the starting point and the ending points are specified in the instance (Garey and Johnson 1979) - to the current problems. Each transformation converts a directed graph $G = (V, E, v_x, v_y)$, where V is the

[2] We also remark that, since viability in both event systems are in NP, and NP-hardness of the stronger version is given in the current paper, Theorem 1 (to be provided in Section 3) can also be applied for almost-simple event systems as defined by (Nebel and Bäckström 1994). The additional benefit with a stronger version as presented here in this paper is that, Theorem 5 is thus easier to prove. Theorem 1 and Theorem 5 altogether enable us to draw one more edge crossing the tractability line in Figure 5.

set of vertices, E is the set of edges, v_x is the starting point, and v_y is the ending point, into an event system $\Theta(G)$, which is at least *almost-simple*, so that its cause-and-effect relationship can be represented as a directed graph. Proof ideas are given in this section for the theorems. Detailed proof for Theorem 1 is provided in particular.

Theorem 1 *Given a planing task $\Pi = \langle \Phi, \mathcal{I}, \mathcal{G} \rangle$, a plan $\Delta_\Phi = \langle \Phi, \mathcal{E}, \mathcal{O} \rangle$, a corresponding event system $\Theta = \langle \Delta_\Phi, \mathcal{I} \rangle$, and a condition $p \in \mathcal{G}$, if Θ is an almost-simple event system, deciding $p \in Viable(\Pi, \Delta_\Phi)$ is NP-complete.*

Proof Idea: Here we briefly explain how to generate an almost-simple event system $\Theta(G)$ from a given directed graph G. From a given node v_i in G, we first introduce in $\Theta(G)$ two conditions v_{i_L} and v_{i_R}, one event type t_{i_L,i_R}, and accordingly one and only one event e_{i_L,i_R}. The condition v_{i_L} is thus a so-called *left-node condition* whereas v_{i_R} is a so-called *right-node condition*, respectively. Event-types in $\Theta(G)$ corresponding to nodes in G (e.g., t_{i_L,i_R} corresponds to v_i), are referred to as *node event-types*. Similarly, an event like e_{i_L,i_R} is called a *node event*.

Note that, in each of the three polytime transformations for the NP-hardness proofs, we require that each event type in $\mathcal{T}$ corresponds to one and only one actual event in $\mathcal{E}$.

Given (v_i, v_j) in G, $v_{i\_L}, v_{i\_R}, t_{i\_L,i\_R}, e_{i\_L,i\_R}$, $v_{j\_L}, v_{j\_R}, t_{j\_L,j\_R}$, and e_{j_L,j_R}, we introduce *the edge event-type* t_{i_R,j_L} and *the edge event* e_{i_R,j_L}. Hence the graph depicted in Figure 1 is transformed into a cause-and-effect graph depicted in Figure 2, excluding for the moment the conditions $v^2_{x\_R}$, $v^2_{1\_R}$, and the two event-types (edges) leaving them.

Building up the set of initial conditions $\mathcal{I}$ starts with the membership of v_{x_L} in $\mathcal{I}$.

When constructing the ordering constraints $\mathcal{O}$ of $\Theta(G)$, we require that the node event e_{y_L,y_R} is preceded by all node events e_{i_L,i_R} where $i \neq y$. Hence if there exists an admissible event sequence $\boldsymbol{f_1}$ from e_{x_L,x_R} to e_{y_L,y_R}, achieving the condition v_{y_R}, this sequence must correspond to a DHP $\boldsymbol{P} = (V_1, E_1)$ in G from v_x to v_y. Conversely, if there exists a DHP $\boldsymbol{P}_1$ in G from v_x to v_y, then $\boldsymbol{P}_1$ corresponds to an admissible sequence $\boldsymbol{f_1}$ from e_{x_L,x_R} to e_{y_L,y_R} in $\Theta(G)$.

In general, $|E_1| < |E|$. In other words, $\boldsymbol{f_1}$ does not include all edge events. As such, *in-node* event-types are introduced such that the in-degree of a *right-node condition*, for each i excluding y in the cause-and-effect graph, equals to the condition's out-degree. However, for the right-node condition $v^1_{y\_R}$, its in-degree is 1 more than its out-degree, since if we have a linearization $\boldsymbol{f}$, it finishes with $v^1_{y\_R} \in Result(\mathcal{I}, \boldsymbol{f})$.

To this end, the cause-and-effect graph is completely constructed, as shown in Figure 2. Additionally, the *left-node conditions* for these in-node event-types are in the set $\mathcal{P}$ and it is required that these conditions are included in $\mathcal{I}$. It is also required that all in-node events are preceded by the event e_{y_L,y_R}.

Returning to Figure 1, the DHP $\boldsymbol{P}$ in G is $\langle v_x, v_1, v_2, v_y \rangle$, whereas its corresponding $\boldsymbol{f_1}$ is: $\langle e^{1,1}_{x\_L,x\_R}, e^{1,1}_{x\_R,1\_L}, e^{1,1}_{1\_L,1\_R}, e^{1,1}_{1\_R,2\_L}, e^{1,1}_{2\_L,2\_R}, e^{1,1}_{2\_R,y\_L}, e^{1,1}_{y\_L,y\_R} \rangle$, which can be further extended to a linearization $\boldsymbol{f}$: $\langle e^{1,1}_{x\_L,x\_R}, e^{1,1}_{x\_R,1\_L}, e^{1,1}_{1\_L,1\_R}, e^{1,1}_{1\_R,2\_L}, e^{1,1}_{2\_L,2\_R}, e^{1,1}_{2\_R,y\_L}, e^{1,1}_{y\_L,y\_R}, e^{2,1}_{x\_R,x\_R}, e^{1,1}_{x\_R,2\_L}, e^{2,1}_{1\_R,1\_R}, e^{1,1}_{1\_R,y\_L} \rangle$.

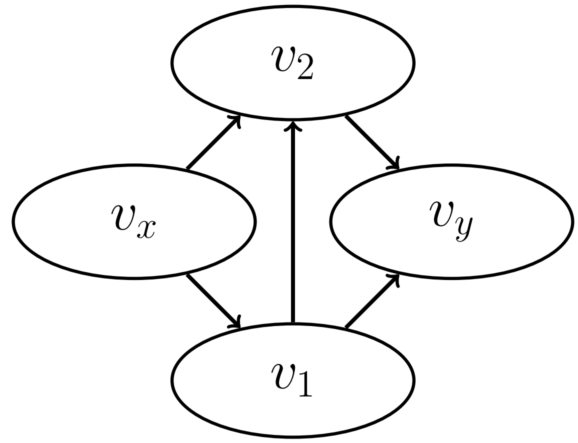

Figure 1: A digraph and its Hamiltonian path

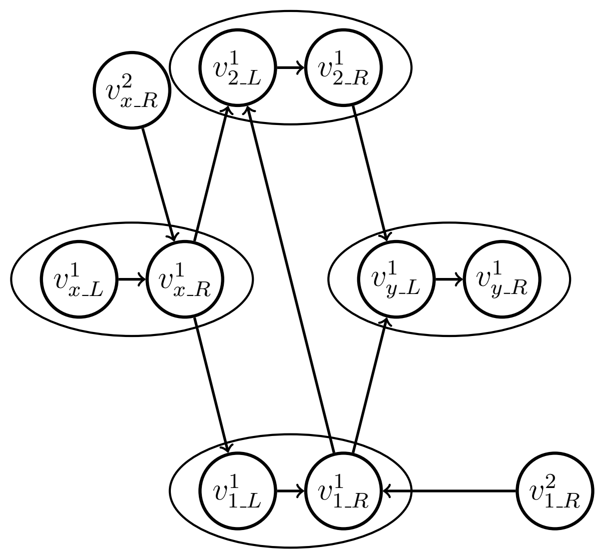

Figure 2: The cause-and-effect graph of $\Theta(G)$

Proof. The problem is in NP. Given a tuple $\langle \Pi, p, \Delta_\Phi \rangle$ (thus an event system Θ), and an event sequence $\boldsymbol{f}$, the membership of $\boldsymbol{f} \in \texttt{ACS}(\Theta)$ and $p \in Result(\mathcal{I}, \boldsymbol{f})$ can be verified in polytime.

Let $G = (V, E, v_x, v_y)$ be a digraph, where $V = \{v_1, ..., v_n, v_x, v_y\}$ and $|V| = n + 2$, $E = \{e_1, ..., e_m\}$ and $|E| = m$, $in(v_i)$ and $out(v_i)$ are the number of edges entering and leaving the node v_i, respectively, and $v_x, v_y \in V$ are the start point and the end point for a DHP, respectively, we define as follows an almost-simple (as $|\mathcal{I}| > 1$ in general) event system $\Theta = \langle \mathcal{P}, \mathcal{T}, \mathcal{R}, \mathcal{E}, \mathcal{O}, \mathcal{I} \rangle$ such that

- $\mathcal{P} = \mathcal{P}_{nodes} \cup \mathcal{P}_{in\_nodes} \cup \mathcal{P}_{in\_y}$ where

– $\mathcal{P}_{nodes} = \{v^1_{i\_L}; v^1_{i\_R} | 1 \le i \le n$ or $i = x$ or $i = y\}$,

– $\mathcal{P}_{in\_nodes} = \{v^{j+1}_{i\_R} | out(v_i) = k, 1 \le j < k, 1 \le i \le n$, or $i = x\}$ (for any node other than v_y, if the out-degree of v_i is k such that $k \ge 2$, then $(k-1)$ extra conditions, i.e., from $v^2_{i\_R}$ to $v^k_{i\_R}$, are also created in $\mathcal{P}$),

– $\mathcal{P}_{in\_y} = \{v^{j+1}_{y\_R} | out(v_y) = k, 1 \le j \le k\}$ (for v_y, however, if the out-degree of v_y is k such that $k \ge 1$, then k extra conditions, i.e., from $v^2_{i\_R}$ to $v^{k+1}_{i\_R}$, are also created in $\mathcal{P}$)

- $\mathcal{T} = \mathcal{T}_{edges} \cup \mathcal{T}_{nodes} \cup \mathcal{T}_{1\_R} \cup \mathcal{T}_{2\_R} \cup \ldots \cup \mathcal{T}_{n\_R} \cup \mathcal{T}_{x\_R} \cup \mathcal{T}_{y\_R}$ where

– $\mathcal{T}_{edges} = \{t^{1,1}_{i\_R,j\_L} | (v_i, v_j) \in E\}$ (edge event-types),

– $\mathcal{T}_{nodes} = \{t^{1,1}_{i\_L,i\_R} | 1 \le i \le n$ or $i = x$ or $i = y\}$ (node event-types),

– $\mathcal{T}_{i\_R} = \{t^{j,1}_{i\_R,i\_R} | out(v_i) = k, k \ge 2, 1 < j \le k\}$ (in-node event-types for all i except y),

– $\mathcal{T}_{y\_R} = \{t^{j,1}_{y\_R,y\_R} | out(v_i) = k, k \ge 1, 1 < j \le k+1\}$ (in-node event-types for y);

- $\mathcal{R} = \{\langle t^{q,r}_{i,j}, \{v^q_i\}, \{v^r_j\}, \{v^q_i\}\rangle | t^{q,r}_{i,j} \in \mathcal{T}\}$;

- $\mathcal{E} = \{e^{q,r}_{i,j} | t^{q,r}_{i,j} \in \mathcal{T}\}$; That is, each event type t in $\mathcal{T}$ has one and only one actual event occurrence e in $\mathcal{E}$.

- $\mathcal{O} = \mathcal{O}_1 \cup \mathcal{O}_2$ where

– $\mathcal{O}_1 = \{e^{1,1}_{x\_L,x\_R} \prec \{e | e \in \mathcal{E}$ iff $t \in \mathcal{T}_{nodes}\} \prec e^{1,1}_{y\_L,y\_R}\}$ (the x node event precedes all node events, which precede the y node event),

– $\mathcal{O}_2 = \{e^{1,1}_{y\_L,y\_R} \prec \{e | e \in \mathcal{E}$ iff $t \in \mathcal{T}_{i\_R}, 1 \le i \le n$ or $i = x$ or $i = y\}\}$ (the y node event precedes all in-node events);

- $\mathcal{I} = \{v^1_{x\_L} \cup \{v^j_{i\_R} | j > 1, 1 \le i \le n$ or $i = x$ or $i = y\}\}$ (the left-node condition for the start point v_x, and all in-node conditions, are initially set to true).

There exists a DHP from v_x to v_y in G iff $v^1_{y\_R} \in Viable(\Pi, \Delta)$.

($\Rightarrow$): If there exists a DHP $\boldsymbol{P}$ from v_x to v_y in G, then an admissible event sequence in Θ from edge event $e^{1,1}_{x\_L,x\_R}$ to edge event $e^{1,1}_{y\_L,y\_R}$, say $\boldsymbol{f_1}$, which starts with initial condition $v^1_{x\_L}$ and achieves condition $v^1_{y\_R}$ in the end, can be constructed accordingly. Since $\boldsymbol{P}$ is a DHP, all node events will occur, and occur exactly once, in $\boldsymbol{f_1}$, making the ordering constraint $\mathcal{O}_1$ satisfied. After $\boldsymbol{f_1}$, all in-node events, which are required by $\mathcal{O}_2$ to be preceded by the edge event $e^{1,1}_{y\_L,y\_R}$, can occur, enabling the remaining edge events that have not occurred in $\boldsymbol{f_1}$ to occur admissibly, and occur exactly once (note that, for all i except y, exactly $out(v_i) - 1$ in-node events are introduced to achieve the i^{th} right-node condition $out(v_i) - 1$ times). Hence, $\boldsymbol{f_1}$ can be extended to an $\boldsymbol{f}$ such that $\boldsymbol{f} \in \texttt{ACS}(\Theta)$ and $v^1_{y\_R} \in Result(\mathcal{I}, \boldsymbol{f})$.

($\Leftarrow$): Assume there exists an admissible event sequence $\boldsymbol{f} \in \texttt{ACS}(\Theta)$ such that $v^1_{y\_R}$ is achieved after $\boldsymbol{f}$, i.e., $v^1_{y\_R} \in Result(\mathcal{I}, \boldsymbol{f})$. Due to the specifications of the initial conditions $\mathcal{I}$ and the partial ordering constraints $\mathcal{O}$, we know that, first, the edge event $e^{1,1}_{x\_L,x\_R}$ must be the first event of $\boldsymbol{f}$; second, the subsequence $\boldsymbol{f}/e^{1,1}_{y\_L,y\_R}$, (from $e^{1,1}_{x\_L,x\_R}$ to $e^{1,1}_{y\_L,y\_R}$ inclusive in $\boldsymbol{f}$, written as $\boldsymbol{f_1}$), must include all node events and some edge events, but exclude any in-node events. Since $\boldsymbol{f_1}$ is admissible, it corresponds to a Hamiltonian path from v_x to v_y in G. □

Theorem 2 *Deciding $p \in Viable(\Pi', \Delta_{\Phi'})$, where Θ' is a simple event system, is NP-complete.*

Proof Idea: We continue with the previous example to explain the basic proof idea. $\Theta(G)$ is further transformed into a simple event system $\Theta'(G)$, where $|\mathcal{I}'| = 1$ and $v^1_{x\_L} \in \mathcal{I}'$.

We know that, whenever there exists an admissible event sequence $\boldsymbol{f_1'}$ in $\Theta(G)$, $v^1_{y\_R}$ holds after $\boldsymbol{f_1'}$. Since $\Theta'(G)$ is a simple event system, in-node conditions ($v^2_{x\_R}$ and $v^2_{1\_R}$ in Figure 2 and Figure 3) are not included in $\mathcal{I}'$. Alternatively, we use $v^1_{y\_R}$ repeatedly as a hub (kind of) to achieve them. More precisely, for each in-node condition $v^j_{i\_R}$, we introduce the *leaves-y* event-type $t^{1,j}_{y\_R,i\_R}$, which achieves $v^j_{i\_R}$ if $v^1_{y\_R}$ holds. Leaves-y event-types are marked by the numbers "1" and "6" in Figure 3.

From an in-node condition $v^j_{i\_R}$, an edge-event, which leaves $v^1_{i\_R}$ and is not included in $\boldsymbol{f_1'}$, is able to occur after $\boldsymbol{f_1'}$. After the occurrence of this edge-event, we need to find a way to achieve $v^1_{y\_R}$ again so that the remaining edge events could also be added to construct $\boldsymbol{f'}$ from $\boldsymbol{f_1'}$. As such, *out-node* event-types are introduced such that the out-degree of a *left-node condition*, for each i excluding x in the cause-and-effect graph, equals to the condition's out-degree. For the left-node condition $v^1_{x\_L}$, its out-degree is 1 more than its in-degree, since if we have a linearlization $\boldsymbol{f'}$, it starts with $v^1_{x\_L} \in \mathcal{I}'$. Out-node event types are marked by the numbers "4" and "9" in Figure 3. For each out-node condition $v^j_{i\_L}$, we introduce *returns-to-y* event-type $t^{j,1}_{i\_L,y\_R}$, which achieves $v^1_{y\_R}$ if $v^j_{i\_L}$ holds. Returns-to-y event-types are marked by the numbers "5" and "10" in Figure 3. The cause-and-effect graph, as shown in Figure 3, is completely constructed.

Note that, $\mathcal{O}'$ extends $\mathcal{O}$ by including additional constraints that $e^{1,1}_{y\_L,y\_R}$ precedes all in-node events, out-node events, all leaves-y events, and all returns-to-y events.

One possible event sequence $\boldsymbol{f'}$ transformed from the DHP $\boldsymbol{P}$ in $G = \{v_x, v_1, v_2, v_y\}$ (Figure 1) is $\langle e^{1,1}_{x\_L,x\_R}, e^{1,1}_{x\_R,1\_L}, e^{1,1}_{1\_L,1\_R}, e^{1,1}_{1\_R,2\_L}, e^{1,1}_{2\_L,2\_R}, e^{1,1}_{2\_R,y\_L}, e^{1,1}_{y\_L,y\_R}, e^{1,2}_{y\_R,x\_R}, e^{2,1}_{x\_R,x\_R}, e^{1,1}_{x\_R,2\_L}, e^{1,2}_{2\_L,2\_L}, e^{2,1}_{2\_L,y\_R}, e^{1,2}_{y\_R,1\_R}, e^{2,1}_{1\_R,1\_R}, e^{1,1}_{1\_R,y\_L}, e^{1,2}_{y\_L,y\_L}, e^{2,1}_{y\_L,y\_R}\rangle$, where the last ten events in $\boldsymbol{f'}$, from $e^{1,2}_{y\_R,x\_R}$ to $e^{2,1}_{y\_L,y\_R}$, are marked according in Figure 3, from "1" to "10".

Theorem 3 *Deciding $p \in Viable(\Pi'', \Delta_{\Phi''})$, where Θ'' is an almost-simple event system and the cause-and-effect graph is a DAG,*[3] *is NP-complete.*

[3]From the transformation we will see that the DAG is in particular a forest of trees.

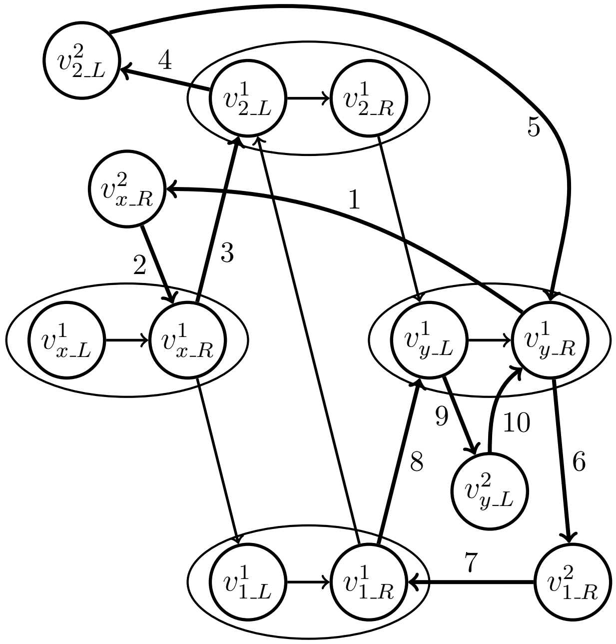

Figure 3: The cause-and-effect graph of $\Theta'(G)$

Proof Idea: Given $G = (V, E, v_x, v_y)$, we now construct Θ'', which is a modification of Θ (see the proof section for Theorem 1, for the definition of Θ). More specifically, the $\mathcal{T}''_{edges}$ in $\mathcal{T}''$ of Θ'' (also see the proof section for Theorem 1, for the definitions of $\mathcal{T}$ and $\mathcal{T}_{edges}$ in Θ) is defined alternatively as follows.

$$\mathcal{T}''_{edges} = \{t^{1,L}_{i\_R,i\_j}; t^{R,1}_{i\_j,j\_L} | (v_i, v_j) \in E\}.$$

That is, each edge in E is cut into two segments, corresponding to a *left-edge event-type* and a *right-edge event-type* in $\mathcal{T}''$. Accordingly, we have two new conditions: $v^L_{i\_j}$ and $v^R_{i\_j}$, reflecting the fact that two new nodes will be created after cutting an edge (v_i, v_j) in E of G. Two new actual events, a left-edge event $e^{1,L}_{i\_R,i\_j}$ and a right-edge event $e^{R,1}_{i\_j,j\_L}$ will be created accordingly in $\mathcal{E}''$ of Θ''.

The set of partial order $\mathcal{O}''$ of Θ'' is the union of $\mathcal{O}$ of Θ and $\mathcal{O}_3$, where

$$\mathcal{O}_3 = \{e^{1,L}_{i\_R,i\_j} \prec e^{R,1}_{i\_j,j\_L} | (v_i, v_j) \in E\}.$$

The initial state $\mathcal{I}''$ extends $\mathcal{I}$ by further including all right-edge conditions: $v^R_{i\_j}$, for all i and j such that $(v_i, v_j) \in E$. Note again that these "cutting activities" create a forest of trees as the cause-and-effect graph for Θ''.

Given an admissible event sequence $\boldsymbol{f}''_{\mathbf{1}}$, from e_{x_L,x_R} to e_{y_L,y_R}, achieving the condition v_{y_R}, we know that 1) the first occurrence of an edge-event in $\boldsymbol{f}''_{\mathbf{1}}$ must be a left-edge event, otherwise $\mathcal{O}''$ will be violated; 2) since any of the other left-edge events must occur after some occurrence of a right-edge event, the first left-edge event actually must be $e^{1,L}_{x\_R,x\_i}$ for some i, which is the last event in an event segment that achieves $v^L_{x\_i}$ from $v^1_{x\_L}$(in the example of Figure 4, $i = 1$); 3) the event right after $e^{1,L}_{x\_R,x\_i}$ must be a right-edge event, because $v^L_{x\_i}$ enables no event and all other conditions that are true in the state reached by $e^{1,L}_{x\_R,x\_i}$ are right-edge conditions and those in-node conditions, which enable events that are not allowed to occur in $\boldsymbol{f}''_{\mathbf{1}}$ because of $\mathcal{O}$; 4) that right-edge event must be $e^{R,1}_{x\_i,i\_L}$, otherwise again $\mathcal{O}''$ will be violated. Based on the above observations we can use mathematical-induction to prove that a right-edge event occurs right after its corresponding left-edge events in $\boldsymbol{f}''_{\mathbf{1}}$. Hence, it is still safe to claim that, as in the case for Θ, there is a DHP $\boldsymbol{P} = (V_1, E_1)$ in G from v_x to v_y iff in Θ'' there exists an $\boldsymbol{f}''_{\mathbf{1}}$.

Example cause-and-effect graph is illustrated in Figure 4, where one possible event sequence $\boldsymbol{f}''$ transformed from the DHP $\boldsymbol{P}$ in the graph G as shown in Figure 1 is
$\langle e^{1,1}_{x\_L,x\_R}, e^{1,L}_{x\_R,x\_1}, e^{R,1}_{x\_1,1\_L}, e^{1,1}_{1\_L,1\_R}, e^{1,L}_{1\_R,1\_2}, e^{R,1}_{1\_2,2\_L}, e^{1,1}_{2\_L,2\_R}, e^{1,L}_{2\_R,2\_y}, e^{R,1}_{2\_y,y\_L}, e^{1,1}_{y\_L,y\_R}, e^{2,1}_{x\_R,x\_R}, e^{1,L}_{x\_R,x\_2}, e^{R,1}_{x\_2,2\_L}, e^{2,1}_{1\_R,1\_R}, e^{1,L}_{1\_R,1\_x}, e^{R,1}_{1\_y,y\_L} \rangle$.

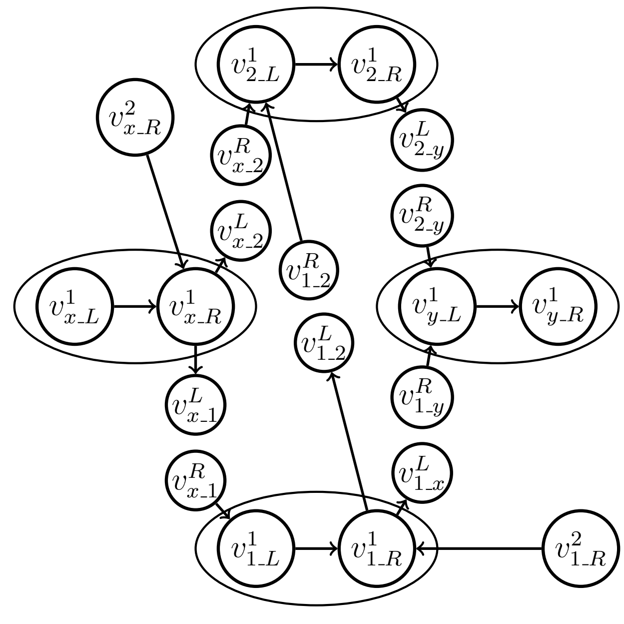

Figure 4: The cause-and-effect graph of $\Theta''(G)$

3.2 Tractability Results

We turn now to identifying tractable classes of the problems we have been considering.

Theorem 4 *Deciding $p \in Viable(\Pi''', \Delta_{\Phi'''})$, where Θ''' is a simple event system and the cause-and-effect graph G''' for the set of event types $\mathcal{T}'''$ in the causal structure Φ''' of Π''' is a DAG, is polytime solvable.*

Proof Idea: Although we are here proving a tractable plan viability result, it is safe to assume that each $t \in \mathcal{T}^{'''}$ has only one unique occurrence $e \in \mathcal{E}^{'''}$, since

- if there exist two events e_1 and e_2 both corresponding to a particular event type $t_{L,R}$, no admissible event sequence with respect to the event system $\Theta^{'''}$ can then achieve the condition t_L twice, since $\Theta^{'''}$ is simple and $G^{'''}$ is a DAG. Hence, the set $\mathtt{ACS}(\Theta^{'''})$ is empty and thus $p \notin Viable(\Pi^{'''}, \Delta_{\Phi^{'''}})$ for any condition p;
- if an event type $t_{L,R}$ does not have an actual event occurrence, then simply remove it from $\mathcal{T}^{'''}$.

It is also safe to assume that, for any two event types t_1 and t_2 in $G^{'''}$, there exists at least a path in between (a path from t_1 to t_2, or a path from t_1 to t_2, but not both), otherwise, there does not exist an Euler path for $G^{'''}$. Consequently, $p \notin Viable(\Pi^{'''}, \Delta_{\Phi^{'''}})$.

With these assumptions, the problem becomes trivial. The in-degree and out-degree of any condition p_i in $G^{'''}$ are at most one: if two event types, say t_1 and t_2, both achieve, or falsify, $_i$, then there does not exist a path between t_1 and t_2 in $G^{'''}$. Hence $G^{'''}$ is actually simply a chain and consequently the satisfiability of $\mathcal{O}^{'''}$ can be checked in linear time.

The following theorem states that, when constraints of partial orders are not present, plan viability problems are polynomial time solvable.

Theorem 5 *Deciding $p \in Viable(\Pi^*, \Delta_{\Phi^*})$, where Θ^* is an almost-simple event system and the partial-order $\mathcal{O}^*$ in Θ^* is empty, is polytime solvable.*

Proof Idea: Since $\mathcal{O}^*$ is empty, we need to find here an admissible complete sequence that achieves p. Since in Θ^*, an event type might have more than one actual event occurrences, we can transform the cause-and-effect graph $\Theta^*(G)$ built from the set of causal rules, to a multi-graph $\Theta^*(G^*)$, where edges are actual events and multi-edges reflect those multiple actual event occurrences. The problem is then equivalent to the polytime problem of finding an Euler tour in $\Theta^*(G^*)$ from multiple start points (the initial state) to multiple end points (the goal state). In particular, the plan is viable iff 1) the out-degree is one more than the in-degree for start points; 1) the out-degree is one less than the in-degree for end points; 3) p is one of the end points; 4) the out-degree equals its in-degree for any other vertices in the graph. Note that, for the algorithm to work properly, the size of the current state should not vary in any time touring the graph. In other words, a condition is currently true should not be achieved again by another event.

4 Summary and Discussion

The results are summarized graphically in Figure 5, in which an arrow connects a problem to its extension (in the sense of additional restrictions). Our major observations are:

- NP-completeness of the problem for unconditional event systems (the top-left box in the figure) is demonstrated by Theorem 6.4 in (Nebel and Bäckström 1994) (referred to in that paper as the admissible possible truth problem).

  The proof of the theorem employs a special event system whose set of partial order is empty (top-right box). This is a clear indication that other constraints, not partial order alone, are also the sources of the intractability.
- In problems without a partial order (right column of the figure), the intractability boundary occurs between unconditional event systems and almost-simple event systems, which means that the size of preconditions and effects in the causal rules also contribute to the intractability.
- The topological structure of the cause-and-effect graphs and the size of the initial conditions both contribute to the intractability. This is indicated by the fact that applying the DAG constraints and $|\mathcal{I}| = 1$ (i.e., simple event system) alone can not bring the problem into polytime-solvable zone, whereas when applied together, the problem becomes tractable.
- Considering the tractable problems (left-bottom box), if we do not require admissibility, then the revised problem corresponds to the problem studied in (Nebel and Bäckström 1994), which is NP-complete.[4] This means that "inadmissibility" contributes to the intractability as well.
- The inadmissable but effectless occurrence of events can alternatively be interpreted as disjunctive (i.e, conditional) preconditions for event-types, i.e., each event-type is associated with an additional causal rule, where the sets φ, α, δ are all empty.

In summary, the partial-order plan viability problem is NP-complete and our work indicates that all of the following constraints contribute to the computational intractability of the problem: 1) partial orders, 2) conditionality of the causal rules, 3) size of the preconditions and the effects of the causal rules, 4) size of the initial conditions, 5) topological structure of the cause-and-effect graphs, and 6) inadmissability.

It is shown in (Helmert 2004) that use of planning heuristics, which are based on decomposing the causal graph of a planning task, brings substantial improvement in solving practical instances of planning existence problems. Nevertheless, in (Jonsson, Jonsson, and Lööw 2013), it is demonstrated that plan existence problems are still computationally difficult in the worst cases (PSPACE-complete), even when the causal graph of a given planning problem is a DAG. This paper might shed some light on the design and development of new planning heuristics that are based on constraints other than DAG.

Because the plan viability problem and the *admissible* possible truth problem are equivalent, results in this paper are applicable to the admissible possible truth problem, as shown in (Tan 2012). Further, restricted classes on the cause-and-effect graphs, and on partial orders, are investigated in (Tan 2012) and (Tan and Gruninger 2009), in order to establish a partial borderline between tractability and intractability, for the inadmissible possible truth problem.

[4] General versions are presented in (Dean and Boddy 1988), most restricted version is presented as Theorem 3.3 in (Nebel and Bäckström 1994).

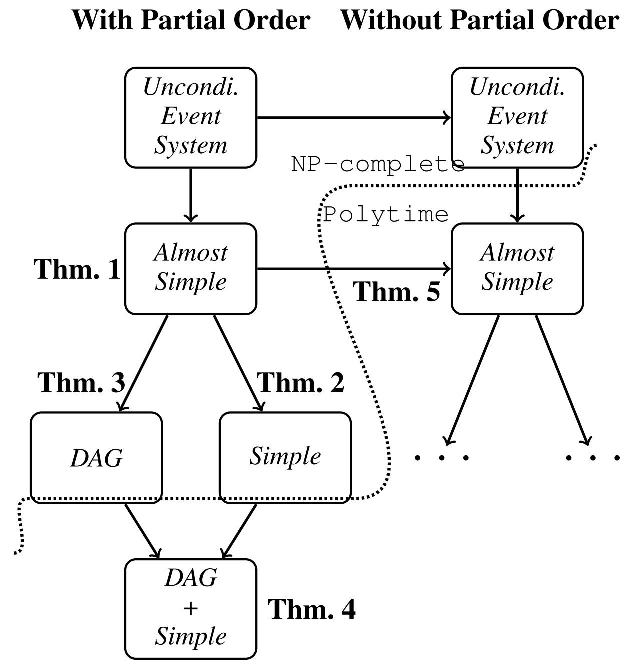

Figure 5: The computational hierarchy of the partially-ordered plan viability problems

Since the formalism adopts a propositional STRIPS-like presentation, an event system can be easily specified in the Planning Domain Definition Language (PDDL) (Russell and Norvig 2003), as PDDL includes sublanguages for STRIPS, among others such as the Action Description Language (ADL). Hence, the complexity analysis presented here can be easily extended to the benchmark planning problems expressed in PDDL, with minimal additional effort required.

In the context of plan execution monitoring, the partial-order plan viability problem is defined in Proposition 1 of (Muise, McIlraith, and Beck 2011) as checking the existence of a linearization $\boldsymbol{f}$ with respect to a state s, whereas at least one suffix of $\boldsymbol{f}$ can achieve the goal state from s. All our complexity results remain applicable to this variant definition, as we can easily generate worst cases where a plan is only possibly viable through its complete linearizations, instead of through any suffix of any linearization. In those cases, regression would not be able to generate promising intermediate states, which lead to the goal state. Additionally, as stated in Proposition 5.2 and Theorem 5.9 of (Nebel and Bäckström 1994), the plan validity problem is co-NP-complete for general causal structures but is polytime solvable for unconditional causal structures. Hence, computationally efficient regression-based approaches on plan validity problems might exist for cases where plan viability problems are intractable.

Finally, we note one interesting implication of Theorem 3 – in a system containing topologically isolated local cause-and-effect diagrams, predicting consequence is still computationally challenging as long as these diagrams remain temporally related.

5 Acknowledgments

This work is based in a large part on the first author's Ph.D. thesis (Tan 2012). We are thankful to anonymous reviewers, for their helpful feedback on earlier drafts of this paper. The final version of the paper was prepared when the first author was an NSERC-CREATE-sponsored Postdoctoral Research Fellow, hosted by Professor Wojtek Michalowski at the University of Ottawa.

References

Bercher, P.; Geier, T.; and Biundo, S. 2013. Using state-based planning heuristics for partial-order causal-link planning. In *Proceedings of the 36th Annual German Conference on AI (KI 2013)*, 1–12.

Bresina, J. L.; Jónsson, A. K.; Morris, P. H.; and Rajan, K. 2005. Activity planning for the Mars exploration rovers. In *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS 2005)*, 40–49.

Dean, T., and Boddy, M. 1988. Reasoning about partially ordered events. *Artificial Intelligence* 36(3):375–399.

Garey, M., and Johnson, D. 1979. *Computers and intractability - a guide to NP-completeness*. Cambridge, MA, USA: San Francisco: W.H. Freeman and Company.

Helmert, M. 2004. A planning heuristic based on causal graph analysis. In *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, 161–170.

Jonsson, A.; Jonsson, P.; and Lööw, T. 2013. When acyclicity is not enough: Limitations of the causal graph. In *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling (ICAPS 2013)*, 117–125.

Muise, C. J.; McIlraith, S. A.; and Beck, J. C. 2011. Monitoring the execution of partial-order plans via regression. In *Proceedings of the 22th International Joint Conference on Artificial intelligence (IJCAI-11)*, 1975–1982.

Nebel, B., and Bäckström, C. 1994. On the computational complexity of temporal projection, planning, and plan validation. *Artificial Intelligence* 66(1):125–160.

Nguyen, X., and Kambhampati, S. 2001. Reviving partial order planning. In *Proceedings of the 17th International Joint Conference on Artificial intelligence (IJCAI-01)*, 459–466.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice Hall.

Tan, X., and Gruninger, M. 2009. Towards tractable reasoning on temporal projection problems. In *Proceedigns of the IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 1, 723–727.

Tan, X. 2012. *The Application of Ontologies to Reasoning with Process Modeling Formalisms*. Ph.D. Dissertation, University of Toronto.

Weld, D. S. 2011. (AAAI-10 classic paper award) Systematic nonlinear planning: A commentary. *AI Magazine* 32(1):101–103.