UNIVERSITÀ di VERONA

Dipartimento di INFORMATICA

Rapporto di ricerca
Research report

111/2025

May 2025

Canonical Form of Nested Diamond Structures

Luke Hunsberger
Roberto Posenato

Questo rapporto è disponibile sul sito:
This report is available on the site:
https://iris.univr.it/

Abstract

A Simple Temporal Network with Uncertainty (STNU) is a data structure for representing and reasoning about time in contexts where some actions may have uncertain, but bounded durations. An STNU is *dynamically controllable* (DC) if there exists a strategy for executing its controllable timepoints such that no matter how the uncertain durations turn out, within their known bounds, all relevant constraints in the network will necessarily be satisfied. Several polynomial-time DC-checking algorithms have been presented in the literature. A real-time execution strategy, called RTE*, has been defined that preserves maximum flexibility while requiring minimal real-time computation; however, that strategy only guarantees a successful execution if: (1) the network is *extended* to accommodate conditional *wait* constraints; and (2) the network satisfies an additional property called *dispatchability*, that is stronger than dynamic controllability.

This report presents a novel theory of the dispatchability of Extended STNUs (ESTNUs) that is based on the *canonical form of nested diamond structures.* Each such structure entails an ordinary constraint that must be satisfied by any dispatchable ESTNU. This theory provides an avenue through which to explore more efficient algorithms for finding equivalent dispatchable networks having a minimal number of edges, which is important for limiting the computational requirements during real-time execution.

Keywords: Temporal constraint networks, dispatchable networks, simple temporal networks with uncertainty, dynamic controllability

(a) A sample STN graph

(b) A sample STNU graph

Figure 1: Sample STN and STNU graphs

1 Introduction and Background

1.1 Simple Temporal Networks

A *Simple Temporal Network* (STN) is a pair, $(\mathcal{T}, \mathcal{C})$, where $\mathcal{T}$ is a set of real-valued variables called *timepoints* (TPs) and $\mathcal{C}$ is a set of binary difference constraints, called *ordinary constraints,* each of the form $Y - X \leq \delta$, where $X, Y \in \mathcal{T}$ and $\delta \in \mathbb{R}$ [3]. The timepoints typically represent starting or ending times of actions; the constraints can represent deadlines, release times, and duration or inter-action constraints. Typically, we let $n = |\mathcal{T}|$ and $m = |\mathcal{C}|$. An STN is *consistent* if it has a solution as a constraint satisfaction problem. With no loss of generality, it is convenient to assume that each STN has a special timepoint Z whose value is fixed at *zero* (or some other convenient timestamp) and is constrained to occur at or before every other timepoint.[1] Each STN has a corresponding graph, $(\mathcal{T}, \mathcal{E})$, where the timepoints in $\mathcal{T}$ serve as nodes and each constraint $Y - X \leq \delta$ in $\mathcal{C}$ corresponds to a labeled directed edge $X \xrightarrow{\delta} Y$ in $\mathcal{E}$, called an *ordinary edge.* For convenience, such edges will be notated as (X, δ, Y). Figure 1a shows a sample STN graph. An STN is consistent if and only if its graph has no negative cycles [3].

Several well-known algorithms play important roles in the work presented in this paper, including: Bellman-Ford, Dijkstra, and Johnson's algorithm [2]. The $O(mn)$-time Bellman-Ford algorithm can be used to generate a solution for a consistent STN. Such a solution can then be used as a *potential function* to effectively convert the edge-weights in an STN graph to non-negative values. That, in turn, enables Dijkstra's $O(m + n \log n)$-time algorithm to be used. Johnson's algorithm uses Bellman-Ford to generate a potential function and then applies Dijkstra n times, once for each timepoint as source node, to do an All-Pairs Shortest-Paths (APSP) computation in $O(mn + n^2 \log n)$ time.

1.2 Simple Temporal Networks with Uncertainty

A *Simple Temporal Network with Uncertainty* (STNU) augments an STN to include *contingent links* that can be used to represent actions with uncertain durations [13]. Formally, an STNU is a triple $(\mathcal{T}, \mathcal{C}, \mathcal{L})$ where $(\mathcal{T}, \mathcal{C})$ is an STN and $\mathcal{L}$ is a set of *contingent links.* Each contingent link has the form (A, x, y, C) where $A, C \in \mathcal{T}$ and $0 < x < y < \infty$ [13]. A is called the *activation* TP and C the *contingent* TP; and we let $\Delta_C = y - x$. With no loss of generality, we assume that no contingent TP can serve as the activation TP for another contingent link, and that distinct contingent links have distinct activation TPs. Contingent links represent *uncontrollable possibilities* in the sense that an executor typically controls only the execution of A, but not C. The executor only *observes* the execution of C in real-time, knowing only that C will be executed such that $C - A \in [x, y]$. For example, your taxi ride might be represented by the contingent link $(A, 15, 25, C)$, where A is when you enter the taxi, C is when you arrive at your destination, and $C - A \in [15, 25]$ is the uncertain duration, learned only when you arrive. We let $k = |\mathcal{L}|$; and notate the set of contingent timepoints as $\mathcal{T}_c$; and the non-contingent (i.e., executable) timepoints as $\mathcal{T}_x = \mathcal{T} \backslash \mathcal{T}_c$.

Each STNU $(\mathcal{T}, \mathcal{C}, \mathcal{L})$ has a corresponding graph, $(\mathcal{T}, \mathcal{E}_o \cup \mathcal{E}_l \cup \mathcal{E}_u)$, where: $(\mathcal{T}, \mathcal{E}_o)$ is the graph for the STN $(\mathcal{T}, \mathcal{C})$; $\mathcal{E}_l$ is a set of *lower-case* (LC) edges; and $\mathcal{E}_u$ is a set of *upper-case* (UC) edges. The LC and UC edges correspond to the contingent links in $\mathcal{L}$, as follows. For each contingent link $(A, x, y, C) \in \mathcal{L}$, there is an LC edge $A \xrightarrow{c:x} C$ in $\mathcal{E}_l$ and a UC edge $C \xrightarrow{C:-y} A$ in $\mathcal{E}_u$, respectively representing the *uncontrollable possibilities* that the duration $C - A$ might take on its lower bound x or its upper bound y. For convenience, such LC and UC edges may be notated as $(A, c{:}x, C)$ and $(C, C{:}-y, A)$, respectively. Using this notation, $\mathcal{E}_o = \{(X, \delta, Y) \mid (Y - X) \leq \delta \in \mathcal{C}\}$; $\mathcal{E}_\ell = \{(A, c{:}x, C) \mid (A, x, y, C) \in \mathcal{L}\}$; and $\mathcal{E}_u = \{(C, C{:}-y, A) \mid (A, x, y, C) \in \mathcal{L}\}$. Figure 1b shows a sample STNU graph with a contingent link $(A, 1, 10, C)$. For convenience, we frequently blur the distinction between an STNU and its graph, and between edges and constraints.

An STNU is *dynamically controllable* (DC) if there exists a dynamic strategy for executing its *non-contingent* timepoints such that all of the constraints in $\mathcal{C}$ will necessarily be satisfied no matter how the contingent durations turn out—within their specified bounds [13, 4]. A strategy is dynamic in that it can react in real time to observations of contingent executions, but its execution decisions cannot depend on advance knowledge of

[1] It is not hard to show that in any consistent STN there is at least one timepoint that can play the role of Z.

contingent durations. As is common in the literature, this paper assumes that strategies can react instantaneously to observations [10].

Algorithms for determining whether any given STNU is DC are called *DC-checking algorithms.* Morris [11] presented the first $O(n^3)$-time DC-checking algorithm for STNUs. Cairo et al. [1] gave a $O(mn + k^2n + kn \log n)$-time algorithm, called `RUL`$^-$, that is faster on sparse networks. It is the DC-checking algorithm with the best worst-case time complexity. However, `RUL2021`, which is a modification of `RUL`$^-$, has been shown to be an order-of-magnitude faster on a variety of STNU benchmarks, although having the same theoretical complexity [5].

The *LO-graph* and the *OU-graph*. Certain subsets of edges appear regularly throughout the paper. *LO-edges* are either LC or ordinary edges, while *OU-edges* are either ordinary or UC edges. Similarly, *LO-paths* comprise LO-edges, while *OU-paths* comprise OU-edges. Finally, the *LO-graph*, notated as $\mathcal{G}_{\ell o}$, is the subgraph comprising all of the LO-edges, while the *OU-graph*, notated as $\mathcal{G}_{ou}$, is the subgraph comprising the OU-edges. Either of these subgraphs can be viewed as an STN by ignoring the alphabetic labels on its edges. In this way, the Bellman-Ford algorithm can be used to generate a potential function for it, thereby enabling the use of Dijkstra's algorithm (e.g., to update the potential function in response to the insertion of new edges).

1.3 STN Dispatchability

Although checking the consistency of an STN and finding a solution for it can be done in polynomial time, fixing a solution in advance of execution undermines the inherent flexibility of the STN representation. Instead, it can be desirable to preserve as much flexibility as possible until actions are actually performed (e.g., to enable reacting to unanticipated events), while minimizing the need for real-time computation. Toward that end, Tsamardinos et al. [16, 15] first specified a real-time execution algorithm for STNs and then defined an STN to be *dispatchable* if every run of that algorithm on that STN was guaranteed to generate a solution.[2] Their real-time execution algorithm provides maximum flexibility during execution by maintaining *time windows* for each timepoint, while requiring minimal real-time computation by only *locally* propagating the effects of each real-time execution event (i.e., propagating only to X's *neighbors*; that is, timepoints connected to X by a single edge). However, it does *not* provide a constraint-satisfaction guarantee for *all* runs of the algorithm on consistent STNs, which motivates the work on STN *dispatchability,* as follows.

Definition 1 (STN Dispatchability, [15])**.** An STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ is *dispatchable* if every run of the RTE algorithm on the corresponding STN graph $\mathcal{G} = (\mathcal{T}, \mathcal{E})$ necessarily generates a solution for $\mathcal{S}$.

It can be shown that *every* solution for a dispatchable STN can be generated by a run of the RTE algorithm.

Muscettola *et al.* [15] showed that for consistent STNs, the APSP graph is necessarily dispatchable, but its $O(n^2)$ edges cancel the benefits of *local* propagation. Their $O(n^3)$-time *edge-filtering* algorithm computes an equivalent dispatchable STN having a *minimal* number of edges by first constructing the APSP graph and then removing *dominated* edges (i.e., edges not needed for dispatchability). A faster $O(mn + n^2 \log n)$-time algorithm accumulates the undominated edges without first building the APSP graph [16]. In both algorithms, having the *minimal* number of edges in the dispatchable output is important because it minimizes the amount of local propagation required during execution.

Vee-paths and STN dispatchability. Morris [12] later found a graphical characterization of STN dispatchability in terms of *vee-paths.*

Definition 2 (Vee-paths [12])**.** A *vee-path* is a path comprising zero or more negative edges followed by zero or more non-negative edges. A vee-path from X to Y that is also a shortest path from X to Y is called a *shortest vee-path* (SVP).

Figure 2 shows a sample vee-path from X to Y that dominates the (dashed) direct edge from X to Y. For this vee-path, the enablement condition in the RTE algorithm ensures that the algorithm will execute B before A, and A before X; hence, local propagation ensures the satisfaction of the edges $(X, -7, A)$ and $(A, -3, B)$. On the other side, if RTE executes C before B, then the edge $(B, 1, C)$ is automatically satisfied; otherwise, local propagation ensures its satisfaction. Similarly, RTE necessarily satisfies the edge $(C, 4, D)$. Since RTE satisfies all the edges in the vee-path, it also satisfies the direct edge $(X, -5, Y)$. Hence, that edge is not needed to ensure dispatchability.

Theorem 1 ([12])**.** *An STN is dispatchable if and only if for each path from any X to any Y in the STN graph, there is a* shortest vee-path *from X to Y.*

[2] Muscettola et al. [15] refer to their algorithm as either the *Time Dispatching Algorithm* (TDA) or the *Dispatching Execution Controller* (DEC). Hunsberger and Posenato presented a more detailed version of the RTE algorithm along with a computational complexity analysis [9].

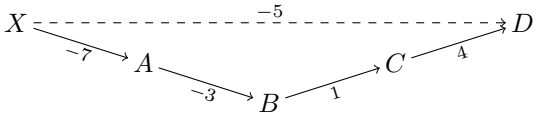
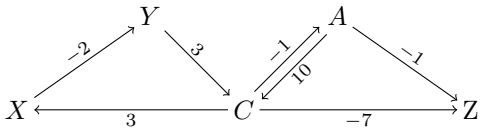
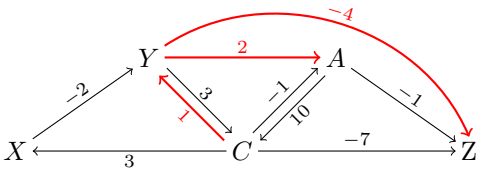


Figure 2: A sample *vee-path* that dominates a direct edge.

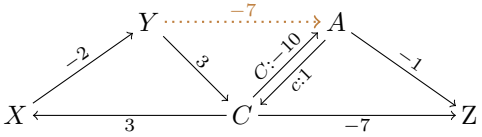


(a) The STN from Figure 1a

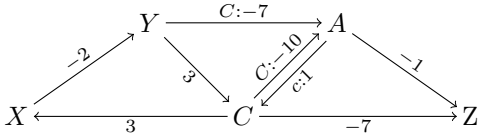


(b) An equivalent *dispatchable* STN

Figure 3: The STN from Figure 1a and an equivalent *dispatchable* STN



(a) The STNU from Figure 1b with an extra (dotted) edge that makes the network non-DC



(b) An ESTNU with a wait edge $(Y, C{:}-7, A)$ that preserves the DC property

Figure 4: The sample STNU graph and a similar ESTNU graph

Figure 3b shows a dispatchable STN that is equivalent to the STN from Figure 3a (new edges are thick and red). It is easy to check that each path has a corresponding SVP.

Definition 3 (Vee-path completeness)**.** An STN graph $\mathcal{G}$ is *vee-path complete* if for every X and Y for which there is a path from X to Y in $\mathcal{G}$ there is a shortest vee-path from X to Y.

With this definition, Theorem 1 can be restated as: An STN is dispatchable iff it is vee-path complete.

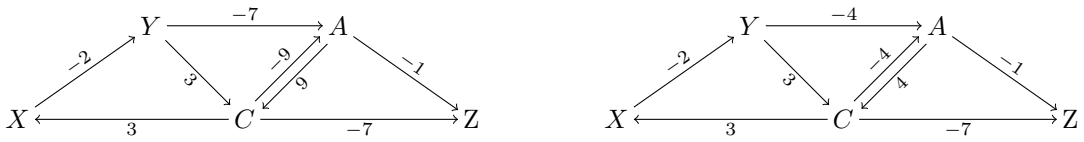
RTE complexity. With appropriate data structures, the RTE algorithm can be implemented to run in a total of $O(n^2)$ worst-case time, while allowing for maximum flexibility in the selection of each timepoint to execute next and the time at which to execute it [9].

1.4 Toward Dispatchability for (E)STNUs

As with STNs, it is important to preserve maximal flexibility for STNUs during execution, while requiring minimal real-time computation. Hence, it is important to define dispatchability for STNUs. Unfortunately, a DC STNU need not have an equivalent dispatchable *STNU*. For example, recall the sample STNU from Figure 1b, repeated in Figure 4a. (Ignore the dotted edge for now.) If A executes at time 0, then C may execute at any time $t \in [1, 10]$. To ensure that the constraint $C - Y \leq 3$ (i.e., $C \leq Y + 3$) is satisfied, it follows that as long as C remains unexecuted, Y cannot execute before time 7. But enforcing this constraint by inserting the edge $(Y, -7, A)$, shown as dotted in the figure, would result in an STNU that not only is not equivalent to the original, but is not even dynamically controllable. (With the dotted edge, the path $(A, c{:}1, C, 3, X, -2, Y, -7, A)$ constitutes a negative cycle in the LO-graph.) Instead, as described later in this section, to create equivalent dispatchable networks for arbitrary DC STNUs, it is necessary to increase the expressivity of STNUs by including a new kind of edge, called a *wait edge,* that represents a *conditional* constraint.

Anticipating this, Morris [11] defined an *Extended STNU* (ESTNU) to be an STNU augmented with wait edges. He then *defined* an ESTNU to be dispatchable if all of its projections (defined below) were dispatchable (as STNs); and then argued that a dispatchable ESTNU would necessarily be successfully executed by an appropriate real-time execution algorithm, but did not provide detailed proofs. Later, Hunsberger and Posenato [9] formalized a real-time execution algorithm for ESTNUs, called `RTE`*, and then proved that for a dispatchable ESTNU (according to Morris' definition), every run of `RTE`* on that ESTNU would necessarily satisfy all of its constraints. By doing so, they formally confirmed that every DC STNU has an equivalent dispatchable ESTNU.[3]

[3]Whereas the formal work on STN dispatchability first defined dispatchability as a performance guarantee with respect to the RTE algorithm and then proved that every dispatchable STN was vee-path complete, Morris' original approach to ESTNU dispatchability



(a) Projection where $\omega_c = C - A = 9$ (b) Projection where $\omega_c = C - A = 4$

Figure 5: Two projections of the sample ESTNU from Figure 4b

Wait edges. Many of the early DC-checking algorithms generate a new kind of edge, called a *wait edge,* that represents a *conditional* constraint [10]. A typical wait edge may be notated as $(V, C{:}{-}w, A)$, where A and C are the activation and contingent timepoints for a contingent link. Such a wait can be glossed as *"If the contingent timepoint C has not yet executed, then V must wait until at least w after the activation timepoint A."* If C happens to execute late, then V must wait until at least w after A; but if C executes early, then the wait is automatically satisfied and V can be executed at any time. Although the notation for wait edges is similar to that of UC edges, they are distinguished by the fact that V is distinct from the contingent timepoint C.

In general, it is not necessary to generate wait edges to determine the DC property (e.g., as seen in Morris' 2014 algorithm [11], the RUL$^-$ algorithm [1], and the `RUL2021` algorithm [5]). However, waits turn out to be necessary for enforcing the dispatchability of STNUs. Thus, Morris [11] defined an ESTNU to be an STNU augmented with a set of conditional wait constraints, and an ESTNU graph to include a corresponding set $\mathcal{E}_w$ of wait edges. Figure 4b shows a sample ESTNU graph. It is the same as that STNU from Figure 1b, except that it includes the wait edge, $(Y, C{:}{-}7, A)$. In earlier work [10], Morris proved that the conditional constraint represented by this wait edge must be satisfied by any valid dynamic execution strategy for this network.

Definition 4 (ESTNU graph)**.** An ESTNU graph is a tuple, $(\mathcal{T}, \mathcal{E}_o \cup \mathcal{E}_l \cup \mathcal{E}_u \cup \mathcal{E}_w)$, where $(\mathcal{T}, \mathcal{E}_o \cup \mathcal{E}_l \cup \mathcal{E}_u)$ is an STNU graph and $\mathcal{E}_w$ is a set of wait edges, each of the form $(V, C{:}{-}w, A)$, where A and C are the endpoints of some UC edge $(C, C{:}{-}y, A) \in \mathcal{E}_u$, $V \in \mathcal{T}\backslash\{C\}$, and $w \in \mathbb{R}$.

As already mentioned, Morris [11] defined the dispatchability of an ESTNU in terms of its STN *projections.* A projection of an ESTNU is the STN that results from assigning fixed durations to its contingent links [13, 11, 6]. Each such assignment of fixed durations is represented by a *situation* [13].

Definition 5 (Situation)**.** Let $\mathcal{S}$ be an STNU (or ESTNU) with k contingent links whose duration ranges are $[x_1, y_1], \ldots, [x_k, y_k]$. A *situation* for $\mathcal{S}$ is a k-tuple $\omega = (\omega_1, \omega_2, \ldots, \omega_k)$ where $\omega_i \in [x_i, y_i]$ for each $i \in \{1, 2, \ldots, k\}$. If C is the contingent TP for a link (A, x, y, C), then the duration $C - A$ in ω may be notated as ω_c.

A situation not only specifies the duration of each contingent link, it also determines the impact of each wait. This information is captured by the *projection* of an ESTNU [12, 5].

Definition 6 (Projection of an ESTNU)**.** For an ESTNU graph $\mathcal{G} = (\mathcal{T}, \mathcal{E}_o \cup \mathcal{E}_l \cup \mathcal{E}_u \cup \mathcal{E}_w)$, and a situation ω, the *projection* of $\mathcal{G}$ onto ω is the STN graph $\mathcal{G}_\omega = (\mathcal{T}, \mathcal{E}_o \cup \mathcal{E}^\omega_l \cup \mathcal{E}^\omega_u \cup \mathcal{E}^\omega_w)$, where:

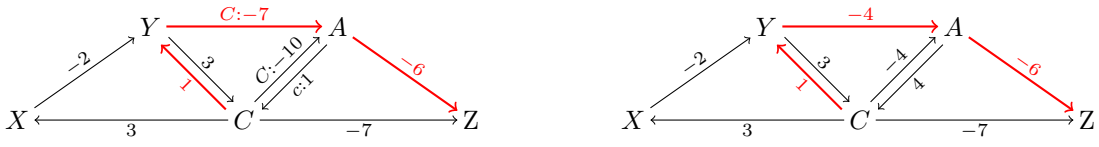
$$\begin{aligned}
\mathcal{E}^\omega_l &= \{(A_i, \omega_i, C_i) \quad | \; \exists (A_i, c_i{:}x_i, C_i) \in \mathcal{E}_l\} \\
\mathcal{E}^\omega_u &= \{(C_i, -\omega_i, A_i) \; | \; \exists (C_i, C_i{:}{-}y_i, A_i) \in \mathcal{E}_u\} \\
\mathcal{E}^\omega_w &= \{(V, \delta_i, A_i) \quad | \; \exists (V, C_i{:}-v, A_i) \in \mathcal{E}_w\}, \textit{where } \delta_i \text{ abbreviates } \max\{-\omega_i, -v\}
\end{aligned}$$

Similarly, for any path $\mathcal{P}$ in $\mathcal{G}$, the *projection* of $\mathcal{P}$ in the situation ω, is the ordinary path $\mathcal{P}_\omega$ whose edges are obtained by projecting the corresponding edges from $\mathcal{P}$. In particular, each LC edge $(A, c{:}x, C)$ projects to (A, ω_c, C), each UC edge $(C, C{:}{-}y, A)$ projects to $(C, -\omega_c, A)$, and each wait edge $(V, C{:}{-}v, A)$ projects to (V, δ, A), where $\delta = \max\{-\omega_c, -v\}$.

The edges in $\mathcal{E}^\omega_l$ and $\mathcal{E}^\omega_u$ fix each duration $C_i - A_i$ to the value ω_i. The edges in $\mathcal{E}^\omega_w$ reflect the *effective* constraints imposed by the waits in $\mathcal{E}_w$ in the situation ω. For example, consider the wait edge $(Y, C{:}{-}7, A)$ from Figure 4b. In the situation where $\omega_c = C - A = 4$, the wait projects onto $(Y, -4, A)$, reflecting that the conditional constraint vanishes when C executes early at $A + 4$. (Note that $\max\{-4, -7\} = -4$.) But in the situation where $\omega_c = 9$, the wait projects onto $(Y, -7, A)$, reflecting that Y must wait the full 7 units after A, since C did not execute early. (Note that $\max\{-9, -7\} = -7$.) Figure 5 shows the corresponding projections for the ESTNU from Figure 4b.

When first formalizing dispatchability for ESTNUs, Morris [11] argued that any valid dynamic execution strategy for an ESTNU, when restricted to a particular situation (unknown to the strategy), was equivalent to a

went in the opposite direction: defining an ESTNU to be dispatchable if each of its projections was STN-dispatchable, and then arguing that a dispatchable ESTNU would necessarily be successfully executed by a real-time execution algorithm. Hunsberger and Posenato's work confirmed that one could equivalently mirror the STN approach by first defining ESTNU dispatchability in terms of a performance guarantee with respect to the `RTE`* algorithm, and then proving that the projections of every dispatchable ESTNU would necessarily be STN-dispatchable.



(a) A dispatchable ESTNU

(b) Its projection where $\omega_c = C - A = 4$

Figure 6: A dispatchable ESTNU (left) equivalent to the STNU from Figure 1b and the ESTNU from Figure 4b; and one of its projections (right)

dispatchable execution of the corresponding STN projection (equivalently, to a run of the RTE algorithm on that projection). Therefore, he *defined* the dispatchability of an ESTNU in terms of the dispatchability of its STN projections, as follows.

Definition 7 (ESTNU dispatchability [11])**.** An ESTNU is dispatchable if *all* of its projections are STN-dispatchable.

Since STN-dispatchability can be checked using the graphical characterization in terms of shortest vee-paths (cf. Theorem 1), this definition for STNU-dispatchability is attractive. For example, it is easy to check that the sample ESTNU graph in Figure 4b is not ESTNU-dispatchable since both of its projections shown in Figure 5 are not STN-dispatchable. For example, there is no SVP from C to Y in either projection; and there is no SVP from A to Z in the projection where $C - A = 4$. (Similar remarks apply to the sample STNU from Figure 1b.) In contrast, Figure 6a shows a *dispatchable* ESTNU that is equivalent to both the STNU from Figure 1b and the ESTNU from Figure 4b. Figure 6b shows its projection in the situation where $\omega_c = C - A = 4$, which *is* STN-dispatchable.

Morris [11] described a modification of his $O(n^3)$-time DC-checking algorithm for STNUs so that, when given a DC STNU as input, it generated an equivalent dispatchable ESTNU as output, thereby ensuring that each DC STNU has an equivalent dispatchable ESTNU.[4] Hunsberger and Posenato presented an $O(mn + kn^2 + n^2 \log n)$-time algorithm, called $\texttt{FD}_{\text{STNU}}$, that is faster on sparse ESTNU graphs [6]; they also [9] confirmed the correctness of Morris' insights regarding ESTNU dispatchability by:

1. formally defining a real-time execution algorithm for ESTNUs, called $\texttt{RTE}^*$, that preserves maximal flexibility during execution while requiring minimal real-time computation; and
2. proving that running their $\texttt{RTE}^*$ algorithm on any ESTNU that satisfied Morris' definition of dispatchability (i.e., Definition 7) would necessarily result in an execution sequence that satisfied all of the ESTNU's constraints, no matter how the contingent durations turned out.

That result provides a solid theoretical foundation for ESTNU dispatchability. In addition, Hunsberger and Posenato proved that the $\texttt{RTE}^*$ algorithm has a time complexity of $O(m + nk \log(nk))$, where m is the number of edges, n the number of nodes, and k the number of contingent links.

1.5 Algorithm $\texttt{minDisp}_{\text{ESTNU}}$ for Solving the MinDispESTNU Problem

Since the number of edges directly impacts the computations done during a real-time execution of a ESTNU, more recent work has focused on solving the MinDispESTNU problem: computing an equivalent dispatchable ESTNU having a *minimal number of edges.*

Definition 8 (MinDispESTNU Problem)**.** Given a DC STNU (or ESTNU) $\mathcal{G}$, the *MinDispESTNU problem* is the problem of computing an equivalent dispatchable ESTNU $\mathcal{G}'$ having a minimal number of edges. Such an ESTNU may be called a μESTNU for $\mathcal{G}$.

The first algorithm for solving the MinDispESTNU Problem is the $\texttt{minDisp}_{\text{ESTNU}}$ algorithm [7], whose complexity is $O(kn^3)$-time, where n is the number of timepoints and k the number of contingent links.

1.5.1 Preliminary Observations.

Suppose $\mathcal{G}$ is a dispatchable ESTNU and that $e = (V, \delta, W)$ is an ordinary edge in $\mathcal{G}$. If every projection of $\mathcal{G}$ contains a shortest vee-path (SVP) from V to W whose length is less than δ, then e can be removed from the network without threatening its dispatchability. However, as will be seen, this determination can be complicated by the fact that the SVPs from V to W may follow different routes in different projections. The algorithms presented in this section address this issue by exploring certain structures in the ESTNU graph called *diamond*

[4] In this context, an STNU and ESTNU are called equivalent if they share the same set of valid dynamic execution strategies.

(a) Fixing *weak* and *misleading* waits

(b) Initial stand-in edges

Figure 7: Fixing wait edges and generating initial stand-in edges (dashed)

structures. In addition, they also address how to determine which wait edges can be removed without threatening the network's dispatchability.

Definition 9 (Path lengths in different projections)**.** Let $\mathcal{G}$ be any ESTNU graph; $\mathcal{P}$ any path in $\mathcal{G}$; and ω any situation. If $\mathcal{P}$ has only ordinary edges, then $|\mathcal{P}|$ denotes its length; and $\mathrm{d}(V, W)$ denotes the length of the shortest path from V to W that has only *ordinary* edges. More generally, since the projection of *any* path $\mathcal{P}$ is, by definition, an ordinary path, its length is denoted by $|\mathcal{P}_\omega|$. We may also refer to $|\mathcal{P}_\omega|$ as the length of $\mathcal{P}$ in the situation ω and notate it as $|\mathcal{P}|_\omega = |\mathcal{P}_\omega|$. For any V and W, $\mathrm{d}_\omega(V, W)$ denotes the length of any SVP from V to W in $\mathcal{G}_\omega$; and $d_*(V, W) = \max_\omega\{\mathrm{d}_\omega(V, W)\}$ denotes the maximum such length *across all projections.* If context allows, we may notate a path by listing its timepoints (e.g., a path from V to A to C may be notated as VAC, and the length of its projection as $|VAC|_\omega$).

Using this notation, if $e = (V, \delta, W)$ is an ordinary edge in a dispatchable ESTNU $\mathcal{G}$, then e can be removed from $\mathcal{G}$ without threatening its dispatchability if $d_*(V, W) < \delta$. In addition, if $d_*(V, W) = \delta$, then e can be safely removed if every projection $\mathcal{G}_\omega$ has an SVP from V to W that does not use e.

The $\texttt{minDisp}_{\text{ESTNU}}$ algorithm computes, for each pair of timepoints U and W, the value of $d_*(U, W)$, which can be understood not only as the maximum length of any SVP from U to W across all projections, but also as specifying the strongest ordinary constraint on $W - U$ that must be satisfied by any dynamic execution strategy. The basic approach of $\texttt{minDisp}_{\text{ESTNU}}$ is to generate ordinary "stand-in" edges that are entailed by various combinations of ESTNU edges, especially those entailed by "nested diamond structures". After inserting those stand-in edges into the network, $\texttt{minDisp}_{\text{ESTNU}}$ then uses an STN-dispatchability algorithm on the network's ordinary edges (including the stand-in edges) to determine which ordinary edges can be removed without threatening the ESTNU's dispatchability. It then separately determines which wait edges can be removed.

The $\texttt{minDisp}_{\text{ESTNU}}$ algorithm takes the following steps:

1. it fixes *weak* and *misleading* wait edges;
2. it generates ordinary *stand-in* edges that are entailed by individual labeled edges or possibly-nested diamond structures;
3. it runs an STN-dispatchability algorithm on the resulting set of ordinary edges; and
4. it removes wait edges that are not needed for ESTNU-dispatchability.

Each of these steps is discussed in detail, below.

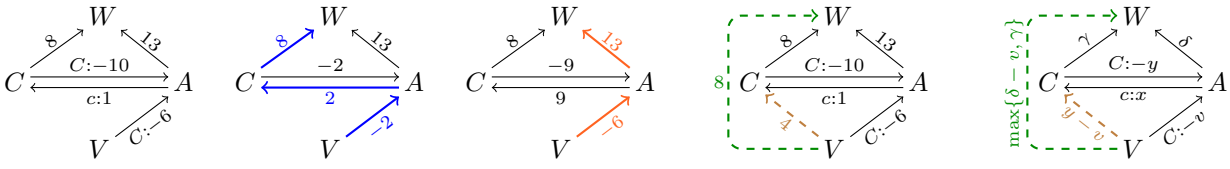
1.5.2 Fixing "weak" and "misleading" wait edges.

In Figure 7a, $(Y, C{:}-2, A)$, shown in red, is a *weak wait*, since the minimum duration of the contingent link $(A, 3, 10, C)$ is 3, which implies that C cannot execute before the wait time of 2 expires. So this wait is effectively unconditional and hence can be *replaced* by the ordinary edge $(Y, -2, A)$, shown in green. In general, the *Unconditional Unordered Reduction* rule [13] ensures that any wait edge $(Y, C{:}-v, A)$ where (A, x, y, C) is the relevant contingent link can be replaced by the ordinary edge $(Y, -v, A)$ if $v \leq x$ (i.e., the wait time is less than the minimum duration of the contingent link). In contrast, $(W, C{:}-15, A)$, also shown in red, is a *misleading wait* since C *must* execute no later than 10 after A. This wait is fixed by changing the wait time to the maximum of 10: $(W, C{:}-10, A)$, shown in green. In general, any wait edge $(W, C{:}-w, A)$ where (A, x, y, C) is the relevant contingent link can be replaced by the wait edge $(W, C{:}-y, A)$ if $w > y$ (i.e., the wait time is greater than the maximum duration of the contingent link).

Definition 10 (Regular wait edge)**.** A wait edge $(W, C{:}-v, A)$ associated with a contingent link (A, x, y, C) is called *regular* if it is neither weak nor misleading (i.e., if its wait amount satisfies $x < v \leq y$).

1.5.3 Generating Stand-in Edges.

After fixing all weak and misleading waits, the $\texttt{minDisp}_{\text{ESTNU}}$ algorithm focuses on generating *stand-in* edges. Each stand-in edge is an *ordinary* edge that is used to make explicit a constraint entailed by individual labeled edges or combinations of ESTNU edges across different STN projections. The stand-in edges are subsequently used to determine what other edges can be removed from the network without threatening its dispatchability.



(a) ESTNU (b) $\omega_c = 2$ (c) $\omega_c = 9$ (d) Dashed stand-in edges (e) General case

Figure 8: A sample ESTNU, two of its projections with (colored) SVPs, and two (dashed) stand-in edges

Afterward, the stand-in edges are eventually removed from the network—because they are, by definition, entailed by other paths in the network.

Initial stand-in edges. Figure 7b shows (dashed) stand-in edges that represent the strongest ordinary constraints that are entailed by various labeled ESTNU edges associated with the contingent link $(A, 3, 10, C)$. The stand-in edge for the LC edge $(A, c{:}3, C)$ is $(A, 10, C)$, representing that $C - A \leq 10$ in every projection. The stand-in edge for the UC edge $(A, C{:}{-}10, C)$ is $(C, -3, A)$, representing that $A - C \leq -3$ (i.e., $C - A \geq 3$) in every projection. For the wait edge $(V, C{:}{-}6, A)$, the stand-in edge is $(V, -3, A)$, representing that V must *unconditionally* wait at least 3 after A, since C cannot execute sooner than that. These observations motivate the following definition.

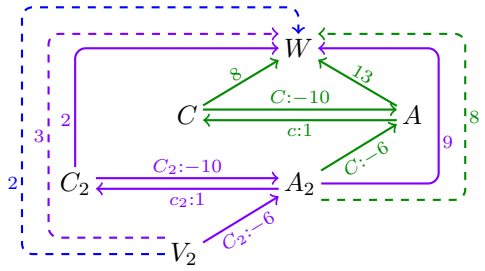
Definition 11 (Stand-in edges for individual labeled edges)**.** Let (A, x, y, C) be any contingent link. The *stand-in* edge for the LC edge $(A, c{:}x, C)$ is the ordinary edge (A, y, C); the stand-in edge for the UC edge $(C, C{:}{-}y, A)$ is the ordinary edge $(C, -x, A)$; and the stand-in edge for the regular wait edge $(W, C{:}{-}v, A)$ is the ordinary edge $(W, -x, A)$.

The VAC rule. In Figure 7b, there is one more (dashed) stand-in edge, $(V, 4, C)$, shown in brown. It is entailed by the two-edge path comprising the wait edge followed by the LC edge. In particular, in each projection, $|VAC|_\omega = \max\{-6, -\omega_c\} + \omega_c = \max\{\omega_c - 6, 0\} \leq 4$, since $\omega_c \leq 10$. In general, for any regular wait edge $(V, C{:}{-}v, A)$ and associated contingent link (A, x, y, C), the *VAC rule* generates the ordinary edge $(V, y - v, C)$.

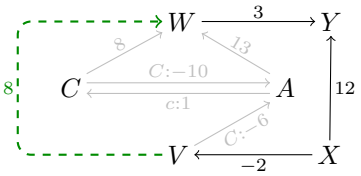
Stand-in edges for diamond structures. Although the kinds of stand-in edges discussed above are needed by the $\texttt{minDisp}_{\text{ESTNU}}$ algorithm, stand-in edges can also be entailed by other structured combinations of edges, which we call *diamond* structures. For example, consider the ESTNU in Figure 8a. Since it has only one contingent link, each projection ω is determined by the value ω_c that is assigned to $C - A$. In the projection where $\omega_c = 2$, shown in Figure 8b, the SVP from V to W, indicated by thick blue edges, has length 8, whereas in the projection shown in Figure 8c, where $\omega_c = 9$, the SVP from V to W, indicated by thick orange edges, has length 7. It is not hard to check that for this ESTNU, the length of the SVP from V to W, across all projections, is at most 8. In particular, in projections where $\omega_c \leq 5$, $|VACW|_\omega = \max\{-\omega_c, -6\} + \omega_c + 8 = \max\{8, \omega_c + 2\} \leq 8$; but in projections where $\omega_c \geq 5$, $|VAW|_\omega = \max\{-\omega_c, -6\} + 13 = \max\{13 - \omega_c, 7\} \leq 8$. Hence, this combination of ordinary and labeled edges entails the stand-in edge $(V, 8, W)$, shown as green and dashed in Figure 8d. For reference, the (brown and dashed) stand-in edge $(V, 4, C)$ generated by the VAC rule, described earlier, is also shown. Figure 8e shows the general case of deriving a (green and dashed) stand-in edge for a $VACW$ diamond, as well as the (brown and dashed) stand-in edge generated by the VAC rule. As shown in the formal analysis in Section 2, the length of the stand-in edge is $\max\{\delta - v, \gamma\}$, where $\delta - v$ is the length of the path VAW in the projection $\omega_c = y$; and γ is the length of $VACW$ in the projection $\omega_c = x$. In addition, the value, $\theta = \delta - \gamma$, plays a key role. In particular, the length of the stand-in edge is γ if $\theta \in (x, v]$, but is $\delta - v$ if $\theta \in (v, y]$. If $\theta \notin (x, y]$, then the stand-in edge is not needed.

Stand-in edges for nested diamonds. Figure 9a shows a more complicated ESTNU, where the diamond structure involving the solid green edges is nested inside the diamond structure involving the solid purple edges. Ignoring the green edges, for now, the solid purple edges can be shown to entail the (purple, dashed) stand-in edge $(V_2, 3, W)$. In particular, in projections where $\omega_2 = C_2 - A_2 \leq 7$, the length of the path $V_2 A_2 C_2 W$ is: $\max\{-\omega_2, -6\} + \omega_2 + 2 = \max\{2, \omega_2 - 4\} \leq 3$. In contrast, if $\omega_2 \geq 7$, the length of the alternative path $V_2 A_2 W$ is: $\max\{-\omega_2, -6\} + 9 = \max\{9 - \omega_2, 3\} \leq 3$.

Next, since the green diamond is isomorphic to the diamond from Figure 8a, it entails the (green, dashed) stand-in edge $(A_2, 8, W)$. But now, using that stand-in edge instead of the purple edge $(A_2, 9, W)$, a new analysis of the purple structure shows that it now entails a stronger (blue, dashed) stand-in edge $(V_2, 2, W)$. In other words, nested diamond structures can sometimes combine to entail stronger stand-in edges. Not only that, the order in which nested diamonds are analyzed can affect the overall computational effort required. For example, if the green diamond is analyzed first, then the purple diamond only needs to be analyzed once.

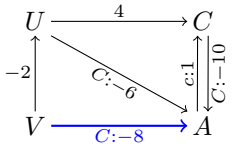


(a) Stand-in edges entailed by nested diamonds

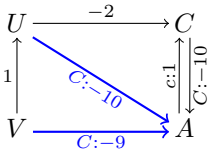


(b) Using a stand-in edge to dominate $(X, 12, Y)$

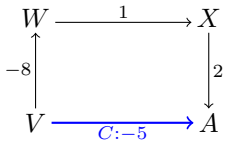
Figure 9: Stand-in edges entailed by nested diamonds; and using a stand-in edge to dominate another edge



(a) Dominated by another wait



(b) Dominated by a UC edge



(c) Dominated by an ordinary path

Figure 10: Examples of (blue) wait edges that can be removed without threatening ESTNU dispatcability

The importance of stand-in edges. The network in Figure 9b includes the diamond structure from Figure 8a (with its original ESTNU edges drawn in light gray), its (green, dashed) stand-in edge $(V, 8, W)$, and some additional edges. Assuming that the stand-in edge is present in the network, then the *vee-path* $XVWY$, which contains only ordinary edges, has length $-2+8+3 = 9$, which is less than the length of the edge $(X, 12, Y)$. That implies that the edge $(X, 12, Y)$ can be removed from the network without affecting its dispatchability. Crucially, this conclusion depended solely on ordinary edges and, therefore, in general, can be determined by applying an STN-dispatchability algorithm to those ordinary edges. (That is Step 3 of the $\texttt{minDisp}_{\text{ESTNU}}$ algorithm.) Returning to Figure 9b, once the edge $(X, 12, Y)$ has been removed, the stand-in edge has played its role and can be discarded because, in each projection, either $XVAWY$ or $XVACWY$ will provide an SVP from X to Y.

1.5.4 Removing Unneeded Wait Edges.

Computing a μESTNU also requires determining which wait edges can be removed without threatening the dispatchability of the network. There are three cases: wait edges that are dominated by

1. other wait edges;
2. UC edges;
3. ordinary paths.

Waits dominated by other waits. Figure 10a shows an ESTNU having a (blue) wait edge $(V, C{:}-8, A)$ that is dominated by another wait edge $(U, C{:}-6, A)$. In particular, in any situation $\omega_c \in [1, 10]$, $|VUA|_{\omega_c} = -2 + \max\{-6, -\omega_c\} = \max\{-8, -2-\omega_c\} \leq \max\{-8, -\omega_c\} = |(V, C{:}-8, A)|_{\omega_c}$. The key features behind this derivation are that the path from V to U (in this case, a single edge) has negative length (i.e., $\mathrm{d}(V, U) < 0$) and $\mathrm{d}(V, U) - u \leq -v$. In general, $|VUA|_{\omega_c} = \mathrm{d}(V, U) + \max\{-u, -\omega_c\} = \max\{\mathrm{d}(V, U) - u, \mathrm{d}(V, U) - \omega_c\} \leq \max\{-v, -\omega_c\} = |(V, C{:}-v, A)|_{\omega_c}$.

Waits dominated by UC edges. Figure 10b shows two (blue) wait edges, $(U, C{:}-10, A)$ and $(V, C{:}-9, A)$, that are dominated by the UC edge $(C, C{:}-10, A)$ via the paths, UCA and $VUCA$, respectively.[5] For each $\omega_c \in [1, 10]$, $|UCA|_{\omega_c} = -2 + |(C{:}-10, A)|_{\omega_c} = -2 - \omega_c < -\omega_c \leq \max\{-\omega_c, -10\} = |(U, C{:}-10, A)|_{\omega_c}$. The key feature behind this derivation is that the path from U to C (in this case, a single edge) has negative length (i.e., $\mathrm{d}(U, C) = -2 < 0$). It can similarly be shown that $|VUCA|_{\omega_c} \leq |(V, C{:}-9, A)|_{\omega_c}$ for each $\omega_c \in [1, 10]$, again, the key feature being that $\mathrm{d}(V, C) < 0$.

Waits dominated by ordinary paths. In Figure 10c, the (blue) wait $(V, C{:}-5, A)$ is dominated by the ordinary path $VWXA$, since for each ω_c, $|VWXA|_{\omega_c} = -5 \leq \max\{-5, -\omega_c\} = |(V, C{:}-5, A)|_{\omega_c}$.

[5]The Upper-Case rule [14] would generate the *misleading* wait edges, $(U, C{:}-12, A)$ and $(V, C{:}-11, A)$. Fixing the first wait would yield $(U, C{:}-10, A)$, since 10 is the maximum duration of the contingent link. Afterward, the Upper-Case rule could be used to generate the wait edge $(V, C{:}-9, A)$. However, even if the wait edges are not fixed, a similar analysis shows that they would be still be dominated by the UC edge.

Algorithm 1: $\texttt{minDisp}_{\text{ESTNU}}$: Solving the MinDispESTNU problem

Input: $\mathcal{G} = (\mathcal{T},\ \mathcal{E}_o \cup \mathcal{E}_l \cup \mathcal{E}_u \cup \mathcal{E}_w)$, dispatchable ESTNU
Output: A μESTNU for $\mathcal{G}$

1 $(\mathcal{E}_o^{\text{si}}, d) := \texttt{genStandIns}(\mathcal{T},\ \mathcal{E}_o \cup \mathcal{E}_l \cup \mathcal{E}_u \cup \mathcal{E}_w)$ *//Compute the set of (ordinary) stand-in edges*
2 $(\mathcal{T}, \mathcal{E}_o^*, \hat{\mathcal{E}}_{\text{l}}, \hat{\mathcal{E}}_{\text{u}}, \hat{\mathcal{E}}_{\text{w}}) := \texttt{disp}_{\text{STN}}(\mathcal{T}, \mathcal{E}_o \cup \mathcal{E}_o^{\text{si}}, \mathcal{E}_l, \mathcal{E}_u, \mathcal{E}_w)$ *//STN dispatchability on ordinary edges, reorienting labeled edges*
3 $\hat{\mathcal{E}}_o^* := \mathcal{E}_o^* \backslash \mathcal{E}_o^{\text{si}}$ *//Remove any remaining stand-in edges from* $\mathcal{E}_o^*$
4 $\hat{\mathcal{E}}_{\text{w}} := \hat{\mathcal{E}}_{\text{w}} \backslash\ \texttt{markWaits}(\mathcal{T}_c, \hat{\mathcal{E}}_{\text{w}}, d)$ *//Remove dominated waits*
5 **return** $\mathcal{G} = (\mathcal{T}, \hat{\mathcal{E}}_o^* \cup \hat{\mathcal{E}}_{\text{l}} \cup \hat{\mathcal{E}}_{\text{u}} \cup \hat{\mathcal{E}}_{\text{w}})$

These observations motivate the following definition.

Definition 12 (Dominated wait edge)**.** Let $E = (V, C{:}{-}v, A)$ be any regular wait edge, where (A, x, y, C) is the relevant contingent link. E is *dominated* by an ordinary path $\mathcal{P}$ from V to A if $|\mathcal{P}| \leq -v$. E is *dominated* by the UC edge $(C, C{:}{-}y, A)$ if $d(V, C) < 0$. E is *dominated* by another wait edge $(U, C{:}{-}u, A)$ if $d(V, U) < 0$ and $d(V, U) - u \leq -v$.

1.5.5 High-level Pseudocode for the $\texttt{minDisp}_{\text{ESTNU}}$ Algorithm.

This section presents a high-level pseudocode for the $\texttt{minDisp}_{\text{ESTNU}}$ algorithm (cf. Algorithm 1). It takes a dispatchable ESTNU $\mathcal{G} = (\mathcal{T}, \mathcal{E}_o \cup \mathcal{E}_l \cup \mathcal{E}_u \cup \mathcal{E}_w)$ as input, and generates as output a μESTNU (i.e., an equivalent dispatchable ESTNU with a minimal number of edges). Since the input is dispatchable, the algorithm need only determine which edges can be removed while preserving dispatchability.[6]

The pseudocode has four steps:

1. It computes $\mathcal{E}_o^{\text{si}}$, the set of (ordinary) stand-in edges entailed by both individual labeled edges and (possibly nested) diamond structures. These stand-in edges only play a supporting role in determining which other edges can be safely removed from the network. Since the stand-in edges are, by definition, entailed by other edges, they will also be removed before the end of the algorithm.

2. The STN-based dispatchability algorithm [16], called $\texttt{disp}_{\text{STN}}$ here, is used to transform the STN $(\mathcal{T}, \mathcal{E}_o \cup \mathcal{E}_o^{\text{si}})$ into an equivalent dispatchable STN $(\mathcal{T}, \mathcal{E}_o^*)$ with a minimal number of edges (for the STN). Since $\texttt{disp}_{\text{STN}}$ computes so-called *rigid components* (RCs) and may *re-orient* some edges to point at *representative* timepoints of those RCs, the version of $\texttt{disp}_{\text{STN}}$ used here will, if necessary, re-orient labeled edges to similarly point at representative timepoints, thereby preserving the equivalence of the resulting ESTNU.

3. If any stand-in edges remain in $\mathcal{E}_o^*$, they are removed.

4. All wait edges that are dominated by (1) ordinary paths; (2) UC edges; or (3) other waits are removed from the network.

Finally, $\texttt{minDisp}_{\text{ESTNU}}$ outputs the μESTNU.

2 The Canonical Form of Nested Diamond Structures

This section presents a novel theory of the *canonical form of nested diamond structures* that provides a new understanding of ESTNU dispatchability. It begins by recalling and elaborating some previously published results (cf. Lemmas 1 to 5) [8]. Then, it presents the main theory, which reveals numerous properties of nested diamonds (cf. Theorem 2) that make the proofs of Theorems 3 and 4 more rigorous and comprehensive as compared to the proof sketches in prior work [8].

Lemma 1 (Fixing Weak/Misleading Waits)**.** *Let (A, x, y, C) be any contingent link. Each* weak *wait, $E = (V, C{:}{-}v, A)$ where $v \leq x$, is equivalent to its ordinary replacement $\hat{E} = (V, -v, A)$, since for each value of $\omega_c \in [x, y]$, $|E|_{\omega_c} = -v = |\hat{E}|_{\omega_c}$ (i.e., they have the same length in each projection). Similarly, each* misleading *wait, $F = (U, C{:}{-}u, A)$ where $u > y$, is equivalent to its fixed version $\hat{F} = (U, C{:}{-}y, A)$, since for each $\omega_c \in [x, y]$, $|F|_{\omega_c} = -\omega_c = |\hat{F}|_{\omega_c}$.*

Proof. In each situation $\omega_c = C - A \in [x, y]$, $|E|_{\omega_c} = \max\{-v, -\omega_c\} = -v = |\hat{E}|$, since $-\omega_c \leq -x \leq -v$. Similarly, $|F|_{\omega_c} = \max\{-u, -\omega_c\} = -\omega_c = \max\{-y, -\omega_c\} = |\hat{F}|_{\omega_c}$, since $-\omega_c \geq -y > -u$. □

[6] For any given STNU, the $\texttt{FD}_{\text{STNU}}$ algorithm [6] can be used to generate an initial equivalent dispatchable STNU; or $\perp$ if the STNU is not DC.

Lemma 2 (Stand-in Edges associated with a contingent link)**.** *Let (A, x, y, C) be any contingent link; and let $\omega_c = C - A \in [x, y]$. Then:*

1. *the LC edge $e = (A, c{:}x, C)$ and its stand-in edge $\hat{e} = (A, y, C)$ satisfy: $|e|_{\omega_c} \leq |\hat{e}|$, with equality when $\omega_c = y$;*
2. *the UC edge $E = (C, C{:}{-}y, A)$ and its stand-in edge $\hat{E} = (C, -x, A)$ satisfy: $|E|_{\omega_c} \leq |\hat{E}|$, with equality when $\omega_c = x$; and*
3. *any regular wait edge $F = (V, C{:}{-}v, A)$ and its stand-in edge $\hat{F} = (V, -x, A)$ satisfy: $|F|_{\omega_c} \leq |\hat{F}|$, with equality when $\omega_c = x$.*

Proof. 1. $|e|_{\omega_c} = \omega_c \leq y = |\hat{e}|$, with equality when $\omega_c = y$;
2. $|E|_{\omega_c} = -\omega_c \leq -x = |\hat{E}|$, with equality when $\omega_c = x$; and
3. $|F|_{\omega_c} = \max\{-v, -\omega_c\} \leq -x = |\hat{F}|$, since $-v \leq -x$ for a regular wait edge, and $-\omega_c \leq -x$.

□

Lemma 3 (VAC rule)**.** *Let $e = (A, c{:}x, C)$ be the LC edge for a contingent link (A, x, y, C) and let $E_w = (V, C{:}{-}v, A)$ be any regular wait edge labeled by C. Let $\mathcal{P}$ be the two-edge path consisting of E_w followed by e. Then $\mathcal{P}$ entails the ordinary edge $(V, y - v, C)$.*

Proof. In any situation ω, $|\mathcal{P}|_\omega = |E_w|_\omega + |e|_\omega = \max\{-v, -\omega_c\} + \omega_c = \max\{\omega_c - v, 0\} \leq \max\{y - v, 0\} = y - v = |(V, y - v, C)|$. (Since E_w is non-misleading, $y - v \geq 0$.) □

Henceforth, this section assumes that all waits are fixed and that the ESTNU $\mathcal{G}$ includes all of the stand-in edges derived from individual labeled edges and the VAC rule, as described above.

Definition 13 (Labeled edge sets, contingent links, and ESTNU subgraphs)**.** Let $\mathcal{G}$ be any ESTNU graph; (A, x, y, C), any contingent link in $\mathcal{G}$; Λ, any set of labeled edges in $\mathcal{G}$; and U and W, any timepoints.

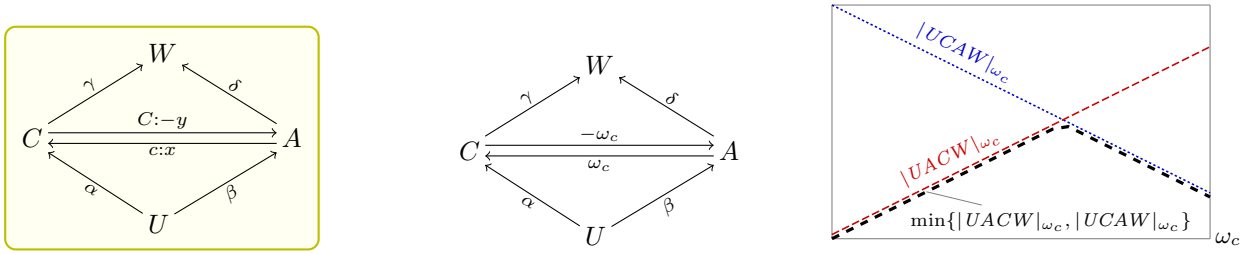
- An LC, UC, or wait edge that is labeled by c or C is said to be *associated* with the contingent link (A, x, y, C).
- If Λ contains at least one labeled edge associated with (A, x, y, C), then (A, x, y, C) is said to be *represented* in Λ. The total number of contingent links represented in Λ is denoted by $||\Lambda||$. (Note that $||\Lambda|| = 0$ if and only if $\Lambda = \emptyset$.)
- $\Lambda^{-(A,x,y,C)}$ denotes the set of labeled edges that is the same as Λ, except that it does not include any labeled edges associated with (A, x, y, C). If the context makes the choice of a contingent link clear, then we may simply write Λ^-.
- The ESTNU *subgraph induced by* Λ is the ESTNU graph $\mathcal{G}^\Lambda$ that is the same as $\mathcal{G}$ except that the only labeled edges in $\mathcal{G}^\Lambda$ are those in Λ. The timepoints and ordinary edges in $\mathcal{G}^\Lambda$ (including all standin edges) are the same as those in $\mathcal{G}$.
- A situation ω for $\mathcal{G}^\Lambda$ specifies durations for each contingent link represented in Λ. For any such ω,
  - $d^\Lambda_\omega(U, W) = \min\{|\mathcal{P}|_\omega$ such that $\mathcal{P}$ is a path from U to W in $\mathcal{G}^\Lambda\}$; and
  - $d^\Lambda_*(U, W) = \max_\omega\{d^\Lambda_\omega(U, W)\}$.

  If ω is such that $d^\Lambda_\omega(U, W) = d^\Lambda_*(U, W)$ (see Definition 9), then ω is said to be *maximal* for $d^\Lambda_*(U, W)$.
- If $d^\Lambda_*(U, W) = d_*(U, W)$, then the edges in Λ are said to be *sufficient* for $d_*(U, W)$. If, in addition, no proper subset of Λ has this property, then Λ is said to be *minimally sufficient* for $d_*(U, W)$. Similarly, if $\Lambda_1 \subseteq \Lambda_2$ and $d^{\Lambda_1}_*(U, W) = d^{\Lambda_2}_*(U, W)$, then the edges in Λ_1 are *sufficient* for $d^{\Lambda_2}_*(U, W)$ and if, in addition, no proper subset of Λ_1 has this property, then Λ_1 is *minimally sufficient* for $d^{\Lambda_2}_*(U, W)$.

In the case where $\mathcal{G}^\Lambda = \mathcal{G}$, the definitions of d^Λ_ω and d^Λ_* reduce to d_ω and d_*, respectively. And if $\Lambda = \emptyset$ is minimally sufficient for $d_*(U, W)$, then $d_*(U, W)$ is determined solely by ordinary edges and hence equals $d(U, W)$.

Lemma 4. *Let U and W be any timepoints in a dispatchable ESTNU $\mathcal{G}$; Λ, a set of labeled edges that are minimally sufficient for $d_*(U, W)$; and (A, x, y, C), a contingent link represented in Λ.*
If (A, x, y, C) is the only *contingent link represented in Λ, then for some wait edge $(V, C{:}{-}v, A)$, it holds that $\Lambda = \{(A, c{:}x, C), (V, C{:}{-}v, A)\}$, and*

$$d_*(U, W) \;=\; \mathrm{d}(U, V) + \max\{\mathrm{d}(A, W) - v, \mathrm{d}(C, W)\}.$$



(a) A sample diamond (b) That diamond in the situation ω_c (c) Path-length functions vs. $\omega_c \in [x, y]$

Figure 11: The diamond structure considered in Case 3 of Lemma 4

Proof. Given the premise of the lemma, $d_*(U, W) = d_*^\Lambda(U, W)$, and every projection must have a simple SVP from U to W that derives from an ESTNU path involving zero or more ordinary edges and *at least one* edge labeled by c or C. (If some projection had an SVP from U to W deriving from only ordinary edges in the ESTNU, it would contradict that Λ is minimally sufficient.)

The following cases show that Λ must include the LC edge $e = (A, c{:}x, C)$ and at least one wait edge $E_v = (V, C{:}{-}v, A)$, but not the UC edge $E = (C, C{:}{-}y, A)$.

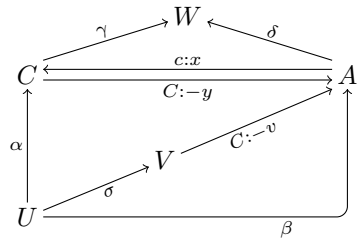
Case 1: $\Lambda = \{e\}$ (i.e., e is the only needed labeled edge). Let ω be the situation in which $C - A = y$, the maximum duration. Since Λ is minimally sufficient for $d_*(U, W)$, there must be some ESTNU path $\mathcal{P}$ from U to W whose only labeled edge is e and for which $|\mathcal{P}|_\omega \leq d_*(U, W)$. Let $\mathcal{P}^{\text{si}}$ be the *ordinary* path that is the same as $\mathcal{P}$ except that e has been replaced by its stand-in edge $\hat{e} = (A, y, C)$. By Lemma 2, $|e|_\omega = y = |\hat{e}|$, whence $|\mathcal{P}^{\text{si}}| = |\mathcal{P}|_\omega \leq d_*(U, W)$, contradicting that Λ is minimally sufficient for $d_*(U, W)$.

Case 2: $e \notin \Lambda$ (i.e., e is not *needed).* Consider the situation ω where $C - A = x$, the minimum duration. There must be some simple ESTNU path $\mathcal{P}$ from U to W whose only labeled edge is either the UC edge or a wait edge labeled by C, and for which $|\mathcal{P}|_\omega \leq d_*(U, W)$. (Since the UC edge and all wait edges labeled by C terminate at A, there can only be one such edge in any simple path.) Let $\mathcal{P}^{\text{si}}$ be the *ordinary* path that is the same as $\mathcal{P}$ except that the UC or wait edge has been replaced by its standin edge. By Lemma 2, $|\mathcal{P}^{\text{si}}| = |\mathcal{P}|_\omega \leq d_*(U, W)$, contradicting that Λ is minimally sufficient for $d_*(U, W)$.

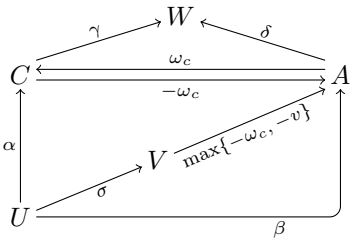
Case 3: $\Lambda = \{e, E\}$ (i.e., no wait edges are needed). For each situation ω, there must be an ESTNU path $\mathcal{P}$ from U to W that contains e or E (not both) and such that $\mathcal{P}_\omega$ is a simple shortest path in $\mathcal{G}_\omega$ for which $|\mathcal{P}|_\omega \leq d_*(U, W)$. (No simple path can include both e and E.) Hence, each such path must be either $UACW$ or $UCAW$, as shown in Figure 11a, where the arrows labeled by Greek letters represent shortest ordinary paths from the ESTNU. In any given projection, the lengths of $UACW$ and $UCAW$ are given by $|UACW|_{\omega_c} = \beta + \omega_c + \gamma$ and $|UCAW|_{\omega_c} = \alpha - \omega_c + \delta$, as illustrated in Figure 11b. Note that $|UCAW|_{\omega_c}$ decreases with ω_c, while $|UACW|_{\omega_c}$ increases with ω_c, as shown in Figure 11c. Meanwhile, the paths UCW and UAW, which comprise only ordinary edges, must satisfy: $\alpha + \gamma = |UCW| > d_*(U, W)$ and $\beta + \delta = |UAW| > d_*(U, W)$; otherwise, they would contradict that Λ is minimally sufficient. Now, since both the LC and UC edges are needed for $d_*(U, W)$, and $|UACW|_{\omega_c}$ increases with ω_c, while $|UCAW|_{\omega_c}$ decreases, it follows that the simple shortest from U to W must be $UACW$ for smaller values of ω_c, and $UCAW$ for larger values. Therefore, the *maximum* length of any such shortest path from U to W across any value of ω_c must occur where $|UCAW|_{\omega_c} = |UACW|_{\omega_c}$, as shown in Figure 11c. That intersection occurs when $\omega_c = \hat{\omega}_c = \frac{1}{2}(\alpha + \delta - \beta - \gamma)$. In addition, the value of $\hat{\omega}_c$ must lie within the interval $[x, y]$, since otherwise only one of the labeled edges would be needed, contradicting that Λ is minimally sufficient. But that maximum value is $|UCAW|_{\hat{\omega}_c} = |UACW|_{\hat{\omega}_c} = \beta + \hat{\omega}_c + \gamma = \beta + \frac{\alpha+\delta-\beta-\gamma}{2} + \gamma = \frac{(\alpha+\gamma)+(\beta+\delta)}{2} = \frac{|UCW|+|UAW|}{2} > \frac{d_*(U,W)+d_*(U,W)}{2} = d_*(U, W)$, a contradiction.

End of Cases 1-3.

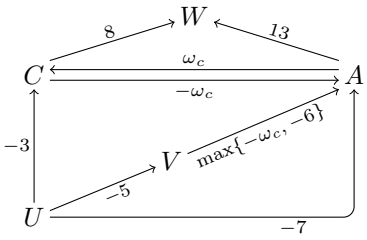
From Cases 1–3 above, it follows that Λ must include e and at least one wait edge $E_v = (V, C{:}{-}v, A)$. For now, suppose that E_v is the only wait edge in Λ. Then, as illustrated in Figures 12a and 12b, where the arrows labeled by Greek letters represent shortest ordinary paths, each SVP from U to W in any projection $\mathcal{G}_\omega$ must, by construction, be one of the four paths, $UVAW$, $UACW$, $UVACW$ or $UCAW$, whose lengths are functions of $\omega_c = C - A$, as listed in Table 1. Figure 12c shows a sample instance with numbers instead of Greek letters; and Figure 12d plots the corresponding length functions.



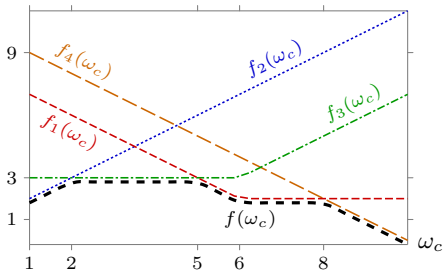
(a) ESTNU paths in a diamond structure



(b) The same paths in a projection, ω_c



(c) A sample projection with numbers



(d) Plots of path-length functions; $\omega_c \in [1, 10]$

Figure 12: ESTNU paths and projections considered in the proof of Lemma 4

$$
\begin{aligned}
f_1(\omega_c) &= |UVAW|_\omega &&= \sigma + \max\{-v, -\omega_c\} + \delta \\
f_2(\omega_c) &= |UACW|_\omega &&= \beta + \omega_c + \gamma \\
f_3(\omega_c) &= |UVACW|_\omega &&= \sigma + \max\{-v, -\omega_c\} + \omega_c + \gamma \\
f_4(\omega_c) &= |UCAW|_\omega &&= \alpha - \omega_c + \delta \\
f(\omega_c) &= \min\{f_i(\omega_c)\}
\end{aligned}
$$

Table 1: Length functions for the paths in Figure 12

Given that both e and E_v are needed for $d_*(U, W)$, the following facts are entailed.

Fact 1. $d_*(U, W) < |UCW| = \alpha + \gamma$, since $\alpha + \gamma$ is the length of the ordinary path UCW.

Fact 2. $d_*(U, W) < |UAW| = \beta + \delta$, since $\beta + \delta$ is the length of the ordinary path VAW.

Fact 3. $\beta > \sigma - v$, since otherwise $|UA|_{\omega_c} = \beta \leq \sigma - v \leq \sigma + \max\{-v, -\omega_c\} = |UVA|_{\omega_c}$, contradicting the need for the wait edge, $(V, C{:}{-}v, A)$.

Fact 4. $\alpha > \sigma$, since otherwise $|UCA|_{\omega_c} = \alpha - \omega_c \leq \sigma - \omega_c \leq \sigma + \max\{-v, -\omega_c\} = |UVA|_{\omega_c}$, contradicting the need for the wait edge $(V, C{:}{-}v, A)$.

Now, each of the length functions in Table 1 is continuous, piecewise-linear, and monotone (i.e., non-decreasing or non-increasing). Thus, the *minimum* of the length functions, $f(\omega_c) = \min_{1 \leq i \leq 4}\{f_i(\omega_c)\}$ (black and dashed in Figure 12d) must be continuous and piecewise linear, although not necessarily monotone. So the *maximum* value of $f(\omega_c)$ (i.e., $d_*(U, W)$) must occur at an endpoint of one of the piecewise-linear segments of $f(\omega_c)$ (i.e., where two of the length functions intersect—or where $\omega_c = x$ or $\omega_c = y$). The following claims show that f_1 and f_3 suffice to determine the value of $d_*(U, W)$.

Claim 1. The values of $f_2(x), f_2(y), f_4(x)$ and $f_4(y)$ have no bearing on the value of $d_*(U, W)$.

Proof of Claim 1.

- $f_4(x) = \alpha - x + \delta$ equals the length of the *ordinary* path $UCAW$, using the stand-in edge $(C, -x, A)$; so if it were equal to $d_*(U, W)$, it would contradict that Λ was minimally sufficient.
- Similarly, $f_2(y) = \beta + y + \gamma$ equals the length of the *ordinary* path $UACW$, using the stand-in edge (A, y, C); so if it were equal to $d_*(U, W)$, it would contradict that Λ was minimally sufficient.

- If $f_2(x)$ were equal to $d_*(U,W)$, and less than all other $f_i(x)$ values, then $f_2(\omega_c)$ would have to equal $f(\omega_c)$ on some interval $[x, x+\epsilon]$; but since f_2 is an increasing function, $f_2(x)$ could not be the maximum value of f on $[x,y]$.
- Similarly, if $f_4(y)$ were equal to $d_*(U,W)$, and less than all other $f_i(y)$ values, then $f_4(\omega_c)$ would have to equal $f(\omega_c)$ on some interval $[y-\epsilon, y]$, but since f_4 is a decreasing function, $f_4(y)$ could not be the maximum value of f on $[x,y]$.

Next, let $\mathcal{I}$ be the set of values of ω_c at all points of intersection among the length functions $f_i(\omega_c)$ listed in Table 1. For each i and j with $1 \le i < j \le 4$, let τ_{ij} denote the value of ω_c where $f_i(\omega_c) = f_j(\omega_c)$. For example, τ_{13} denotes the value of ω_c at which $f_1(\omega_c) = f_3(\omega_c)$ (i.e., where $|UVAW|_{\omega_c} = |UVACW|_{\omega_c}$). Intersection points that depend on the projected length of the wait edge (i.e., $\max\{-v, -\omega_c\}$) are given the superscript v (for the case where $-v \ge -\omega_c$) or ω (for $-\omega_c \ge -v$). It is easy but tedious to compute all eight possible points of intersection:[7]

$$\tau_{12}^v = \sigma+\delta-v-\beta-\gamma \qquad \tau_{12}^\omega = \tfrac{\sigma+\delta-\beta-\gamma}{2} \qquad \tau_{13} = \delta-\gamma \qquad \tau_{14}^v = \alpha-\sigma+v$$
$$\tau_{23}^\omega = \sigma-\beta \qquad \tau_{24} = \tfrac{\alpha+\delta-\beta-\gamma}{2} \qquad \tau_{34}^v = \tfrac{\alpha+\delta-\sigma+v-\gamma}{2} \qquad \tau_{34}^\omega = \alpha+\delta-\sigma-\gamma$$

Claim 2. For each $\tau \in \mathcal{I}$, $f(\tau) = \min\{f_1(\tau), f_3(\tau)\} = \min\{|UVAW|_\tau, |UVACW|_\tau\}$ (i.e., the values of $f_2(\tau)$ and $f_4(\tau)$ are irrelevant).

Proof of Claim 2. We show the proof for just a few of the eight cases, leaving the rest to the reader.

(τ_{12}^v) This is the case where $f_1 = f_2$ and $-v \ge -\omega_c$, yielding $\tau_{12}^v = \sigma+\delta-v-\beta-\gamma$. If $f(\tau_{12}^v)$ equals $f_1(\tau_{12}^v) = f_2(\tau_{12}^v)$ or $f_3(\tau_{12}^v)$, then the claim holds. It only remains to show that the minimum value is *not* $f_4(\tau_{12}^v) = \alpha - (\sigma - v + \delta - \beta - \gamma) + \delta = (\alpha+\gamma) + (\beta - \sigma + v)$. Since $(\alpha+\gamma) > d_*(U,W)$, by Fact 1; and $(\beta-\sigma+v) > 0$, by Fact 3, we get that $f_4(\tau_{12}^v) > d_*(U,W)$.

(τ_{12}^ω) This is the case where $f_1 = f_2$ and $-\omega_c \ge -v$, yielding $\tau_{12}^\omega = \frac{\sigma+\delta-\beta-\gamma}{2}$. As above, it suffices to show that the minimum value is *not* $f_4(\tau_{12}^\omega) = \alpha - \tau_{12}^\omega + \delta = \frac{\alpha-\sigma}{2} + \frac{\alpha+\gamma}{2} + \frac{\beta+\delta}{2}$. Now, $(\alpha-\sigma) > 0$, by Fact 4; $(\alpha+\gamma) > d_*(U,W)$, by Fact 1; and $(\delta+\beta) > d_*(U,W)$, by Fact 2. Hence, $f_4(\tau_{12}^\omega) > \frac{\alpha+\gamma}{2} + \frac{\delta+\beta}{2} > \frac{d_*(U,W)}{2} + \frac{d_*(U,W)}{2} = d_*(U,W)$.

(τ_{13}) This is the case where $f_1 = f_3$, yielding $\tau_{13} = \delta - \gamma$. Then $f_2(\tau_{13}) = \beta + \tau_{13} + \gamma = \beta + (\delta-\gamma) + \gamma = \beta + \delta > d_*(U,W)$, by Fact 2. Similarly, $f_4(\tau_{13}) = \alpha - \tau_{13} + \delta = \alpha - (\delta-\gamma) + \delta = \alpha + \gamma > d_*(U,W)$, by Fact 1. Hence, neither $f_2(\tau_{13})$ nor $f_4(\tau_{13})$ can be the minimal value.

Given Claims 1 and 2, the paths, $UVAW$ and $UVACW$, suffice to determine the value $d_*(U,W)$, which requires computing only their *single* point of intersection: $\tau_{13} = \delta - \gamma$. At that point, $f_1(\tau_{13}) = f_1(\delta-\gamma)$ $= \sigma + \max\{-v, -(\delta-\gamma)\} + \delta = \sigma + \max\{\delta - v, \gamma\} = \mathrm{d}(U,V) + \max\{\mathrm{d}(A,W) - v, \mathrm{d}(C,W)\}$. Hence:

$$d_*(U,W) = \mathrm{d}(U,V) + \max\{\mathrm{d}(A,W) - v, \mathrm{d}(C,W)\}$$

In the sample plot from Figure 12d, $\delta - \gamma = 13 - 8 = 5$ and $d_*(U,W) = f_1(5) = -5 + \max\{13-6, 8\} = -5+8 = 3$.

Claim 3. $x < \delta - \gamma \le y$.

Proof of Claim 3. If $\delta - \gamma < x$, then let $\mathcal{P}_{cw}$ be the *ordinary* path obtained by concatenating $(C, -x, A)$ and AW, where $(C, -x, A)$ is the stand-in edge for the UC edge $(C, C{:}{-y}, A)$. Then $|\mathcal{P}_{cw}| = -x + \delta < \gamma$, since $\delta - \gamma < x$. But this contradicts that γ is the shortest path-length from C to W. Similarly, if $\delta - \gamma > y$, then let $\mathcal{P}_{aw}$ be the *ordinary* path obtained by concatenating (A, y, C) and CW, where (A, y, C) is the stand-in edge for the LC edge $(A, c{:}x, C)$. Then $|\mathcal{P}_{aw}| = y + \gamma < \delta$, since $\delta - \gamma > y$. But this contradicts that δ is the shortest path-length from A to W. Finally, if $\delta - \gamma = x$, then let $\mathcal{P}_{uw}$ be the *ordinary* path obtained by concatenating UV, $(V, -x, A)$ and AW, where $(V, -x, A)$ is the stand-in edge for $(V, C{:}{-y}, A)$. Then $|\mathcal{P}_{uw}| = \sigma - x + \delta = \sigma + \gamma$, since $\delta - \gamma = x$. Meanwhile:

[7] τ_{14}^ω is undefined since $f_1^\omega(\omega_c) = f_4(\omega_c)$ holds only if $\sigma = \alpha$, which is impossible, by Fact 4. Similarly, τ_{23}^v is undefined since $f_2(\omega_c) = f_3^v(\omega_c)$ holds only if $\beta = \sigma - v$, which is impossible, by Fact 3.

$$\begin{aligned} d_*(U,W) &= \sigma + \max\{\delta - v, \gamma\} \\ &= \sigma + \max\{\gamma + x - v, \gamma\} \quad \text{(Since } \delta - \gamma = x\text{)} \\ &= \sigma + \gamma \quad \text{(Since } x < v \text{ for all regular waits)} \end{aligned}$$

Hence, $|\mathcal{P}_{uw}| = d_*(U,W)$, contradicting the need for labeled edges associated with (A, x, y, C).

Claim 4. UV necessarily comprises only negative edges, while CW necessarily comprises only non-negative edges.

Proof of Claim 4. First, consider the values of $f_2(\delta - \gamma)$ and $f_4(\delta - \gamma)$:

$$f_2(\delta - \gamma) = \beta + (\delta - \gamma) + \gamma = \beta + \delta > d_*(U,W).$$
$$f_4(\delta - \gamma) = \alpha - (\delta - \gamma) + \delta = \alpha + \gamma > d_*(U,W).$$

Since both of these are greater than $d_*(U,W)$, while $f_1(\delta - \gamma) = f_3(\delta - \gamma) = d_*(U,W)$, the fact that all of the involved functions are continuous implies that $\min\{f_1(\omega_c), f_3(\omega_c)\} < \min\{f_2(\omega_c), f_4(\omega_c)\}$ *for all* ω_c *in some open interval around* $\delta - \gamma$. In addition, since Claim 3 gives that $x < \delta - \gamma$, we may conclude that for some $\epsilon > 0$, the above inequality holds for all $\omega_c \in (\delta - \gamma - \epsilon, \delta - \gamma) \subseteq (x, \delta - \gamma)$. Furthermore, for all such ω_c:

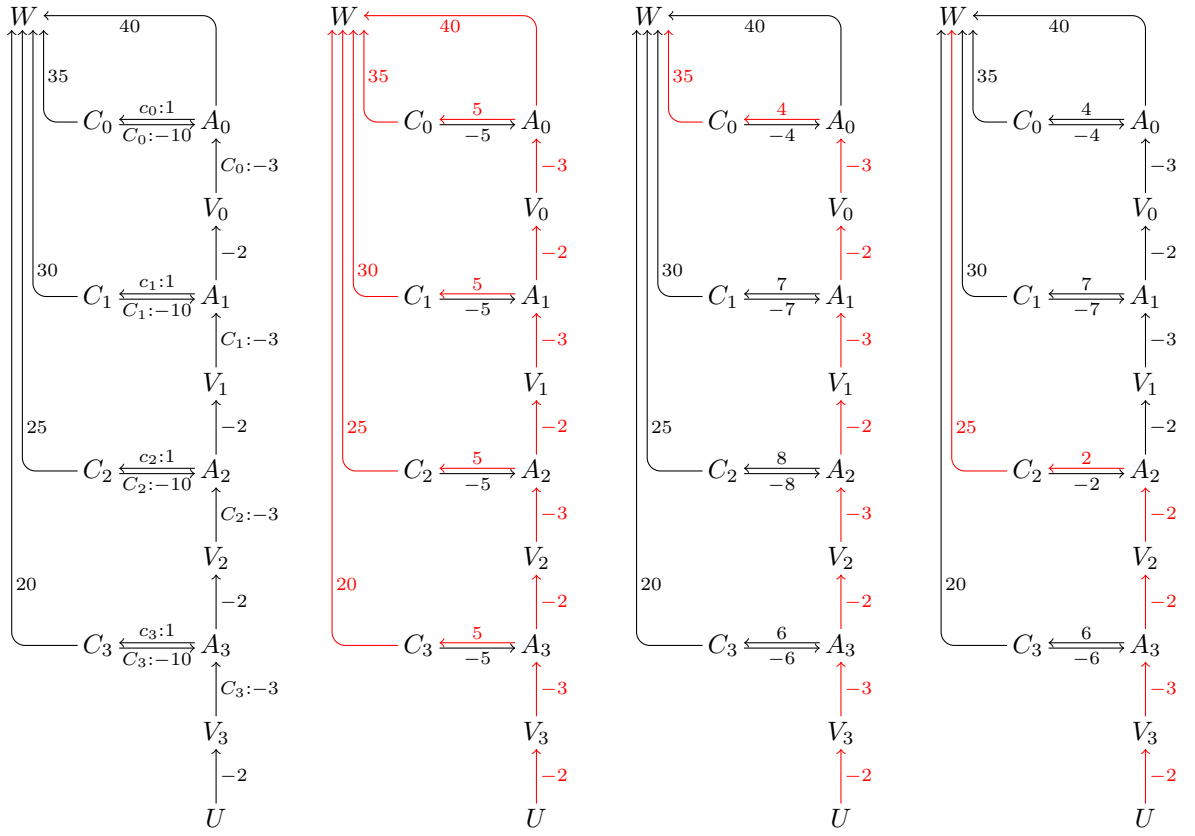
$$\begin{aligned} f_3(\omega_c) &= \sigma + \max\{-v, -\omega_c\} + \omega_c + \gamma \\ &< \sigma + \max\{-v, -\omega_c\} + \delta \text{ (Since } \omega_c < \delta - \gamma\text{)} \\ &= f_1(\omega_c) \\ &< \min\{f_2(\omega_c), f_4(\omega_c)\} \end{aligned}$$

That implies that for all such ω_c, $UVACW$ must be an SVP, in which case the subpaths UV and CW must also be SVPs. Since UV precedes a negative edge (namely, the wait edge), and CW follows a non-negative edge (namely, the LC edge), the result follows.

Note 1. Up to this point, we have assumed that Λ contained only one wait edge $E_v = (V, C{:}{-}v, A)$. Now consider the possibility of Λ containing other wait edges. For any other wait edge $E_{v'} = (V', C{:}{-}v', A)$, the value of $\delta - \gamma$ is fixed, but the value of $f(\delta - \gamma) = \sigma + \max\{\delta - v', \gamma\}$ depends on both $\sigma = d(U, V')$ and v'. Therefore, from all of the wait edges in Λ, let E_v be one that *minimizes* the value of $f(\delta - \gamma)$. Then, by the preceding analysis, for each value of ω_c, the corresponding projection must include an SVP $\mathcal{P}_v$ from U to W for which $|\mathcal{P}_v|_{\omega_c} \leq f(\delta - \gamma)$. Therefore, $f(\delta - \gamma) \leq d_*(U,W)$. But, by the selection of E_v, no *shorter* vee-path from U to W exists in the projection where $\omega_c = \delta - \gamma$. Therefore, $f(\delta - \gamma) = d_*(U,W)$; and since Λ is minimally sufficient for $d_*(U,W)$, it follows that Λ cannot contain more than one such wait edge.

Note 2. Finally, the proof so far has assumed that $W \notin \{A, C\}$. $W \not\equiv A$ is justified since the projection of the LC edge $(A, c{:}x, C)$ in any situation cannot be in any simple SVP ending in A and hence cannot be needed by $d_*(U, A)$. In addition, since the UC edge and any wait edge labeled by C necessarily point at A, any SVP from U to A whose last edge derived from a UC or wait edge would, in the situation where $\omega_c = x$, have length $-x$ (i.e., the length of the corresponding stand-in edge), implying that the UC or wait edge was not needed. $W \not\equiv C$ is justified since the only way that a wait edge $(V, C{:}{-}v, A)$ can be needed for $d_*(U, C)$ is if, in each projection, there is an SVP that includes the concatenation of $(V, C{:}{-}v, A)$ and the LC edge $(A, c{:}x, C)$. The length of that two-edge subpath is maximized when $\omega_c = y$, where: $|VAC|_y = \max\{-v, -y\} + y = \max\{y - v, 0\} = y - v$. ($y - v > 0$, given the assumption that all misleading waits have been fixed.) But then, in that projection, VAC is dominated by the stand-in edge, $(V, y - v, C)$, generated by the VAC rule, which is presumed to already be in the network, which implies that $(V, C{:}{-}v, A)$ cannot be needed for $d_*(U, C)$.

Similar arguments show that $U \notin \{A, C\}$ (i.e., the contingent link (A, x, y, C) cannot be needed for $d_*(A, W)$ or $d_*(C, W)$). In particular, if $U \equiv A$, then the only labeled edge whose projection could participate in any simple SVP emanating from A would be the LC edge $(A, c{:}x, C)$. But in the situation where $\omega_c = y$, the projection of that LC edge is its stand-in edge (A, y, C), which would imply that there was an ordinary path from A to W with length at most $d_*(A, W)$, contradicting the need for edges associated with (A, x, y, C). Similarly, if $U \equiv C$, then the only labeled edge whose projection could participate in any simple SVP emanating from C would be the UC edge $(C, C{:}{-}y, A)$. But in the situation where $\omega_c = x$, the projection of that UC edge is its stand-in edge $(C, -x, A)$, which would imply that there was an ordinary path from C to W with length at most $d_*(C, W)$, contradicting the need for edges associated with (A, x, y, C). □



(a) ESTNU paths

(b) Projection $\omega = (5, 5, 5, 5)$, $d_\omega(U, W) = 20 = d_*(U, W)$

(c) Projection $\omega = (4, 7, 8, 6)$, $d_\omega(U, W) = 19 \leq d_*(U, W)$

(d) Projection $\omega = (4, 7, 2, 6)$, $d_\omega(U, W) = 18 \leq d_*(U, W)$

Figure 13: Illustrating the canonical form of nested diamond structures

2.1 The Structure of Nested Diamonds

This section introduces the *canonical form of nested diamonds*. It shows that if a value $d_*(U, W)$ depends on nested diamonds, then there is necessarily a chain of nested diamonds with certain characteristics. Figure 13a shows a sample ESTNU graph that illustrates the canonical form of nested diamonds in the case of four contingent links. For ease of discussion, the contingent links all have the same bounds: $[1, 10]$; the wait edges all have the same value: -3; and the ordinary edges preceding the wait edges all have the same value: -2. In general, for each C_i, there is a shortest path from C_i to W that comprises one or more non-negative ordinary edges, while for each V_i, there is a shortest path from A_{i+1} to V_i that comprises zero or more negative ordinary edges. (Let $A_4 = U$.) Figure 13b shows the *maximizing* situation/projection where each contingent link has the same duration: 5. In this situation, all of the red paths from U to W have the same length: $d_*(U, W) = 20$. In particular, letting

$$\begin{aligned}
\Lambda_0 &= \{(A_0, 1, 10, C_0), (V_0, C_0{:}{-}3, A_0)\} \\
\Lambda_1 &= \Lambda_0 \cup \{(A_1, 1, 10, C_1), (V_1, C_1{:}{-}3, A_1)\} \\
\Lambda_2 &= \Lambda_1 \cup \{(A_2, 1, 10, C_2), (V_2, C_2{:}{-}3, A_2)\} \\
\Lambda_3 &= \Lambda_2 \cup \{(A_3, 1, 10, C_3), (V_3, C_3{:}{-}3, A_3)\}
\end{aligned}$$

observe that

$C_0 - A_0 = 5$ is maximizing for $d_*^{\Lambda_0}(A_1, W)$;
$C_1 - A_1 = C_0 - A_0 = 5$ is maximizing for $d_*^{\Lambda_1}(A_2, W)$;
$C_2 - A_2 = C_1 - A_1 = C_0 - A_0 = 5$ is maximizing for $d_*^{\Lambda_2}(A_3, W)$; and
$C_3 - A_3 = C_2 - A_2 = C_1 - A_1 = C_0 - A_0 = 5$ is maximizing for $d_*^{\Lambda_3}(U, W)$.

Similarly to the situation in Lemma 4 where the paths $UVAW$ and $UVACW$ have the same length in the maximizing situation where $C - A = \delta - \gamma$, for each $i \in \{0, 1, 2, 3\}$ in Figure 13b, $C_i - A_i = \delta_i - \gamma_i = 5$, where

$\delta_i = d_*^{\Lambda_{i-1}}(A_i, W)$ and $\gamma_i = \mathrm{d}(C_i, W)$, ensures that the paths $A_{i+1}\, V_i A_i\, W$ and $A_{i+1}\, V_i A_i C_i\, W$ have the same length. (Let $\Lambda_{-1} = \emptyset$.)

For any situation ω, it will be shown that $d_\omega^{\Lambda_3}(U, W) \leq d_*^{\Lambda_3}(U, W)$. For example, in the situation/projection where $\omega = (4, 7, 8, 6)$, Figure 13c shows that the SVP from U to W (shown in red) follows the chain of activation timepoints until it hits A_0, where the SVP follows the edges $(A_0, 4, C_0)$ and $(C_0, 35, W)$. In that situation, $d_\omega^{\Lambda_3}(U, W) = 19 \leq 20 = d_*^{\Lambda_3}(U, W)$. Similarly, in the situation/projection where $\omega = (4, 7, 2, 6)$, Figure 13d shows that the SVP from U to W (shown in red) follows the chain of activation timepoints until it hits A_2, where it then follows the edges $(A_2, 2, C_2)$ and $(C_2, 25, W)$, resulting in $d_\omega^{\Lambda_3}(U, W) = 18 \leq 20 = d_*^{\Lambda_3}(U, W)$. This highlights the similarity with Lemma 4 where, depending on the situation, sometimes the SVP follows $UVAW$, while other times it follows $UVACW$.

The theorem below addresses several challenges. First, it can be used to recursively specify the chain of contingent links from U to W that form the backbone of the structure of nested diamonds. Second, it confirms that each $d_*^{\Lambda_{i-1}}(C_i, W)$ value is realized by an ordinary path. Third, it can be used to recursively ensure (from W back to U) that any maximizing situation for each $d_*^{\Lambda_{i-1}}(A_i, W)$ can be extended to a maximizing situation for the preceding $d_*^{\Lambda_i}(A_{i+1}, W)$. Fourth, it confirms that each pair of consecutive sets of labeled edges, Λ_{i+1} and Λ_i, differs only in that Λ_i represents exactly one fewer contingent link. The end result is that each $d_*(U, W)$ value is necessarily realized by a structure of nested diamonds in canonical form.

Because it cannot be assumed in advance that the number of contingent links represented by successive Λ_i sets differ by exactly one, the statement of the theorem uses the following notation: for each $i \in \{j, j-1, \ldots\}$, $\Lambda_i \subseteq \Lambda_i^+$ are sets of labeled edges such that Λ_i is minimally sufficient for $d_*^{\Lambda_i^+}(A_{i+1}, W)$ (where $A_{j+1} = U$); and $\Lambda_i^- \subseteq \Lambda_i$ is the same as Λ_i except that it excludes the labeled edges for one contingent link. Then Λ_i^- becomes Λ_{i-1}^+ for the next iteration. Eventually, it is proven that $\Lambda_i^+ = \Lambda_i$ for each $i < j$.

Theorem 2. *Let $\mathcal{G}$ be any dispatchable ESTNU, and W any timepoint in $\mathcal{G}$. For each integer $j > 0$, let $P(j)$ be the following proposition:*
Let U be any timepoint in $\mathcal{G}$; and let Λ_j and Λ_j^+ be any sets of labeled edges from $\mathcal{G}$ such that:

(1) $\Lambda_j \subseteq \Lambda_j^+$;
(2) Λ_j represents j contingent links; and
(3) Λ_j is minimally sufficient for $d_*^{\Lambda_j^+}(U, W)$.

Then, there exists a contingent link $\mathrm{CL}_j = (A_j, x_j, y_j, C_j)$ and an associated wait edge $(V_j, C_j{:}{-}v_j, A_j)$ such that A_j is not *constrained to occur* before *the activation timepoint of any other contingent link represented in Λ_j, and:*

$$d_*^{\Lambda_j}(U, W) \;=\; d_*^{\Lambda_j^+}(U, W) \;=\; \mathrm{d}(U, V_j) + \max\{\delta_j - v_j, \gamma_j\},$$

where:

- *$\delta_j = d_*^{\Lambda_j^-}(A_j, W)$ and $\gamma_j = d_*^{\Lambda_j^-}(C_j, W) = \mathrm{d}(C_j, W)$ (i.e., an ordinary ESTNU path from C_j to W provides an SVP in each situation ω_j^- for $\mathcal{G}^{\Lambda_j^-}$), where:*
- *$\Lambda_j^- = \Lambda_j{}^{-\mathrm{CL}_j}$ is the same as Λ_j, except that it does not include labeled edges associated with CL_j; and*
- *as illustrated in Figure 14a, there exists a path UV_j from U to V_j comprising only negative ordinary edges for which $|UV_j| = \mathrm{d}(U, V_j)$; and a path $C_j\, W$ from C_j to W comprising only non-negative ordinary edges for which $|C_j\, W| = \mathrm{d}(C_j, W)$.*

Furthermore, for any situation ω_j^- for $\mathcal{G}^{\Lambda_j^-}$ that is maximal for $d_*^{\Lambda_j^-}(A_j, W)$, the situation ω_j for Λ_j that extends ω_j^- by specifying the duration $C_j - A_j = \delta_j - \gamma_j$ is necessarily maximal for $d_*^{\Lambda_j}(U, W) = d_*^{\Lambda_j^+}(U, W)$; and, as also illustrated in Figure 14a:

- *the path $UV_j A_j C_j\, W$ (blue in the figure), obtained by concatenating the path UV_j, the wait edge $(V_j, C_j{:}{-}v_j, A_j)$, the LC edge $(A_j, c_j{:}{-}x_j, C_j)$, and the path $C_j\, W$, satisfies $|UV_j A_j C_j\, W|_{\omega_j} = d_*^{\Lambda_j}(U, W) = d_*^{\Lambda_j^+}(U, W)$; and*
- *if $A_j\, W$ is any path in $\mathcal{G}^{\Lambda_j^-}$ from A_j to W such that $|A_j\, W|_{\omega_j^-} = d_*^{\Lambda_j^-}(A_j, W)$, then the path $UV_j A_j\, W$ (red in the figure), obtained by concatenating UV_j, the wait edge $(V_j, C_j{:}{-}v_j, A_j)$, and $A_j\, W$, satisfy $|UV_j A_j\, W|_{\omega_j} = d_*^{\Lambda_j}(U, W)$.*

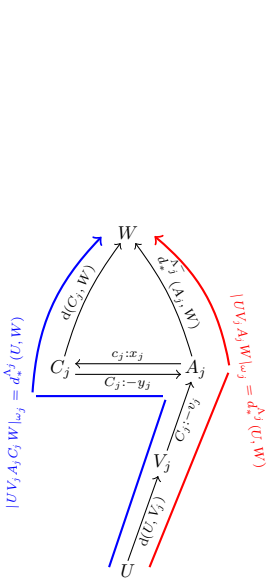
Then $P(j)$ holds for all $j > 0$.

Proof. By strong induction on $j = ||\Lambda_j||$, the number of contingent links represented in Λ_j. Suppose that $P(i)$ holds for all $i < j$. We must prove that $P(j)$ holds. Let U and $\Lambda_j \subseteq \Lambda_j^+$ be as in the statement of the theorem.

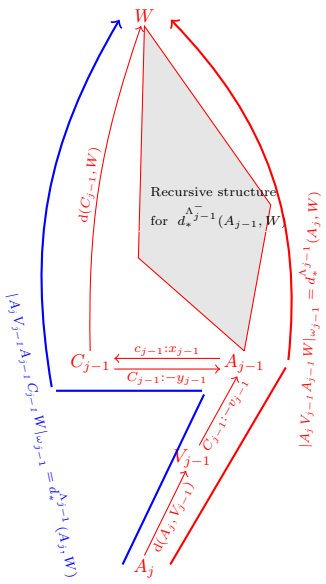
Thus, Λ_j is minimally sufficient for $d_*^{\Lambda_j^+}(U,W)$. Let $\mathrm{CL}_j = (A_j, x_j, y_j, C_j)$ be any contingent link represented in Λ_j such that A_j is *not* constrained to occur *before* the activation timepoint for any *other* contingent link represented in Λ_j. Such a contingent link must exist since otherwise $\mathcal{G}$ would contain a negative cycle of precedence constraints and hence could not be dynamically controllable.

Let Λ_{C_j} be the subset of Λ_j that *only* includes edges associated with the contingent link CL_j. Let $e_j = (A_j, c_j{:}x_j, C_j)$ be the LC edge associated with CL_j, and $E_j = (C_j, C_j{:}{-}y_j, A_j)$ the UC edge. The following cases mirror the corresponding cases from Lemma 4.

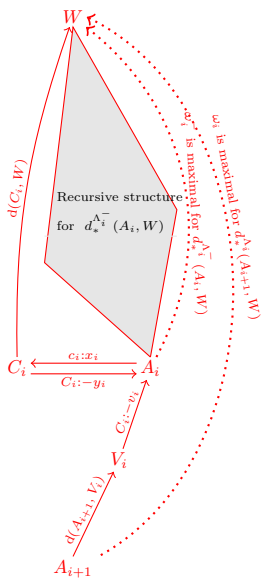
Case 1: $\Lambda_{C_j} = \{e_j\}$. Let ω_j^- be any situation for $\mathcal{G}^{\Lambda_j^-}$ and $\hat{\omega}_j$ be the situation for $\mathcal{G}^{\Lambda_j}$ that is the same as ω_j^- except that it also specifies the duration $C_j - A_j = y_j$ (i.e., its maximum duration). Then there must be an ESTNU path $\mathcal{P}$ from U to W in $\mathcal{G}^{\Lambda_j}$ that includes e_j and whose projection in the situation $\hat{\omega}_j$ has length at most $d_*^{\Lambda_j}(U,W)$. But then modifying $\mathcal{P}$ by replacing e_j by its ordinary stand-in edge (A, y, C) yields a path $\mathcal{P}'$ with edges drawn solely from $\mathcal{G}^{\Lambda_j^-}$ and whose projection in any situation ω_j that is the same as ω_j^- except possibly for the value of the duration $C_j - A_j$ is at most $d_*^{\Lambda_j}(U,W)$. Since the choice of ω_j^- was arbitrary, it follows that



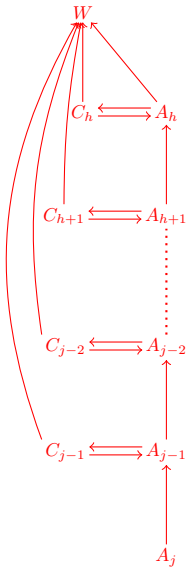
(a) Paths from the statement of Theorem 2



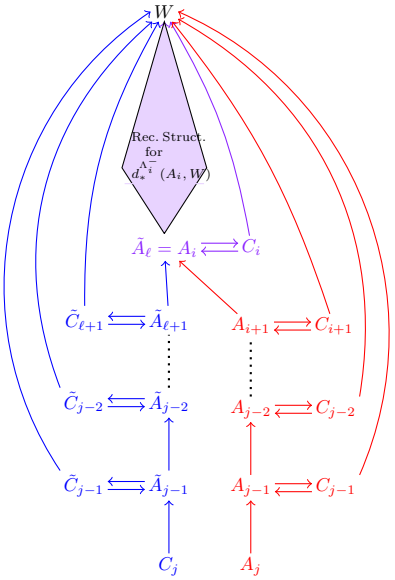
(b) Initial application of the inductive hypothesis to A_j and Λ_j^-



(c) One link in the canonical structure



(d) $\mathcal{S}_{a_j w}$: the canonical structure from A_j to W



(e) Canonical structures from C_j (blue) and from A_j (red) intersecting at $A_i = \tilde{A}_\ell$ (purple)

Figure 14: Canonical structures emanating from C_j (blue) and A_j (red) analyzed in the proof of Theorem 2

e_j is not needed for $d_*^{\Lambda_j}(U, W)$, which is a contradiction.

Case 2: $e_j \notin \Lambda_{C_j}$. Let ω_j^- be any situation for $\mathcal{G}^{\Lambda_j^-}$ and $\hat{\omega}_j$ be the situation for $\mathcal{G}^{\Lambda_j}$ that is the same as ω_j^- except that it also specifies the duration $C_j - A_j = x_j$ (i.e., its minimum duration). Then there must be a *simple* ESTNU path $\mathcal{P}$ from U to W in $\mathcal{G}^{\Lambda_j}$ that includes either E_j or some wait edge $F_j = (V_j, C_j{:}{-}v_j, A_j)$ and whose projection has length at most $d_*^{\Lambda_j}(U, W)$. But modifying $\mathcal{P}$ by replacing E_j or F_j by its ordinary stand-in edge, $(C_j, -x_j, A_j)$ or $(V_j, -x_j, A_j)$, respectively, yields a path $\mathcal{P}'$ that only uses edges from $\mathcal{G}^{\Lambda_j^-}$ and whose projection in any situation ω_j that is the same as ω_j^- except possibly for the value of the duration $C_j - A_j$ is at most $d_*^{\Lambda_j}(U, W)$. Since the choice of ω_j^- was arbitrary, it follows that the edges labeled by C_j are not needed for $d_*^{\Lambda_j}(U, W)$, which is a contradiction.

Case 3: $\Lambda_{C_j} = \{e_j, E_j\}$ *(i.e., no wait edges are needed).* Given that Λ_j is *minimally sufficient* for $d_*^{\Lambda_j^+}(U, W)$, and Λ_j^- does not include any labeled edges associated with CL_j, it follows that there must be some situation ω_j^- for $\mathcal{G}^{\Lambda_j^-}$ for which every SVP from U to W has length greater than $d_*^{\Lambda_j^+}(U, W)$. Next, let $\alpha_j^- = d_{\omega_j^-}^{\Lambda_j^-}(U, C)$, $\beta_j^- = d_{\omega_j^-}^{\Lambda_j^-}(U, A)$, $\gamma_j^- = d_{\omega_j^-}^{\Lambda_j^-}(C, W)$, and $\delta_j^- = d_{\omega_j^-}^{\Lambda_j^-}(A, W)$. Then consider the family Ω_j of situations over $\mathcal{G}^{\Lambda_j}$ where each $\omega_j \in \Omega_j$ is the same as ω_j^- except that it also specifies some value for $C_j - A_j$. By construction, as the value of $C_j - A_j$ varies across the situations $\omega_j \in \Omega_j$, the values of $\alpha_j^-, \beta_j^-, \gamma_j^-$ and δ_j^- are fixed. In addition, $\alpha_j^- + \gamma_j^- > d_*^{\Lambda_j^+}(U, W)$ and $\beta_j^- + \delta_j^- > d_*^{\Lambda_j^+}(U, W)$; and for each $\omega_j \in \Omega_j$, there must be an SVP from U to W that uses either the LC edge or the UC edge associated with CL_j. Hence, we are in the same circumstance as that in Case 3 of Lemma 4, which leads to a contradiction.

As a result of Cases 1–3 above, it follows that Λ_{C_j} must include the LC edge e_j and at least one wait edge labeled by C_j.

Considering $d_*^{\Lambda_j^-}(A_j, W)$. Let $\Lambda_j^- = \Lambda_{\mathrm{j}}{}^{-\mathrm{CL}_j}$ be the same as Λ_j, except that it does not include any labeled edges associated with the contingent link CL_j; and let $\Lambda_{j-1} \subseteq \Lambda_j^-$ be such that Λ_{j-1} is minimally sufficient for $d_*^{\Lambda_j^-}(A_j, W)$. Since the number of contingent links represented in Λ_{j-1} is at most $j-1$, applying the inductive hypothesis to the timepoint A_j (instead of U) and $\Lambda_{j-1} \subseteq \Lambda_j^-$ (instead of $\Lambda_j \subseteq \Lambda_j^+$)[8] ensures that, as illustrated in Figure 14b, there exists a contingent link $\mathrm{CL}_{j-1} = (A_{j-1}, x_{j-1}, y_{j-1}, C_{j-1})$ and an associated wait edge $(V_{j-1}, C_{j-1}{:}{-}v_{j-1}, A_{j-1})$ such that A_{j-1} is not constrained to occur before the activation timepoint of any other contingent link represented in Λ_{j-1}, and:

$$d_*^{\Lambda_{j-1}}(A_j, W) \;=\; d_*^{\Lambda_j^-}(A_j, W) \;=\; \mathrm{d}(A_j, V_{j-1}) + \max\{\delta_{j-1} - v_{j-1}, \gamma_{j-1}\}, \text{ where:}$$

- $\delta_{j-1} = d_*^{\Lambda_{j-1}^-}(A_{j-1}, W)$ and $\gamma_{j-1} = d_*^{\Lambda_{j-1}^-}(C_{j-1}, W) = \mathrm{d}(C_{j-1}, W)$ (i.e., an ordinary ESTNU path from C_{j-1} to W provides an SVP in each situation ω_{j-1}^- for $\mathcal{G}^{\Lambda_{j-1}^-}$), where:
- $\Lambda_{j-1}^- = \Lambda_{\mathrm{j-1}}{}^{-\mathrm{CL}_{j-1}}$ is the same as Λ_{j-1}, except that it does not include any labeled edges associated with the contingent link CL_{j-1}; and
- there exists a path $A_j\, V_{j-1}$ from A_j to V_{j-1} comprising only negative ordinary edges for which $|A_j\, V_{j-1}| = \mathrm{d}(A_j, V_{j-1})$, and a path $C_{j-1}\, W$ from C_{j-1} to W comprising only non-negative ordinary edges for which $|C_{j-1}\, W| = \mathrm{d}(C_{j-1}, W)$;
- any situation ω_{j-1}^- for $\mathcal{G}^{\Lambda_{j-1}^-}$ that is maximal for $d_*^{\Lambda_{j-1}^-}(A_{j-1}, W)$ can be extended to a situation ω_{j-1} for $\mathcal{G}^{\Lambda_{j-1}}$ that is maximal for $d_*^{\Lambda_{j-1}}(A_j, W) = d_*^{\Lambda_j^-}(A_j, W)$, by setting $C_{j-1} - A_{j-1} = \delta_{j-1} - \gamma_{j-1} = d_*^{\Lambda_{j-1}^-}(A_{j-1}, W) - \mathrm{d}(C_{j-1}, W)$; and, as also illustrated in Figure 14b:
  - the path $A_j\, V_{j-1}\, A_{j-1}\, C_{j-1}\, W$ (blue in the figure), obtained by concatenating the path $A_j V_{j-1}$, the wait edge $(V_{j-1}, C_{j-1}{:}{-}v_{j-1}, A_{j-1})$, the LC edge $(A_{j-1}, c_{j-1}{:}{-}x_{j-1}, C_{j-1})$, and the path $C_{j-1}W$, satisfies $|A_j\, V_{j-1}\, A_{j-1}\, C_{j-1}\, W|_{\omega_{j-1}} = d_*^{\Lambda_{j-1}}(A_j, W) = d_*^{\Lambda_j^-}(A_j, W)$; and
  - if $A_{j-1}\, W$ is any path in $\mathcal{G}^{\Lambda_{j-1}^-}$ from A_{j-1} to W such that $|A_{j-1}\, W|_{\omega_{j-1}^-} = d_*^{\Lambda_{j-1}^-}(A_{j-1}, W)$, then the path $A_j\, V_{j-1}\, A_{j-1}\, W$ (red In the figure), obtained by concatenating $A_j\, V_{j-1}$, the wait edge $(V_{j-1}, C_{j-1}{:}{-}v_{j-1}, A_{j-1})$, and $A_{j-1}\, W$, satisfy $|A_j\, V_{j-1}\, A_{j-1}\, W|_{\omega_{j-1}} = d_*^{\Lambda_{j-1}}(A_j, W) = d_*^{\Lambda_j^-}(A_j, W)$.

Next, proceeding recursively, as illustrated in Figures 14c and 14d, applying the inductive hypothesis first to A_{j-1} and $\Lambda_{j-2} \subseteq \Lambda_{j-1}^-$, where Λ_{j-2} is minimally sufficient for $d_*^{\Lambda_{j-1}^-}(A_{j-1}, W)$; then to A_{j-2} and $\Lambda_{j-3} \subseteq \Lambda_{j-2}^-$,

[8] For consistency of notation, it can be helpful to think of U as A_{j+1}, even though U may not be an activation timepoint for any contingent link, and Λ_j^+ as Λ_{j+1}^-, even though there is no $j+1^{\mathrm{st}}$ iteration.

where Λ_{j-3} is minimally sufficient for $d_*^{\Lambda_{j-2}^-}(A_{j-2}, W)$; and so on, must eventually terminate, since each iteration involves a minimally sufficient set of labeled edges representing fewer contingent links. The result is a complete path structure we call the *canonical form of nested diamonds.* (Recall the canonical structures from Figure 13.)

As illustrated in Figure 14d, let $i = h$ be the last iteration having a *non-empty* set of labeled edges, Λ_h, that is minimally sufficient for $d_*^{\Lambda_{h+1}^-}(A_{h+1}, W)$. Let (A_h, x_h, y_h, C_h) be the corresponding contingent link, and let $\delta_h = d_*^{\Lambda_h^-}(A_h, W)$ and $\gamma_h = d_*^{\Lambda_h^-}(C_h, W) = d(C_h, W)$, where Λ_h^- is the same as Λ_h, except that it does not include labeled edges from (A_h, x_h, y_h, C_h). Now, since $i = h$ is the last iteration having a non-empty set of minimally sufficient labeled edges, it follows that $\Lambda_{h-1} = \emptyset$ must be minimally sufficient for $\delta_h = d_*^{\Lambda_h^-}(A_h, W)$ and, hence, $d_*^{\Lambda_h^-}(A_h, W) = d(A_h, W)$. Therefore, the subpath from A_h to W can be presumed to contain only *ordinary* edges. Henceforth, let $\mathcal{S}_{a_j w}$ denote this *canonical structure* of paths from A_j to W, illustrated in Figure 14d. In addition, we may refer to the path from A_j to W that passes through the sequence of activation timepoints in $\mathcal{S}_{a_j w}$ as the *spine* of the canonical structure.

The sets of labeled edges resulting from this recursive application of the inductive hypothesis are summarized below where, for example, $\Lambda_{i-1} \subseteq^* \Lambda_i^-$ is used to represent not only that $\Lambda_{i-1} \subseteq \Lambda_i^-$, but also that Λ_{i-1} is minimally sufficient for $d_*^{\Lambda_i^-}(A_i, W)$:

$$\emptyset = \Lambda_{h-1} \subseteq^* \Lambda_h^- \subset \Lambda_h \subseteq^* \Lambda_{h+1}^- \subset \ldots \Lambda_{i-1} \subseteq^* \Lambda_i^- \subset \Lambda_i \subseteq^* \Lambda_{i+1}^- \ldots \Lambda_{j-2} \subseteq^* \Lambda_{j-1}^- \subset \Lambda_{j-1} \subseteq^* \Lambda_j^- \subset \Lambda_j \subseteq^* \Lambda_j^+$$

However, the following Claim implies that this chain of Λ subsets is simpler than it looks.

Claim 5. For each $i \in \{h, h+1, \ldots, j-1\}$, $\Lambda_{i-1} = \Lambda_i^-$ and, hence, $\Lambda_{i-1} \subseteq^* \Lambda_{i-1}$.

Proof of Claim 5. Suppose that $\Lambda_{j-2} \subset \Lambda_{j-1}^-$ (i.e., Λ_{j-2} is a proper subset of Λ_{j-1}^-). Then there must be some contingent link $\text{CL}^\dagger \in \Lambda_{j-1}^- \backslash \Lambda_{j-2}$, which implies that labeled edges associated with $\text{CL}^\dagger$ are not needed for $d_*^{\Lambda_{j-1}^-}(A_{j-1}, W) = \sigma_{j-2} + \max\{\delta_{j-2} - v_{j-2}, \gamma_{j-2}\}$, where $\sigma_{j-2} = d_*(A_{j-1}, V_{j-2})$ and $\gamma_{j-2} = d_*(C_{j-1}, W)$ depend only on ordinary edges. Since $\delta_{j-1} = d_*^{\Lambda_{j-1}^-}(A_{j-1}, W)$, that implies that labeled edges associated with $\text{CL}^\dagger$ are not needed for $d_*^{\Lambda_j^-}(A_j, W) = \sigma_{j-1} + \max\{\delta_{j-1} - v_{j-1}, \gamma_{j-1}\}$, where σ_{j-1} and γ_{j-1} similarly depend only on ordinary edges. But that contradicts that $\Lambda_{j-1} \subseteq^* \Lambda_j^-$. A similar analysis can be applied to each $i \in \{h, h+1, \ldots, j-1\}$.

Since for each $i \in \{h, h+1, \ldots, j-1\}$, $\Lambda_{i-1} = \Lambda_i^-$, it follows that Λ_{i-1} represents exactly one fewer contingent link than Λ_i. The preceding chain of Λ subsets can now be more compactly written as follows, where each proper subset represents exactly one fewer contingent link:

$$\emptyset = \Lambda_{h-1} \subset \Lambda_h \subset \Lambda_{h+1} \subset \ldots \subset \Lambda_{i-1} \subset \Lambda_i \subset \ldots \Lambda_{j-2} \subset \Lambda_{j-1} \subseteq^* \Lambda_j^- \subset \Lambda_j \subseteq^* \Lambda_j^+$$

Furthermore, abbreviating contingent links (A_f, x_f, y_f, C_f) by CL_f, it follows that for each $i \in \{h, h+1, \ldots, j-1\}$, the set of contingent links represented by the subset Λ_i can be notated as $\{\text{CL}_h, \text{CL}_{h+1}, \ldots, \text{CL}_i\}$. Eventually, it will be shown that, in addition, $\Lambda_{j-1} = \Lambda_j^-$, in which case the set of contingent links represented by Λ_j will be given by $\{\text{CL}_h, \text{CL}_{h+1}, \ldots, \text{CL}_j\}$, but that has not yet been shown.

Next, we shall proceed recursively through the canonical structure $\mathcal{S}_{a_j w}$ in the opposite direction, from A_h and $\Lambda_{h-1} = \Lambda_h^-$ all the way back to A_j and $\Lambda_{j-1} \subseteq^* \Lambda_j^-$, with the goal of generating a situation ω_j^- for $\mathcal{G}^{\Lambda_j^-}$ such that not only is ω_j^- maximal for $d_*^{\Lambda_j^-}(A_j, W)$, but, in addition, for each $i \in \{h, \ldots, j-1\}$, the *restriction* of ω_j^- to the contingent links in $\mathcal{G}^{\Lambda_{i+1}^-}$ is maximal for $d_*^{\Lambda_i}(A_{i+1}, W) = d_*^{\Lambda_{i+1}^-}(A_{i+1}, W)$, where for each i, ω_j^- sets the duration $C_i - A_i = \delta_i - \gamma_i = \text{d}(C_i, W) - d_*^{\Lambda_i^-}(A_i, W)$.

We begin with $\emptyset = \Lambda_{h-1}$, which is minimally sufficient for $d_*^{\Lambda_{h-1}}(A_h, W) = d_*^{\emptyset}(A_h, W) = \text{d}(A_h, W)$. Therefore, the empty situation ω_{h-1} for $\mathcal{G}^{\Lambda_{h-1}} = \mathcal{G}^{\emptyset}$ is maximal for $d_*^{\Lambda_{h-1}}(A_h, W)$. Then, by the inductive hypothesis, ω_{h-1} can be extended to a situation ω_h for $\mathcal{G}^{\Lambda_h}$ that is maximal for $d_*^{\Lambda_h}(A_{h+1}, W)$, where ω_h sets $C_h - A_h = \delta_h - \gamma_h = \text{d}(C_h, W) - d_*^{\Lambda_{h-1}}(A_h, W)$. Continuing in this way, ω_h can be extended to a situation ω_{h+1} for $\mathcal{G}^{\Lambda_{h+1}}$ that is maximal for $d_*^{\Lambda_{h+1}}(A_{h+2}, W)$, where ω_{h+1} sets $C_{h+1} - A_{h+1} = \delta_{h+1} - \gamma_{h+1}$; and so on, until we end up with a situation ω_{j-1} that is maximal for $d_*^{\Lambda_{j-1}}(A_j, W)$, where ω_{j-1} sets $C_{j-1} - A_{j-1} = \delta_{j-1} - \gamma_{j-1}$. At this point, since $\Lambda_{j-1} \subseteq^* \Lambda_j^-$, it follows that ω_{j-1} can be extended to a situation ω_j^- for $\mathcal{G}^{\Lambda_j^-}$ by setting arbitrary durations for the contingent links, if any, that are in Λ_j^-, but not in Λ_{j-1}. In addition, by construction, not only

is ω_j^- maximal for $d_*^{\Lambda_j^-}(A_j, W)$, but for each i, the restriction of ω_j^- to the contingent links in $\mathcal{G}^{\Lambda_{i+1}^-}$ is maximal for $d_*^{\Lambda_{i+1}^-}(A_{i+1}, W) = d_*^{\Lambda_i}(A_{i+1}, W)$. Furthermore, by the inductive hypothesis, since each $C_i - A_i$ equals $\delta_i - \gamma_i$, which is the value for which the path-lengths $|A_{i+1}V_iA_iC_iW|_{\omega_i}$ and $|A_{i+1}V_iA_iW|_{\omega_i}$ are equal (e.g., recall the blue and red paths from Figure 14b), it follows that in the situation ω_j^-, the length of *every* simple path through the canonical structure $\mathcal{S}_{a_jw}$ from A_j to W equals $d_*^{\Lambda_j^-}(A_j, W)$.

Considering $d_*^{\Lambda_j^-}(C_j, W)$. Eventually, we will prove that $d_*^{\Lambda_j^-}(C_j, W) = \mathrm{d}(C_j, W)$; however, at this point, we cannot make that assumption. For now, we can only generate the canonical path structure $\mathcal{S}_{c_jw}$ from C_j to W by recursively applying the inductive hypothesis in the same way that we did for the path structure $\mathcal{S}_{a_jw}$, above. Since there is no guarantee that the sequences of contingent links, labeled edge sets, and situations encountered along the way will be the same for $\mathcal{S}_{c_jw}$ as they were for $\mathcal{S}_{a_jw}$, we make no such assumption. Instead, we start afresh, recursively applying the inductive hypothesis to derive possibly different sequences, using different notation, as illustrated in Figure 14e. In particular, the sequence of activation timepoints along the spine of the structure $\mathcal{S}_{c_jw}$ will be notated as $\tilde{A}_{j-1}, \tilde{A}_{j-2}, \ldots, \tilde{A}_g$; the corresponding sequence of labeled edge sets as

$$\emptyset = \lambda_{g-1} \subset \lambda_g \subset \lambda_{g+1} \subset \ldots \lambda_{\ell-1} \subset \lambda_\ell \subset \ldots \lambda_{j-2} \subset \lambda_{j-1} \subseteq^* \Lambda_j^- \subset \Lambda_j \subseteq^* \Lambda_j^+;$$

and, for each $\ell \in \{g, \ldots, j-2\}$, the situation that maximizes $d_*^{\lambda_\ell}(\tilde{A}_{\ell+1}, W)$ as $\tilde{\omega}_\ell$; and for $\ell = j-1$, the situations, $\tilde{\omega}_{j-1}$ and $\tilde{\omega}_j^-$, that maximize $d_*^{\Lambda_{j-1}}(C_j, W) = d_*^{\Lambda_j^-}(C_j, W)$.

Claim 6. There exists a *single* situation $\hat{\omega}_j^-$ for $\mathcal{G}_j^-$ that is maximal for *both* $d_*^{\Lambda_j^-}(A_j, W)$ and $d_*^{\Lambda_j^-}(C_j, W)$.

Proof of Claim 6. First, if the sequences of activation timepoints, $\{A_i\}$ and $\{\tilde{A}_\ell\}$, appearing along the respective spines of the structures $\mathcal{S}_{a_jw}$ and $\mathcal{S}_{c_jw}$ do not have any common elements, then the sets of contingent durations, $\{C_i - A_i\}$ and $\{\tilde{C}_\ell - \tilde{A}_\ell\}$, that determine the respective values of $d_*^{\Lambda_j^-}(A_j, W)$ and $d_*^{\Lambda_j^-}(C_j, W)$ are independent. Any other contingent durations specified by ω_j^- and $\tilde{\omega}_j^-$ have no impact. Therefore, a common situation $\hat{\omega}_j^-$ can be formed by merging the contingent durations, $\{C_i - A_i\}$, specified by ω_j^- with the durations, $\{\tilde{C}_\ell - \tilde{A}_\ell\}$, specified by $\tilde{\omega}_j^-$, and choosing arbitrary values for all other contingent durations, if any.

On the other hand, suppose some A_i in $\mathcal{S}_{a_jw}$ is the same as some $\tilde{A}_\ell$ in $\mathcal{S}_{c_jw}$. In case of multiple such timepoints, choose A_i to be the first activation timepoint in the spine of $\mathcal{S}_{a_jw}$ that also appears, as some $\tilde{A}_\ell$, in the spine of $\mathcal{S}_{c_jw}$, as illustrated in Figure 14e. Then let $\Lambda_{i-1} \subseteq^* \Lambda_i^- \subset \Lambda_i$ and $\lambda_{\ell-1} \subseteq^* \lambda_\ell^- \subset \lambda_\ell$ be the corresponding sets of labeled edges, where Λ_{i-1} is minimally sufficient for $d_*^{\Lambda_i^-}(A_i, W) = d_*^{\Lambda_{i-1}}(A_i, W)$ and $\lambda_{\ell-1}$ is minimally sufficient for $d_*^{\lambda_\ell^-}(A_i, W) = d_*^{\lambda_{\ell-1}}(A_i, W)$. (By Claim 5, if $i \leq j-1$, then $\Lambda_{i-1} = \Lambda_i^-$; and if $\ell \leq j-1$, then $\lambda_{\ell-1} = \lambda_\ell$, but these facts are not relevant here.) Now, if $\Lambda_{i-1} = \lambda_{\ell-1}$, then no loss of generality follows from assuming that the substructures of $\mathcal{S}_{a_jw}$ and $\mathcal{S}_{c_jw}$ from A_i onward are the same, in which case the situation $\hat{\omega}_j^-$ can be obtained by merging the situations ω_j^- and $\tilde{\omega}_j^-$, as described above since, by the choice of $A_i = \tilde{A}_\ell$, there are no contingent links appearing in *both* $\mathcal{S}_{a_jw}$ and $\mathcal{S}_{c_jw}$ *before* $A_i = \tilde{A}_\ell$.

Otherwise, suppose that $\Lambda_{i-1} \neq \lambda_{\ell-1}$. Next, recall that the negative ordinary and wait edges comprising the spine of a canonical structure imposes a strict order on the participating activation timepoints. Therefore, it follows that no activation timepoints appearing *before* A_i in $\mathcal{S}_{a_jw}$ or *before* $\tilde{A}_\ell = A_i$ in $\mathcal{S}_{c_jw}$ can appear *after* A_i in $\mathcal{S}_{a_jw}$ or *after* $\tilde{A}_\ell = A_i$ in $\mathcal{S}_{c_jw}$. In other words, the contingent links represented in Λ_{i-1} and $\lambda_{\ell-1}$ are distinct from the contingent links preceding $A_i = \tilde{A}_\ell$ in either $\mathcal{S}_{a_jw}$ or $\mathcal{S}_{c_jw}$.

Now, it may happen that $d_*^{\Lambda_{i-1}}(A_i, W) \neq d_*^{\lambda_{\ell-1}}(A_i, W)$ (i.e., $\delta_i \neq \tilde{\delta}_\ell$), since either or both of these values may be greater than $d_*(A_i, W)$. To see this, recall that $\delta_{i+1} = d_*^{\Lambda_i}(A_{i+1}, W) = \sigma_i + \max\{\delta_i - v_i, \gamma_i\}$. In the case where $\gamma_i \geq \delta_i - v_i$, decreasing the value of δ_i will not change the value of δ_{i+1}. In other words, it is possible that $\delta_i > d_*(A_i, W)$, while still being sufficient to determine the value of δ_{i+1}. Similarly, it is possible that $\tilde{\delta}_\ell > d_*(A_i, W)$, while still being sufficient to determine the value of $\delta_{\ell+1}$.

Case 1: $\delta_i < \tilde{\delta}_\ell$. In this case, define the common situation $\hat{\omega}_j^-$ by first setting the duration of each contingent link CL_f that appears in $\mathcal{S}_{a_jw}$ to its value in ω_j^-. Next, set the durations for the contingent links that appear *before* $\tilde{A}_\ell = A_i$ in $\mathcal{S}_{c_jw}$ as follows:

- Let $\tilde{\delta}'_{\ell+1} = \tilde{\sigma}_\ell + \max\{\delta_i - \tilde{v}_\ell, \tilde{\gamma}_\ell\} \leq \tilde{\delta}_{\ell+1} = d_*^{\lambda_{\ell+1}^-}(\tilde{A}_{\ell+1}, W)$, since $\delta_i < \tilde{\delta}_\ell$.

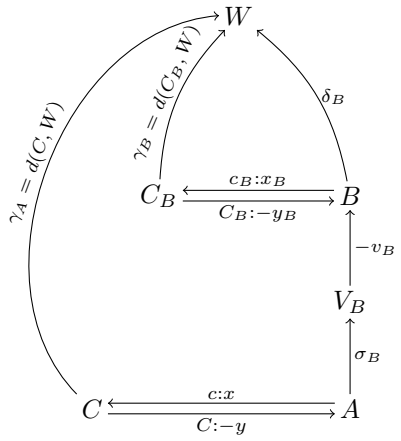


Figure 15: Computing $\delta(A, B, \mathcal{S}) = \sigma_B + \max\{\delta_B, -v_B, \gamma_B\}$ according to Claim 7

- Then set $\tilde{C}_{\ell+1} - \tilde{A}_{\ell+1} = \tilde{\delta}'_{\ell+1} - \tilde{\gamma}_{\ell+1}$.
- Let $\tilde{\delta}'_{\ell+2} = \tilde{\sigma}_{\ell+1} + \max\{\tilde{\delta}'_{\ell+1} - \tilde{v}_{\ell+1}, \tilde{\gamma}_{\ell+1}\} \leq \tilde{\delta}_{\ell+2} = d_*^{\lambda^-_{\ell+2}}(\tilde{A}_{\ell+2}, W)$, since $\tilde{\delta}'_{\ell+1} \leq \tilde{\delta}_{\ell+1}$.
- Then set $\tilde{C}_{\ell+2} - \tilde{A}_{\ell+2} = \tilde{\delta}'_{\ell+2} - \tilde{\gamma}_{\ell+2}$.

  ...
- Let $\tilde{\delta}'_{j-1} = \tilde{\sigma}_{j-2} + \max\{\tilde{\delta}'_{j-2} - \tilde{v}_{j-2}, \tilde{\gamma}_{j-2}\} \leq \tilde{\delta}_{j-1} = d_*^{\lambda^-_{j-1}}(\tilde{A}_{j-1}, W)$, since $\tilde{\delta}'_{j-2} \leq \tilde{\delta}_{j-2}$.
- Then set $\tilde{C}_{j-1} - \tilde{A}_{j-1} = \tilde{\delta}'_{j-1} - \tilde{\gamma}_{j-1}$.
- Let $\tilde{\delta}'_j = \tilde{\sigma}_{j-1} + \max\{\tilde{\delta}'_{j-1} - \tilde{v}_{j-1}, \tilde{\gamma}_{j-1}\} \leq \tilde{\delta}_j = d_*^{\lambda^-_j}(C_j, W)$, since $\tilde{\delta}'_{j-1} \leq \tilde{\delta}_{j-1}$.

Finally, set all other contingent durations, if any, to arbitrary values.
Now, the setting of durations of the contingent links that appear in $\mathcal{S}_{c_j w}$ before $\tilde{A}_\ell = A_i$ is done using the corresponding $\tilde{\delta}'_p$ values which are less than or equal to the corresponding $\tilde{\delta}_p$ values. At each point along the chain, the duration $\tilde{C}_p - \tilde{A}_p$ is set to the value $\tilde{\delta}'_p - \tilde{\gamma}_p$ which, according to the proof of Lemma 4, maximizes the length of the shortest vee-path from A_{p+1} to W across all relevant projections. Since additional labeled edges are employed in the common substructure following $\tilde{A}_\ell = A_i$, that maximum value may decrease. However, the final value $\tilde{\delta}'_j$ *cannot* be less than $d_*^{\lambda^-_j}(C_j, W)$, since it would contradict the value determined by the canonical structure $\mathcal{S}_{c_j w}$. As a result, the situation $\hat{\omega}_j^-$ is necessarily maximal for both $d_*^{\Lambda^-_j}(A_j, W)$ and $d_*^{\Lambda^-_j}(C_j, W)$.

Case 2: $\delta_i < \tilde{\delta}_\ell$. Handled similarly.

Considering $d_*^{\Lambda^-_j}(U, W)$. Although the ultimate goal is to compute $d_*^{\Lambda_j}(U, W)$, for now we consider $d_*^{\Lambda^-_j}(U, W)$: that is, the maximum length in any situation/projection of an SVP from U to W that derives from paths in $\mathcal{G}^{\Lambda^-_j}$, which *exclude* labeled edges associated with $\mathrm{CL}_j = (A_j, x_j, y_j, C_j)$. As was the case for the path structures, $\mathcal{S}_{a_j w}$ and $\mathcal{S}_{c_j w}$, seen above, recursive application of the inductive hypothesis can also be used to generate a canonical structure $\mathcal{S}_{uw}$ from U to W relevant for $d_*^{\Lambda^-_j}(U, W)$. In addition, an extension of the same kind of analysis that was done in the proof of Claim 6 ensures that there must be a *single* situation $\check{\omega}_j^-$ that is simultaneously maximal for the *three* values, $d_*^{\Lambda^-_j}(U, W), d_*^{\Lambda^-_j}(A_j, W)$ and $d_*^{\Lambda^-_j}(C_j, W)$, as follows. The analysis is complicated by the fact that the three structures may intersect in a variety of ways.

Claim 7. Given the three canonical structures, $\mathcal{S}_{a_j w}, \mathcal{S}_{c_j w}$ and $\mathcal{S}_{uw}$, there is a single situation $\check{\omega}_j^-$ for $\mathcal{G}_j^-$ that is simultaneously maximal for the *three* values, $d_*^{\Lambda^-_j}(U, W), d_*^{\Lambda^-_j}(A_j, W)$ and $d_*^{\Lambda^-_j}(C_j, W)$.

Proof of Claim 7. First, let $\mathcal{A}$ denote the set of activation timepoints appearing in the spines of *any* of the three canonical structures, $\mathcal{S}_{a_j w}, \mathcal{S}_{c_j w}$ or $\mathcal{S}_{uw}$, except that $\mathcal{A}$ should not include A_j, C_j or U. Some timepoints in $\mathcal{A}$ may participate in only one structure, others may participate in two of the three structures, and still others may participate in all three structures. Next, note that the negative ordinary

and wait edges along the spines of the three structures ensures that the timepoints in $\mathcal{A}$ are subject to a strict partial order, where $A_p \succ A_q$ if A_p occurs *after* A_q in at least one of the structures (i.e., A_p is closer to W than A_q). This strict partial order determines a directed acyclic graph since, otherwise, there would be a negative cycle in the OW-graph of ordinary and wait edges, contradicting the dispatchability of $\mathcal{G}$. Therefore, the timepoints in $\mathcal{A}$ can be put into a *topological sort* [2] that respects the strict partial order, perhaps notated as $A_{i_1} \succ A_{i_2} \succ A_{i_3} \succ \ldots A_{i_k}$.

Next, we will visit the timepoints in $\mathcal{A}$ in the order of that topological sort. For each activation timepoint $A \in \mathcal{A}$, we compute two values: (1) δ_A, an upper bound on $d_*(A, W)$; and (2) $C - A = \delta_A - d(C, W)$, where C is the contingent timepoint associated with A. These values are computed for each A as follows. First, A may have up to three direct successors in the strict partial order, at most one for each canonical structure. If A has no direct successors, then it is the final activation timepoint in one or more structures; hence, the value of δ_A is determined by ordinary edges: $\delta_A = \mathrm{d}(A, W)$. Otherwise, for each structure $\mathcal{S} \in \{\mathcal{S}_{a_j w}, \mathcal{S}_{c_j w}, \mathcal{S}_{uw}\}$, if that structure has an activation timepoint B that is a direct successor of A (i.e., $B \succ A$), then, as illustrated in Figure 15, let $\delta(A, B, \mathcal{S}) = \sigma_B + \max\{\delta_B - v_B, \gamma_B\}$, where the value of δ_B was computed when B was processed, which necessarily happens before visiting A; and where $\sigma_B = d(A, V_B)$, $-v_B$ is the length of the wait edge $(V_B, C_B{:}{-}v_B, B)$, and $\gamma_B = d(C_B, W)$ are the relevant values from the canonical structure $\mathcal{S}$. If B is a direct successor of A in more than one canonical structure, then multiple $\delta(A, B, \mathcal{S})$ values will be computed for the same B. Of course, in one structure, the direct predecessor B' of A may be different from B. Regardless of how many $\delta(A, B, \mathcal{S})$ are computed, the value of δ_A is then given by $\min\{\delta(A, B, \mathcal{S}) \mid B \text{ is a direct successor of } A \text{ in } \mathcal{S}\}$. Afterward, the duration $C - A$ is set to $\delta_A - \mathrm{d}(C, W)$. Note that this duration is chosen to maximize the value of the length of the shortest vee-path from A to W through any of the three canonical structures. As shown in Figure 12 and the proof of Lemma 4, the duration $C_B - B = \delta_B - \gamma_B = \delta_B - d(C_B, W)$ is chosen to maximize the length of the shortest vee-path across all projections.

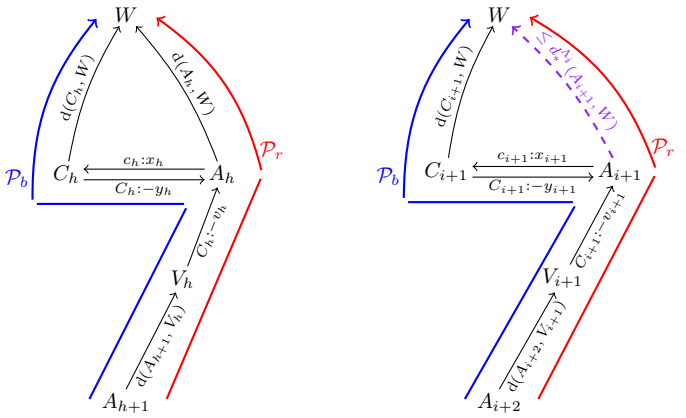
The terminal computations are done for A_j, C_j and U in the same way, except that each has exactly one direct successor. In particular, following earlier notation, the successor of A_j in $\mathcal{S}_{a_j w}$ is A_{j-1} and $\delta_{A_j} = \sigma_j + \max\{\delta_{A_{j-1}} - v_{j-1}, \gamma_{j-1}\}$, where $\delta_{A_{j-1}}$ was computed previously when A_{j-1} was encountered in the topological sort. Similarly, the successor of C_j in $\mathcal{S}_{c_j w}$ is $\tilde{A}_{j-1}$ and $\delta_{C_j} = \tilde{\sigma}_j + \max\{\delta_{\tilde{A}_{j-1}} - \tilde{v}_{j-1}, \tilde{\gamma}_{j-1}\}$, where $\delta_{\tilde{A}_{j-1}}$ was computed previously when $\tilde{A}_{j-1}$ was encountered in the topological sort. Finally, $\delta_U = \dot{\sigma}_j + \max\{\delta_{\dot{A}_{j-1}} - \dot{v}_{j-1}, \dot{\gamma}_{j-1}\}$, where $\delta_{\dot{A}_{j-1}}$ was computed previously when $\dot{A}_{j-1}$, the direct successor of U in $\mathcal{S}_{uw}$ was encountered.

To see why $\check{\omega}_j^-$ is maximal for all three of the values, $d_*^{\Lambda_j^-}(U, W), d_*^{\Lambda_j^-}(A_j, W)$ and $d_*^{\Lambda_j^-}(C_j, W)$, recall from the proof of Claim 6 that the δ values determined by any given canonical form do not necessarily provide the shortest path lengths entailed by the network. Instead, they need only be short enough to determine the relevant $d_*(\_, W)$ values (i.e., they need only be sufficient). For example, in the structure $\mathcal{S}_{a_j w}$, the value $\delta_i = d_*^{\Lambda_i^-}(A_i, W)$ may be larger than the value $d_*(A_i, W)$. This can happen when the δ_i value does not contribute to the value of $\delta_{i+1} = \sigma_i + \max\{\delta_i - v_i, \gamma_i\}$ because $\max\{\delta_i - v_i, \gamma_i\} = \gamma_i$. In this case, setting δ_i to the value $d_*(A_i, W)$ has no effect. The δ_A values computed above effectively replace existing δ values with possibly lower values to ensure that each structure provides the desired value (i.e., $d_*^{\Lambda_j^-}(U, W), d_*^{\Lambda_j^-}(A_j, W)$ or $d_*^{\Lambda_j^-}(C_j, W)$). Since each δ_A value is the minimum of existing δ values, each δ_A cannot be lower than the corresponding $d_*(A, W)$ value.

The following claim ensures that in *every* situation, there is *some* path from A_j to W in the canonical structure $\mathcal{S}_{a_j w}$ whose length is at most $d_*^{\Lambda_j^-}(A_j, W)$. This, together with the fact that in the situation ω_j^- the length of every simple path through the structure $\mathcal{S}_{a_j w}$ equals $d_*^{\Lambda_j^-}(A_j, W)$, ensures that the paths constituting $\mathcal{S}_{a_j w}$ are sufficient to determine the value of $d_*^{\Lambda_j^-}(A_j, W)$.

Claim 8. For every situation $\hat{\omega}_j^-$ for $\mathcal{G}^{\Lambda_j^-}$ there is a simple path $A_j W$ from A_j to W that employs only edges that appear within the canonical structure $\mathcal{S}_{a_j w}$ and whose projection satisfies $|A_j W|_{\hat{\omega}_j^-} \leq d_*^{\Lambda_j^-}(A_j, W)$.

Proof of Claim 8. By induction. For each $i \in \{h, h+1, \ldots, j-1\}$, let $Q(i)$ be the proposition that in any situation $\hat{\omega}_i$ for the contingent links represented in Λ_i, there is a simple path $\mathcal{P}$ from A_{h+1} to W that employs only edges that appear within $\mathcal{S}_{a_j w}$ and whose projection satisfies $|\mathcal{P}|_{\hat{\omega}_i} \leq d_*^{\Lambda_i}(A_{i+1}, W)$.



(a) Base Case of Claim 8 (b) Recursive Case of Claim 8

Figure 16: Paths analyzed in the Proof of Claim 2.2

Base Case: $i = h$. There are only two simple paths from A_{h+1} to W in $\mathcal{S}_{a_j w}$: $\mathcal{P}_b$ and $\mathcal{P}_r$, shown in blue and red, respectively, in Figure 16a. The first, $\mathcal{P}_b$, goes from A_{h+1} to V_h to A_h to C_h to W; the second, $\mathcal{P}_r$, goes from A_{h+1} to V_h to A_h to W.

Let $\tau_h = \delta_h - \gamma_h = \mathrm{d}(A_h, W) - \mathrm{d}(C_h, W)$; and let $\sigma_h = \mathrm{d}(A_{h+1}, V_h)$. Now, $\hat{\omega}_h$ only specifies a single duration, namely, $C_h - A_h$. For any $\hat{\omega}_h \leq \tau_h$:

$$\begin{array}{rcll} |\mathcal{P}_b|_{\hat{\omega}_h} & = & \sigma_h + \max\{-v_h, -\hat{\omega}_h\} + \hat{\omega}_h + \gamma_h & \\ & = & \sigma_h + \max\{\hat{\omega}_h + \gamma_h - v_h, \gamma_h\} & \\ & \leq & \sigma_h + \max\{\tau_h + \gamma_h - v_h, \gamma_h\} & \text{(Since } \hat{\omega}_h \leq \tau_h\text{)} \\ & = & \sigma_h + \max\{\delta_h - v_h, \gamma_h\} & \text{(Since } \tau_h = \delta_h - \gamma_h\text{)} \end{array}$$

Meanwhile, for any $\hat{\omega}_h \geq \tau_h$:

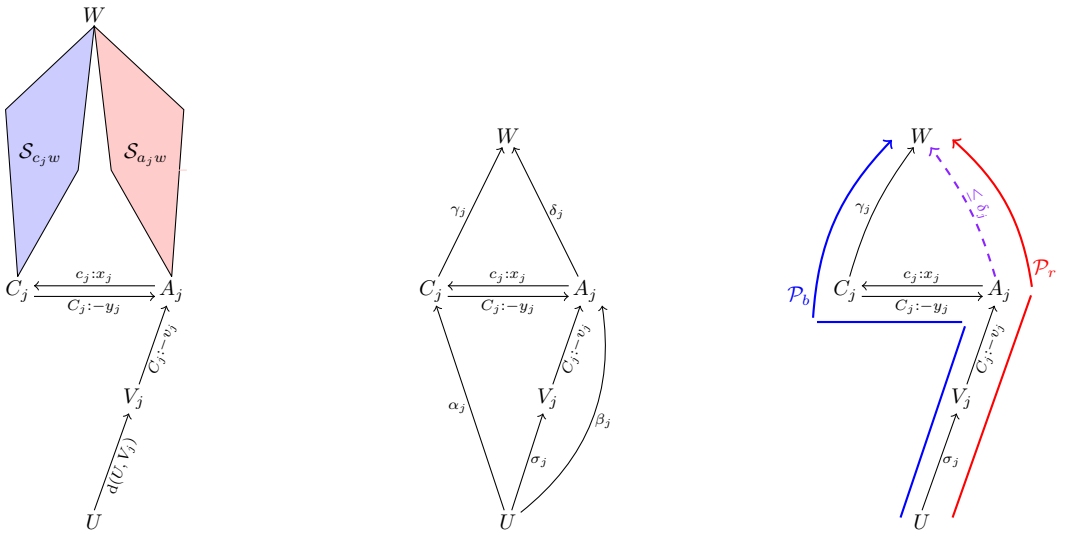
$$\begin{array}{rcll} |\mathcal{P}_r|_{\hat{\omega}_h} & = & \sigma_h + \max\{-v_h, -\hat{\omega}_h\} + \delta_h & \\ & = & \sigma_h + \max\{\delta_h - v_h, \delta_h - \hat{\omega}_h\} & \\ & \leq & \sigma_h + \max\{\delta_h - v_h, \delta_h - \tau_h\} & \text{(Since } \hat{\omega}_h \geq \tau_h\text{)} \\ & = & \sigma_h + \max\{\delta_h - v_h, \gamma_h\} & \text{(Since } \tau_h = \delta_h - \gamma_h\text{)} \end{array}$$

As seen previously, when $\hat{\omega}_h = \tau_h$, $|\mathcal{P}_r|_{\hat{\omega}_h} = |\mathcal{P}_b|_{\hat{\omega}_h} = d_*^{\Lambda_h}(A_{h+1}, W) = \sigma_h + \max\{\delta_h - v_h, \gamma_h\}$. Therefore, for any $\hat{\omega}_h$, one of the two paths has length at most $d_*^{\Lambda_h}(A_{h+1}, W)$.

Recursive Case. Assume that $Q(i)$ holds for some $i \leq j - 1$. We must show that $Q(i+1)$ holds (i.e., that in any situation $\hat{\omega}_{i+1}$ for the contingent links represented in Λ_{i+1}, there is a simple path $\mathcal{P}$ from A_{i+2} to W that employs only edges that appear within $\mathcal{S}_{a_j w}$ and for which $|\mathcal{P}|_{\hat{\omega}_{i+1}} \leq d_*^{\Lambda_{i+1}}(A_{i+2}, W)$). Let $\hat{\omega}_{i+1}$ be any situation for the contingent links represented in Λ_{i+1}. Then by $Q(i)$, the restriction of this situation to contingent links represented in Λ_i yields a path from A_{i+1} to W, shown as purple and dashed in Figure 16b, whose length is at most $d_*^{\Lambda_i}(A_{i+1}, W)$. Let $\mathcal{P}_b$ and $\mathcal{P}_r$ be the paths shown in blue and red, respectively, in Figure 16b, where $\mathcal{P}_b$ goes from A_{i+2} to V_{i+1} to A_{i+1} to C_{i+1} to W, while $\mathcal{P}_r$ goes from A_{i+2} to V_{i+1} to A_{i+1} to W; and let $\tau_{i+1} = \delta_{i+1} - \gamma_{i+1} = d_*^{\Lambda_i}(A_{i+1}, W) - \mathrm{d}(C_{i+1}, W)$. Let τ be the value of $C_{i+1} - A_{i+1}$ specified by $\hat{\omega}_{i+1}$. Similarly to the derivations in the base case, we get that if $\tau \leq \tau_{i+1}$, then $|\mathcal{P}_b|_{\hat{\omega}_{i+1}} \leq d_*^{\Lambda_{i+1}}(A_{i+2}, W)$; and if $\tau \geq \tau_{i+1}$, then $|\mathcal{P}_r|_{\hat{\omega}_{i+1}} \leq d_*^{\Lambda_{i+1}}(A_{i+2}, W)$. The only difference in the derivation is that the length of the purple dashed path from A_{i+1} to W is *at most* $d_*^{\Lambda_i}(A_{i+1}, W) = \delta_{i+1}$, but that does not affect the conclusion.

Corollary to Claim 8. Let $\hat{\omega}_j^-$ be any situation for $\mathcal{G}^{\Lambda_j^-}$; and let $C_j\, W$ be any simple path from C_j to W that employs only edges that appear within the canonical structure $\mathcal{S}_{c_j w}$. Then $|C_j\, W|_{\hat{\omega}_j^-} \leq d_*^{\Lambda_j^-}(C_j, W)$. Similarly, if UW is any simple path from U to W that employs only edges appearing within $\mathcal{S}_{uw}$, then $|UW|_{\hat{\omega}_j^-} \leq d_*^{\Lambda_j^-}(U, W)$.

Claim 9. $d_*^{\Lambda_j^-}(U, W) > d_*^{\Lambda_j}(U, W)$.



(a) Canonical structure $\mathcal{S}_{uw}^{\Lambda_j}$ (b) Paths in the projection $\hat{\omega}_j$ (c) Paths in the situation ω_j

Figure 17: Structures and paths analyzed in the Proof of Claim 10

Proof of Claim 9. Suppose that $d_*^{\Lambda_j^-}(U,W) < d_*^{\Lambda_j}(U,W)$, where the former excludes paths involving labeled edges associated with CL_j, while the latter allows them. But then by the Corollary to Claim 8, for every situation there is an ESTNU path from U to W using only edges from $\mathcal{S}_{uw}$ whose projection has length at most $d_*^{\Lambda_j^-}(U,W) < d_*^{\Lambda_j}(U,W) = d_*^{\Lambda_j^+}(U,W)$. Therefore, the maximum length of any SVP from U to W in any situation is less than $d_*^{\Lambda_j^+}(U,W)$, which is a contradiction. On the other hand, if $d_*^{\Lambda_j^-}(U,W) = d_*^{\Lambda_j}(U,W)$, then $\Lambda_j^- \subset \Lambda_j$ is sufficient for $d_*^{\Lambda_j^+}(U,W)$, contradicting that Λ_j is minimally sufficient for $d_*^{\Lambda_j^+}(U,W)$. Therefore, it must be that $d_*^{\Lambda_j^-}(U,W) > d_*^{\Lambda_j}(U,W)$.

Now we are finally ready to construct the canonical structure from U to W, to be called $\mathcal{S}_{uw}^{\Lambda_j}$, that will have all of the properties listed in the conclusion of the theorem. Starting with the contingent link $\text{CL}_j = (A_j, x_j, y_j, C_j)$, attach the structures $\mathcal{S}_{c_j w}$ and $\mathcal{S}_{a_j w}$ to the timepoints C_j and A_j, respectively, as illustrated in Figure 17a. Next, for each wait edge $(V_j, C_j{:}-v_j, A_j)$ for which there is an ordinary path $\mathcal{P}_{uv_j}$ from U to V_j, attach $\mathcal{P}_{uv_j}$ and the wait edge to the timepoint A_j, as shown in the figure. At this point, we don't know how many wait edges will be needed; however, it will soon be shown that only one is needed.

Claim 10. In any situation ω_j for $\mathcal{G}^{\Lambda_j}$ there is a simple path $\mathcal{P}$ whose edges are drawn solely from $\mathcal{S}_{uw}^{\Lambda_j}$ and whose projection satisfies $|\mathcal{P}|_{\omega_j} \leq d_*^{\Lambda_j}(U,W)$.

Proof of Claim 10. First, recall the situation ω_j^-, which is maximal for $d_*^{\Lambda_j^-}(U,W)$. Since, by Claim 2.3, $d_*^{\Lambda_j^-}(U,W) > d_*^{\Lambda_j}(U,W)$, it follows that in any situation $\hat{\omega}_j$ that extends ω_j^- by also specifying a duration for $C_j - A_j$, any ESTNU path in $\mathcal{G}^{\Lambda_j}$ whose projection in $\hat{\omega}_j$ is an SVP from U to W must include one or more labeled edges associated with the contingent link $\text{CL}_j = (A_j, x_j, y_j, C_j)$, as illustrated in Figure 17b, where: $\alpha_j = d_{\omega_j^-}^{\Lambda_j^-}(U, C_j)$, $\beta_j = d_{\omega_j^-}^{\Lambda_j^-}(U, A_j)$, $\sigma_j = d_{\omega_j^-}^{\Lambda_j^-}(U, V_j)$, $\gamma_j = d_{\omega_j^-}^{\Lambda_j^-}(C_j, W)$ and $\delta_j = d_{\omega_j^-}^{\Lambda_j^-}(A_j, W)$.

To facilitate the analysis of the paths shown in the figure, let Ω_j be the *family* of situations where each $\hat{\omega}_j \in \Omega_j$ is the same as ω_j^-, except that it also assigns a value to $C_j - A_j$. With this construction, all of the lengths in the figure represented by Greek letters are constant across the situations in Ω_j. Therefore, using the analysis in the proof of Lemma 4, there necessarily exists a single wait edge $(V_j, C_j{:}-v_j, A_j)$ such that:

(1) for each $C_j - A_j \in (x_j, y_j]$, at least one of the paths, UV_jA_jW or $UV_jA_jC_jW$, shown in Figure 17b, has length at most $d_*^{\Lambda_j}(U,W)$;

(2) the maximum length SVP from U to W over situations in Ω_j is obtained when $C_j - A_j = \delta_j - \gamma_j \in (x_j, y_j]$, where $|UV_jA_j\,W|_{\hat{\omega}_j} = |UV_jA_jC_j\,W|_{\hat{\omega}_j}$;

(3) that maximum length, call it L, is given by: $\sigma_j + \max\{\delta_j - v_j, \gamma_j\}$;

(4) there exists an SVP from U to V_j comprising solely negative edges; and

(5) there exists an SVP from C_j to W comprising solely non-negative edges.

Regarding item (3), by the definition of $d_*^{\Lambda_j}(U,W)$, it follows that $L \leq d_*^{\Lambda_j}(U,W)$. Subsequently, it will be shown that $L = d_*^{\Lambda_j}(U,W)$.

Regarding item (4), the choice of CL_j ensures that no activation timepoints from any other contingent link represented in Λ_j^- can appear in the SVP from U to V_j; hence, that path cannot contain any labeled edges and, therefore, must comprise solely ordinary negative edges. Therefore, $d_*^{\Lambda_j^-}(U,V_j) = \mathrm{d}(U,V_j)$.

Regarding item (5), recall that the final subpath along the spine of a canonical structure (e.g., the subpath from A_h to W in $\mathcal{S}_{a_jw}$, seen previously in Figure 14d) comprises only ordinary edges. Now, if the SVP from C_j to W contained any LC edges, their activation timepoints could not appear in the spine of the structure $\mathcal{S}_{c_jw}$ since the subpath of that spine from U to any activation timepoint comprises only negative edges, whereas the SVP from A_j to C_j to W comprises only non-negative edges. Therefore, any LC edges appearing in the path from C_j to W cannot appear in $\mathcal{S}_{a_jw}$. Similarly, any such LC edges cannot appear in the canonical structures, $\mathcal{S}_{c_jw}$ or $\mathcal{S}_{uw}$. As a result, the durations of any such LC edges cannot affect the paths in any of those canonical structures. Therefore, no loss of generality results from assuming that the situation ω_j^- assigns maximum durations to any such LC edges. But then any LC edges in the path from C_j to W could be replaced by their ordinary stand-in edges, implying that there must be an SVP from C_j to W comprising solely non-negative ordinary edges.

Next, we stipulate that the canonical structure $\mathcal{S}_{uw}^{\Lambda_j}$ shall contain only the one wait edge $(V_j, C_j{:}{-}v_j, A_j)$, given above and as illustrated in Figure 17a; and we shall show that this structure has all of the needed properties. In particular, we show that in *any* situation ω_j for $\mathcal{G}^{\Lambda_j}$, there exists a simple path $\mathcal{P}$ from U to W whose only edges are drawn from the structure $\mathcal{S}_{uw}^{\Lambda_j}$ and whose projection satisfies $|\mathcal{P}|_{\omega_j} \leq L$. Toward that end, let ω_j be any situation for $\mathcal{G}^{\Lambda_j}$; let ω_c be the value of $C_j - A_j$ specified by ω_j; and let $\mathcal{P}_b$ and $\mathcal{P}_r$ be the paths from U to W shown in Figure 17c, where the paths from U to V_j, and from C_j to W, are shortest paths comprising only ordinary edges. First, for any ω_j for which $\omega_c \leq \delta_j - \gamma_j$:

$$\begin{array}{rcll} |\mathcal{P}_b|_{\omega_j} & = & \sigma_j + \max\{-v_j, -\omega_c\} + \omega_c + \gamma_j & \\ & = & \sigma_j + \max\{\omega_c + \gamma_j - v_j, \gamma_j\} & \\ & \leq & \sigma_j + \max\{\delta_j - v_j, \gamma_j\} & (\text{Since } \omega_c \leq \delta_j - \gamma_j) \\ & = & L & (\text{By item (3) above}) \end{array}$$

Second, by Claim 8, there must exist a path from A_j to W that uses edges drawn solely from $\mathcal{S}_{a_jw}$ whose projection has length at most $\delta_j = d_*^{\Lambda_j^-}(A_j, W)$, shown as dashed and purple in Figure 17c. Therefore, for any ω_j for which $\omega_c \geq \delta_j - \gamma_j$:

$$\begin{array}{rcll} |\mathcal{P}_r|_{\omega_j} & \leq & \sigma_j + \max\{-v_j, -\omega_c\} + \delta_j & \\ & = & \sigma_j + \max\{\delta_j - v_j, \delta_j - \omega_c\} & \\ & \leq & \sigma_j + \max\{\delta_j - v_j, \gamma\} & (\text{Since } \omega_c \geq \delta_j - \gamma_j) \\ & = & L & \end{array}$$

As a result, it follows that $d_*^{\Lambda_j}(U,W) \leq L$. However, since we previously showed that $L \leq d_*^{\Lambda_j}(U,W)$, we now have that $L = d_*^{\Lambda_j}(U,W) = d_*^{\Lambda_j^+}(U,W)$.

□

Corollary 1. *When applied to a dispatchable ESTNU graph $\mathcal{G}$, the* `genStandIns` *helper algorithm for* $\texttt{minDisp}_{\text{ESTNU}}$ *correctly computes $d_*(U,W)$ for each pair of timepoints U and W (i.e., when* `genStandIns` *is finished, $d = d_*$).*

Proof. The `genStandIns` algorithm does at most k iterations. In each iteration it explores every possible combination of timepoints V, A, C and W that form a (non-nested) diamond. For each combination, it considers the contingent link (A, x, y, C), a wait edge $(V, C{:}{-}v, A)$, and the distances, $\delta = d(A,W)$ and $\gamma = d(C,W)$. If the duration $C - A = \delta - \gamma \in (x,y]$ and the value, $\theta = \max\{\delta - v, \gamma\} \leq d(V,W)$, it inserts an ordinary stand-in edge (V, θ, W) that represents the strongest ordinary constraint from V to W entailed by the network that has

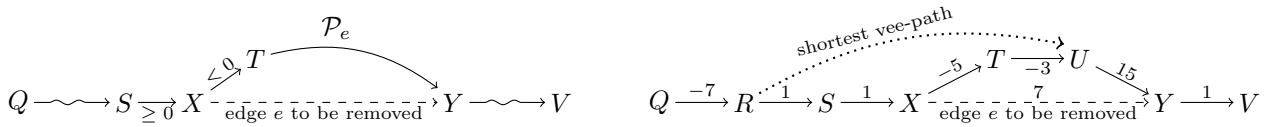


Figure 18: Scenarios (general, left; specific, right) considered in Case 1 of Lemma 5

been found so far. If any such stand-in edges were added during the current iteration, the next iteration begins by running Johnson's algorithm to update all $d(U, W)$ values that might be affected by the new edges.

By induction. For any $j \geq 0$, let $Q(j)$ be the following proposition: For any U and W, if Λ_j^+ is any set of labeled edges for which there exists a set Λ_j such that: (1) $\Lambda_j \subseteq \Lambda_j^+$; (2) Λ_j represents j contingent links; and (3) Λ_j is minimally sufficient for $d_*^{\Lambda_j^+}(U, W)$, then `genStandIns` correctly computes $d_*^{\Lambda_j^+}(U, W)$ by its j^{th} iteration.

Let U and W be any timepoints in $\mathcal{G}$; and let Λ_j^+ be any set of labeled edges from $\mathcal{G}$ for which there exists a set Λ_j satisfying properties (1), (2) and (3) above. Choose any such Λ_j.

Base Case: $j = 0$. Then $\Lambda_j = \emptyset$ and $d_*^{\Lambda_j^+}(U, W) = d_*^{\emptyset}(U, W) = \mathrm{d}(U, W)$, which is obtained by the initial call to Johnson's algorithm.

Recursive Case: $j > 0$. Suppose that $Q(h)$ holds for all $h < j$. Since $j > 0$, Theorem 2 provides that there is a contingent link $\mathrm{CL}_j = (A_j, x_j, y_j, C_j)$ and a wait edge $(V_j, C_j{:}{-}v_j, A_j)$ such that $d_*^{\Lambda_j^+}(U, W) = \mathrm{d}(U, V_j) + \max\{\delta_j - v_j, \gamma_j\}$, where: $\delta_j = d_*^{\Lambda_j^-}(A_j, W)$ and $\gamma_j = d_*^{\Lambda_j^-}(C_j, W) = \mathrm{d}(C_j, W)$, where $\Lambda_j^- = {\Lambda_\mathrm{j}}^{-\mathrm{CL}_j}$ is the same as Λ_j, except that it does not include labeled edges associated with CL_j. Since any subset $\Lambda_{j-1}^- \subseteq \Lambda_j^-$ that is minimally sufficient for $d_*^{\Lambda_j^-}(A_j, W)$ necessarily represents fewer than j contingent links, the inductive hypothesis ensures that δ_j has been correctly computed by the $j - 1^{\text{st}}$ iteration. In addition, $\gamma_j = \mathrm{d}(C_j, W)$ is given by the initial call to Johnson's algorithm. Therefore, by the j^{th} iteration, $d_*^{\Lambda_j^+}(U, W)$ has been correctly computed.

Therefore, $Q(j)$ holds for all $j \geq 0$. But then for any U and W, $d_*(U, W) = d_*^{\Lambda}(U, W)$ is correctly computed by at most the k^{th} iteration, where Λ contains all of the labeled edges from $\mathcal{G}$, which represent k contingent links. □

Corollary 2. *Given $U, W, \Lambda_j, \Lambda_j^+$, and $\mathcal{S}_{uw}^{\Lambda_j}$ as in the proof of Theorem 2, in* any *situation ω_j for $\mathcal{G}^{\Lambda_j}$, there exists a simple path $\mathcal{P}$ from U to W whose only edges are drawn from the structure $\mathcal{S}_{uw}^{\Lambda_j}$ and whose projection satisfies $|\mathcal{P}|_{\omega_j} \leq d_*^{\Lambda_j}(U, W) = d_*^{\Lambda_j^+}(U, W)$.*

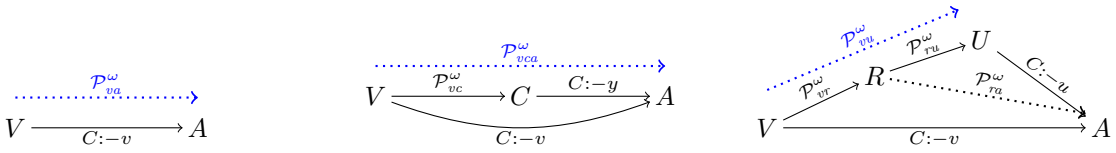
Theorem 3, below, ensures that the wait edges removed by the final step of the $\texttt{minDisp}_{\text{ESTNU}}$ algorithm are not needed for the dispatchability of the network (i.e., their removal cannot threaten its dispatchability). But first, we present an important result about *STN* dispatchability.

Lemma 5. *Suppose that $e = (X, \delta, Y)$ is an (ordinary) edge in a dispatchable* STN *$\mathcal{G}_o$, and $\mathcal{P}_e$ is a shortest vee-path from X to Y in $\mathcal{G}_o$ that does* not *use e. Then removing e from $\mathcal{G}_o$ preserves the dispatchability of $\mathcal{G}_o$.*

Proof. Let $\mathcal{P}_{qv}$ be any SVP from some Q to some V in $\mathcal{G}_o$ that includes e. Since e is an edge in an SVP, it follows that e, by itself, constitutes a shortest path from X to Y. Therefore, the alternative SVP $\mathcal{P}_e$ necessarily satisfies $|\mathcal{P}_e| = |e| = \delta$. Next, let $\mathcal{P}'_{qv}$ be the path obtained from $\mathcal{P}_{qv}$ by replacing e by $\mathcal{P}_e$. By this construction, $|\mathcal{P}'_{qv}| = |\mathcal{P}_{qv}|$; hence, $\mathcal{P}'_{qv}$ is a shortest path. However, suppose that $\mathcal{P}'_{qv}$ is not a vee-path.

Case 1: Replacing e by $\mathcal{P}_e$ inserts a negative edge after a non-negative edge. This scenario is illustrated in general on the lefthand side of Figure 18, where $\mathcal{P}_e$ is the path from X to T to Y. The righthand side shows a specific example, where $e = (X, 7, Y)$, $\mathcal{P}_{qv}$ is the path $QRSXYV$, $\mathcal{P}_e$ is the path $XTUY$, and replacing e by $\mathcal{P}_e$ inserts the negative edge XT after the non-negative edge SX. Note that in this (general or specific) case, $\delta = |e|$ must be non-negative since e follows the non-negative edge SX in a vee-path. As shown in the specific example, let U be the terminus of the last negative edge in $\mathcal{P}_e$; and let R be the terminus of the last non-negative edge in the subpath from Q to X. (It may happen that $Q = R$.) Since the subpath from X to U comprises only negative edges, that subpath must have negative length. Now, since there is a path from R to U in $\mathcal{G}_o$, the dispatchability of $\mathcal{G}_o$ implies that there must be an SVP from R to U, shown as dotted in the figure. Substituting this (dotted) vee-path for the subpath $RSXTU$ in $\mathcal{P}'$, provides an SVP from Q to V that does not use e. (If e were needed by any shortest path $\mathcal{P}_{ru}$ from R to U, then the suffix of $\mathcal{P}_{ru}$ from X to U must be negative, given that XTU is a shortest path with $|XTU| < 0$. But then $\mathcal{P}_{ru}$ cannot be a vee-path, since it would include the non-negative edge e followed by one or more negative edges.)

Case 2: Replacing e by $\mathcal{P}_e$ inserts a non-negative edge before a negative edge. Handled similarly. □



(a) Case 4.1 (b) Case 4.2 (c) Case 4.3

Figure 19: Scenarios addressed in item 4 of Theorem 3

Theorem 3. *Algorithm 1 (*$\texttt{minDisp}_{\text{ESTNU}}$*) preserves the dispatchability of its input ESTNU* $\mathcal{G}$*.*

Proof.

(1) The first step of $\texttt{minDisp}_{\text{ESTNU}}$ inserts the ordinary edges collected by the `genStandIns` helper algorithm. Since they are entailed by the network, the addition of these edges cannot disturb dispatchability.

(2) The second step of $\texttt{minDisp}_{\text{ESTNU}}$ runs the $\texttt{disp}_{\text{STN}}$ algorithm on the ordinary edges, which can insert some ordinary edges while removing others. With no loss of generality, suppose it first inserts all the new edges and then removes edges. Since the edges being inserted are entailed by ordinary paths in the network, their insertion cannot disturb dispatchability. Therefore, let $e = (X, \delta, Y)$ be the first (ordinary) edge whose removal thwarts the ESTNU's dispatchability. Since $\texttt{disp}_{\text{STN}}$ preserves dispatchability among the ordinary edges, there must be an alternative (ordinary) SVP $\mathcal{P}_e$ from X to Y with $|\mathcal{P}_e| \leq |e|$. Furthermore, suppose ω is any situation and $\mathcal{P}$ is an ESTNU path that uses e and whose projection in $\mathcal{G}_\omega$ is an SVP. Now, the dispatchability of the ESTNU $\mathcal{G}$ implies that its projection $\mathcal{G}_\omega$ is a dispatchable STN. Therefore, by Lemma 5, replacing e by $\mathcal{P}_e$ in that projection preserves its dispatchability. Since this holds for every projection $\mathcal{G}_\omega$, replacing e by $\mathcal{P}_e$ in $\mathcal{G}$ cannot threaten the dispatchability of $\mathcal{G}$.

(3) The third step of $\texttt{minDisp}_{\text{ESTNU}}$ is to remove any remaining stand-in edges.

Case 3.1: $\hat{e}$ is a stand-in edge that derives from some individual labeled edge e. By Lemma 2, in any situation ω, $|e|_\omega \leq |\hat{e}|_\omega$. Furthermore, $|\hat{e}|_\omega$ has the same sign as $|e|_\omega$. Therefore, replacing $\hat{e}$ by e in any ESTNU path whose projection is a vee-path will result in an ESTNU path whose projection is a vee-path that can only be shorter or have the same length.

Case 3.2: $\hat{e} = (U, \delta, W)$ is a stand-in edge that derives from some possibly nested diamond structure $\mathcal{S}_{uw}$. In this case, $\delta = d_*(U, W)$. Now, in any situation where the length of an SVP from U to W is less than $|\hat{e}| = \delta = d_*(U, W)$, $\hat{e}$ cannot participate in that SVP and hence is irrelevant. On the other hand, suppose that ω is a situation that is maximal for $d_*(U, W)$ (i.e., where the length of an SVP from U to W equals $\delta = d_*(U, W)$). Now, by Corollary 2, there necessarily exists a simple path $\mathcal{P}$ from U to W whose only edges are drawn from the structure $\mathcal{S}_{uw}$, which does not include $\hat{e}$, and whose projection satisfies $|\mathcal{P}|_\omega \leq d_*(U, W)$. But since ω is maximal for $d_*(U, W)$, that projection cannot be shorter than $d_*(U, W)$; hence, $|\mathcal{P}|_\omega = d_*(U, W) = \delta = |\hat{e}|$. As a result, in *every* projection, there is an SVP from U to W that does not use the edge $\hat{e}$. Then, by Lemma 5, the dispatchability of any STN projection cannot be disturbed by replacing $\hat{e}$ by the corresponding SVP from U to W. Therefore, removing $\hat{e}$ from the ESTNU cannot threaten its dispatchability.

(4) The fourth step of $\texttt{minDisp}_{\text{ESTNU}}$ removes several categories of waits collected by its `markWaits` helper algorithm. We consider each category of wait removals in turn. In each case, the wait edge to be removed is denoted by $(V, C{:}-v, A)$ and the corresponding contingent link by (A, x, y, C) Recall, too, that once `genStandIns` has been completed, $d = d_*$, which remains fixed for the rest of $\texttt{minDisp}_{\text{ESTNU}}$.

Case 4.1: Remove all wait edges $(V, C{:}-v, A)$ for which $\mathrm{d}(V, A) \leq -v$. First, we note that the $\texttt{minDisp}_{\text{ESTNU}}$ algorithm does a pre-processing step that replaces all *weak* waits (i.e., waits for which the wait time v is less than or equal to the minimum duration x of the corresponding contingent link) with (unconditional) ordinary edges. Therefore, we may assume that $-v < -x$, where $-x$ is the length of the stand-in edge for $(V, C{:}-v, A)$. Now, after calling $\texttt{disp}_{\text{STN}}$, which does not affect $d = d_*$, but before removing any remaining stand-in edges, there must have been a simple *ordinary* SVP $\mathcal{P}_{va}$ from V to A of length $\delta = \mathrm{d}(V, A) \leq -v$, some of whose edges may have been stand-in edges, whether associated with individual labeled edges or possibly-nested diamond structures. Furthermore, since $-v < 0$, that SVP must have started with at least one negative edge; and that first negative edge could not be the stand-in edge $(V, -x, A)$ for $(V, C{:}-v, A)$, since the suffix from A to A would have to comprise only zero-length edges and, in any case, the total length of the ordinary path would be $-x > -v$. Now, by Cases 3.1 and 3.2 above, in every situation ω, there exist replacement edges or paths for each of the stand-in edges in $\mathcal{P}_{va}$ that together ensure the existence of an alternative ESTNU path $\mathcal{P}^{\omega}_{va}$ from V to A, shown as blue and dotted in Figure 19a, whose

projection in the situation ω is an SVP. Furthermore, none of the replacements done in Cases 3.1 or 3.2 affect the first negative edge in the original ordinary SVP from V to A. Therefore, $\mathcal{P}^{\omega}_{va}$ cannot include the wait edge $(V, C{:}{-}v, A)$, since it would create a negative-length path from V to V.

Case 4.2: Remove all wait edges $(V, C{:}{-}v, A)$ for which $\mathrm{d}(V, C) < 0$. Let $(V, C{:}{-}v, A)$ be the first such wait edge whose removal thwarts the dispatchability of the ESTNU. After calling $\texttt{disp}_{\text{STN}}$, which does not affect $d = d_*$, but before the third step's removal of stand-in edges, there must have been an ordinary SVP $\mathcal{P}_{vc}$ from V to C, some of whose edges may be stand-in edges, whether arising from individual labeled edges or possibly-nested diamond structures. However, by Cases 3.1 and 3.2 above, in every situation ω, there exist replacement edges or paths for each of those stand-in edges that together ensure the existence of an alternative ESTNU path $\mathcal{P}^{\omega}_{vc}$ from V to C, illustrated in Figure 19b, that does not use any stand-in edges and whose projection in the situation ω is an SVP. Next, let $\mathcal{P}^{\omega}_{vca}$ be the ESTNU path, shown as blue and dotted in the figure, obtained by concatenating $\mathcal{P}^{\omega}_{vc}$ and the UC edge $(C, C{:}{-}y, A)$. Then $|\mathcal{P}^{\omega}_{vca}|_{\omega} = \mathrm{d}(V, C) - \omega_c < -\omega_c \leq \max\{-v, -\omega_c\} = |(V, C{:}{-}v, A)|_{\omega}$. (As seen elsewhere, ω_c denotes the value of $C - A$ in the situation ω.) Therefore, the projection of $\mathcal{P}^{\omega}_{vca}$ is a *shorter* path than the projection of $(V, C{:}{-}v, A)$. Although $\mathcal{P}^{\omega}_{vca}$ might not be an SVP, the dispatchability of $\mathcal{G}_{\omega}$ ensures that there must be an SVP from V to A whose length is at most $|\mathcal{P}^{\omega}_{vca}|_{\omega} < |(V, C{:}{-}v, A)|_{\omega}$. In turn, that implies that $(V, C{:}{-}v, A)$ cannot participate in that SVP. Therefore, removing $(V, C{:}{-}v, A)$ from the ESTNU cannot affect the dispatchability of $\mathcal{G}_{\omega}$. And since the choice of ω was arbitrary, removing that edge cannot affect the ESTNU's dispatchability.

Case 4.3: Remove each wait edge $(V, C{:}{-}v, A)$ for which there is a wait edge $(U, C{:}{-}u, A)$ with $\mathrm{d}(V, U) < 0$ and $\mathrm{d}(V, U) - u \leq -v$. Let $(V, C{:}{-}v, A)$ be the first such wait edge whose removal thwarts the dispatchability of the ESTNU. As in Cases 4.1 and 4.2, after running $\texttt{disp}_{\text{STN}}$, but before removing any remaining stand-in edges, there must have been an ordinary SVP $\mathcal{P}_{vu}$ from V to U, some of whose edges may have been stand-in edges. However, for each situation ω, Cases 3.1 and 3.2 above ensure the existence of replacement edges or paths for those stand-in edges that together ensure the existence of an alternative ESTNU path $\mathcal{P}^{\omega}_{vu}$ from V to U, shown as blue and dotted in Figure 19c, that does not use any stand-in edges and whose projection in the situation ω is an SVP. Hence, $|\mathcal{P}^{\omega}_{vu}|_{\omega} \leq \mathrm{d}(V, U) < 0$. Since $|\mathcal{P}^{\omega}_{vu}|_{\omega} < 0$, that SVP must start with at least one negative edge. Next, let R be the terminus of the last negative edge in $\mathcal{P}^{\omega}_{vu}$; let $\mathcal{P}^{\omega}_{vr}$ be the prefix of $\mathcal{P}^{\omega}_{vu}$ from V to R; and let $\mathcal{P}^{\omega}_{ru}$ be the suffix of $\mathcal{P}^{\omega}_{vu}$ from R to U. Now, since there is a path from R to A, the dispatchability of $\mathcal{G}_{\omega}$ ensures that there is an SVP from R to A. Let $\mathcal{P}^{\omega}_{ra}$ be an ESTNU path whose projection in ω is that SVP from R to A, shown as dotted in the figure. Next, let $\mathcal{P}^{\omega}_{vra}$ be the concatenation of $\mathcal{P}^{\omega}_{vr}$ and $\mathcal{P}^{\omega}_{ra}$. Then the projection of $\mathcal{P}^{\omega}_{vra}$ is a vee-path for which

$$
\begin{array}{rcll}
|\mathcal{P}^{\omega}_{vra}|_{\omega} & = & |\mathcal{P}^{\omega}_{vr}|_{\omega} + |\mathcal{P}^{\omega}_{ra}|_{\omega} & \\
& \leq & |\mathcal{P}^{\omega}_{vr}|_{\omega} + (|\mathcal{P}^{\omega}_{ru}|_{\omega} + |(U, C{:}{-}u, A)|_{\omega_c}) & \text{(Since } \mathcal{P}^{\omega}_{ra} \text{ is a shortest path)} \\
& = & |\mathcal{P}^{\omega}_{vu}|_{\omega} + |(U, C{:}{-}u, A)|_{\omega_c} & \text{(Since } \mathcal{P}^{\omega}_{vu} \text{ is the concatenation of } \mathcal{P}^{\omega}_{vr} \text{ and } \mathcal{P}^{\omega}_{ru}\text{)} \\
& \leq & \mathrm{d}(V, U) + \max\{-u, -\omega_c\} & \\
& \leq & \max\{\mathrm{d}(V, U) - u, \mathrm{d}(V, U) - \omega_c\} & \\
& \leq & \max\{-v, -\omega_c\} & \text{(Since } \mathrm{d}(V, U) - u \leq -v \text{ and } \mathrm{d}(V, U) < 0\text{)} \\
& = & |(V, C{:}{-}v, A)|_{\omega_c}. &
\end{array}
$$

Furthermore, the SVP from R to A cannot include $(V, C{:}{-}v, A)$ since that would imply a chain of negative edges from V to V within the vee-path $\mathcal{P}^{\omega}_{vra}$, contradicting the dispatchability of the network. Therefore, $\mathcal{P}^{\omega}_{vra}$ is an alternative vee-path that is at least as short as the wait edge $(V, C{:}{-}v, A)$. Therefore, the removal of $(V, C{:}{-}v, A)$ cannot thwart the dispatchability of the ESTNU.

□

Theorem 4. *When given a dispatchable ESTNU as input, the* $\texttt{minDisp}_{\text{ESTNU}}$ *algorithm outputs an* equivalent *dispatchable ESTNU having a* minimal *number of edges.*

Proof. The ESTNU output by $\texttt{minDisp}_{\text{ESTNU}}$ is equivalent to the input network since: (1) all fixed waits and stand-in edges are entailed by alternative edges or paths, (2) the $\texttt{disp}_{\text{STN}}$ algorithm generates an equivalent set of ordinary edges, and (3) each removed wait edge is entailed by an alternative path in every situation.

Minimality. Since Theorem 3 ensures that the output ESTNU is dispatchable, it suffices to show that each edge in the output ESTNU is needed for dispatchability (i.e., its removal would thwart dispatchability). To the contrary, suppose that e is some edge *not* needed for dispatchability but belongs to the output ESTNU and, hence, was not removed. For reference, let $\mathcal{G}_o = (\mathcal{T}, \mathcal{E}^*_o)$ be the STN output by $\texttt{disp}_{\text{STN}}$.

Case 1: e is an ordinary edge (U, δ, W). Since all stand-in edges are eventually removed, e cannot be a stand-in edge. Instead, it must be an original ordinary edge or one inserted by $\texttt{disp}_{\text{STN}}$, with $\delta = |e| = d_*(U, W)$. Suppose some set Λ of labeled edges was minimally sufficient for $d_*(U, W)$. If the number of contingent links represented in Λ were greater than zero, then there would be a canonical structure of nested diamonds that would

provide $d_*(U,W)$, contradicting that e was not a stand-in edge. Therefore, $\Lambda = \emptyset$ and $d_*(U,W) = \mathrm{d}(U,W) = \delta$. (Recall that $d_* = d$ at this point in the algorithm.) But $\texttt{disp}_{\texttt{STN}}$ outputs a minimal dispatchable STN, keeping or inserting e, from which it follows that any vee-path $\mathcal{P}$ from U to W in $\mathcal{G}_o$ that does not use e must satisfy $|\mathcal{P}| > \delta$. But that implies that removing e from the ESTNU would yield $d_*(U,W) > \delta = |e|$, contradicting that e is not needed.

Case 2: e is a wait edge $(V, C{:}{-}v, A)$ associated with a contingent link $CL = (A, x, y, C)$. Since e was not removed, the conditions in Cases 4.1, 4.2 and 4.3 of Theorem 3 that are used by the `markWaits` helper algorithm must all be false; hence:

(f1) $d_*(V,A) > -v$;

(f2) $d_*(V,C) \geq 0$; and

(f3) for all *other* waits $(U, C{:}{-}u, A)$, $d_*(V,U) \geq 0$ or $d_*(V,U) - u > -v$.

Let Λ^- denote the set of all labeled edges from $\mathcal{G}$ except those associated with CL; and let $\mathcal{S}_{va}^{\Lambda^-}$ be the canonical structure associated with $d_*^{\Lambda^-}(V,A)$ that is guaranteed by Theorem 2. Let ω be the situation for which (1) $\omega_c = C - A = y$ (i.e., its maximum value); (2) the duration of each contingent link $CL_i = (A_i, x_i, y_i, C_i)$ having labeled edges in the canonical structure $\mathcal{S}_{va}^{\Lambda^-}$ is the value $\delta_i - \gamma_i$ specified in the proof of Theorem 2, where all such durations combine to ensure that ω is maximal for $d_*^{\Lambda^-}(V,A)$; and (3) the duration of every other contingent link (A_g, x_g, y_g, C_g) is its minimum value, x_g.

Since e is not needed to preserve the dispatchability of the ESTNU $\mathcal{G}$, then in the situation ω, there must be a simple ESTNU path $\mathcal{P}^{va}$ from V to A that does not use e, and whose projection satisfies $|\mathcal{P}^{va}|_\omega \leq |e|_\omega = \max\{-\omega_c, -v\} = \max\{-y, -v\} = -v$. Being simple, $\mathcal{P}^{va}$ cannot include the LC edge $(A, c{:}x, C)$ associated with CL; and if it includes the UC edge $(C, C{:}{-}y, A)$ or some wait $(U, C{:}{-}u, A)$ associated with CL, then that edge must be the last edge in $\mathcal{P}^{va}$.

Case 2a: The last edge in $\mathcal{P}^{va}$ is the UC edge $(C, C{:}{-}y, A)$. Since the projection of a negative edge is necessarily negative, then the last edge in the projection of $\mathcal{P}^{va}$ is negative. And since its projection is an SVP, it follows that *every* edge in the projection is negative and, hence, so is every edge in $\mathcal{P}^{va}$. Let $\mathcal{P}^{vc}$ be the subpath of $\mathcal{P}^{va}$ from V to C. Since all of its edges are negative, then in every situation the projection of $\mathcal{P}^{vc}$ must have negative length. Therefore, in every situation, every SVP from V to C must have negative length, implying that $d_*(V,C) < 0$, contradicting condition (f2), above.

Case 2b: The last edge in $\mathcal{P}^{va}$ is a wait edge $(U, C{:}{-}u, A)$ associated with the contingent link CL. Let $\mathcal{P}^{vu}$ be the prefix of $\mathcal{P}^{va}$ from V to U. As in Case 2a, it follows that every edge in $\mathcal{P}^{vu}$ and its SVP projection must be negative, and hence $d_*(V,U) < 0$. In addition, since $-v \geq |\mathcal{P}^{va}|_\omega = |\mathcal{P}^{vu}|_\omega - u$, we get that $|\mathcal{P}^{vu}|_\omega \leq u - v$.

Next, consider any activation timepoint A_i for some contingent link $CL_i = (A_i, x_i, y_i, C_i)$ that appears in $\mathcal{P}^{va}$. By the same logic, $d_*(A_i, A) < 0$. On the other hand, if CL_i is one of the contingent links having edges in the canonical structure $\mathcal{S}_{va}^{\Lambda^-}$, then by the proof of Theorem 2, $\delta_i = d_*(A_i, A)$; and $\delta_i - \gamma_i \in (x_i, y_i]$ implies that $\delta_i \geq \gamma_i + x_i > 0$. In other words, the contingent links having edges in $\mathcal{S}_{va}^{\Lambda^-}$ cannot have any labeled edges in $\mathcal{P}^{va}$ (or $\mathcal{P}^{vu}$). Hence, the UC or wait edges in $\mathcal{P}^{vu}$ necessarily take on their minimum (i.e., *weakest*) values in the situation ω specified above.

But then in *any* situation $\hat{\omega}$, the projections of those UC or wait edges must have lengths that are shorter than or the same as the lengths of their projections in ω. Hence: $|\mathcal{P}^{vu}|_{\hat{\omega}} \leq |\mathcal{P}^{vu}|_\omega \leq u - v$, which implies that every SVP from V to U must have length at most $u - v$, which in turn implies that $d_*(V,U) \leq u - v$ (i.e., $d_*(V,U) - u \leq -v$). Together with $d_*(V,U) < 0$, this contradicts condition (f3), above.

Case 2c: $\mathcal{P}^{va}$ does not contain any labeled edges associated with $CL = (A, x, y, C)$. Since ω is maximal for $d_*^{\Lambda^-}(V,A)$, it follows that in any situation $\hat{\omega}$, $|\mathcal{P}^{va}|_{\hat{\omega}} \leq |\mathcal{P}^{va}|_\omega \leq -v$, which implies that any SVP from V to A must have length at most $-v$, which implies that $d_*(V,A) \leq -v$, contradicting condition (f1), above. □

References

[1] Massimo Cairo, Luke Hunsberger, and Romeo Rizzi. Faster Dynamic Controllablity Checking for Simple Temporal Networks with Uncertainty. In *25th International Symposium on Temporal Representation and Reasoning (TIME-2018)*, volume 120 of *LIPIcs*, pages 8:1–8:16, 2018. doi:10.4230/LIPIcs.TIME.2018.8.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 4th Edition.* MIT Press, 2022. URL: https://mitpress.mit.edu/9780262046305/introduction-to-algorithms.

[3] Rina Dechter, Itay Meiri, and J. Pearl. Temporal Constraint Networks. *Artificial Intelligence*, 49(1-3):61–95, 1991. doi:10.1016/0004-3702(91)90006-6.

[4] Luke Hunsberger. Fixing the semantics for dynamic controllability and providing a more practical characterization of dynamic execution strategies. In *16th International Symposium on Temporal Representation and Reasoning (TIME-2009)*, pages 155–162, 2009. doi:10.1109/TIME.2009.25.

[5] Luke Hunsberger and Roberto Posenato. Speeding up the RUL$^-$ Dynamic-Controllability-Checking Algorithm for Simple Temporal Networks with Uncertainty. In *36th AAAI Conference on Artificial Intelligence (AAAI-22)*, volume 36-9, pages 9776–9785. AAAI Pres, 2022. doi:10.1609/aaai.v36i9.21213.

[6] Luke Hunsberger and Roberto Posenato. A Faster Algorithm for Converting Simple Temporal Networks with Uncertainty into Dispatchable Form. *Information and Computation*, 293(105063):1–21, 2023. doi:10.1016/j.ic.2023.105063.

[7] Luke Hunsberger and Roberto Posenato. Converting Simple Temporal Networks with Uncertainty into Minimal Equivalent Dispatchable Form. In *Proceedings of the Thirty-Fourth International Conference on Automated Planning and Scheduling (ICAPS 2024)*, volume 34, pages 290–300, 2024. doi:10.1609/icaps.v34i1.31487.

[8] Luke Hunsberger and Roberto Posenato. Faster Algorithm for Converting an STNU into Minimal Dispatchable Form. In *31st International Symposium on Temporal Representation and Reasoning (TIME 2024)*, volume 318 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:14, 2024. doi:10.4230/LIPIcs.TIME.2024.11.

[9] Luke Hunsberger and Roberto Posenato. Foundations of Dispatchability for Simple Temporal Networks with Uncertainty. In *16th International Conference on Agents and Artificial Intelligence (ICAART 2024)*, volume 2, pages 253–263. SCITEPRESS, 2024. doi:10.5220/0012360000003636.

[10] Paul Morris. A Structural Characterization of Temporal Dynamic Controllability. In *Principles and Practice of Constraint Programming (CP-2006)*, volume 4204, pages 375–389, 2006. doi:10.1007/11889205_28.

[11] Paul Morris. Dynamic controllability and dispatchability relationships. In *Int. Conf. on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR-2014)*, volume 8451 of *LNCS*, pages 464–479. Springer, 2014. doi:10.1007/978-3-319-07046-9_33.

[12] Paul Morris. The Mathematics of Dispatchability Revisited. In *26th International Conference on Automated Planning and Scheduling (ICAPS-2016)*, pages 244–252, 2016. doi:10.1609/icaps.v26i1.13739.

[13] Paul Morris, Nicola Muscettola, and Thierry Vidal. Dynamic control of plans with temporal uncertainty. In *17th Int. Joint Conf. on Artificial Intelligence (IJCAI-2001)*, volume 1, pages 494–499, 2001. URL: https://www.ijcai.org/Proceedings/01/IJCAI-2001-e.pdf.

[14] Paul H. Morris and Nicola Muscettola. Temporal dynamic controllability revisited. In *20th National Conference on Artificial Intelligence (AAAI-2005)*, pages 1193–1198, 2005. URL: https://www.aaai.org/Papers/AAAI/2005/AAAI05-189.pdf.

[15] Nicola Muscettola, Paul H. Morris, and Ioannis Tsamardinos. Reformulating temporal plans for efficient execution. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning*, KR'98, page 444–452, 1998.

[16] Ioannis Tsamardinos, Nicola Muscettola, and Paul Morris. Fast Transformation of Temporal Plans for Efficient Execution. In *15th National Conf. on Artificial Intelligence (AAAI-1998)*, pages 254–261, 1998. URL: https://cdn.aaai.org/AAAI/1998/AAAI98-035.pdf.

UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**

University of Verona
Department of Computer Science
Strada Le Grazie, 15
I-37134 Verona
Italy

https://www.di.univr.it