Chapter 4

Simple Temporal Networks

The temporal reasoning problems addressed in this thesis are stated and analyzed in terms of Simple Temporal Networks (Dechter, Meiri, and Pearl, 1991). A Simple Temporal Network (STN) comprises a set of time-point variables and binary constraints among those variables. In this thesis, STNs are used to represent (1) temporal constraints among the tasks to which an agent has committed itself; (2) constraints on the durations of actions in act-type and recipe definitions; (3) precedence relations (with temporal offsets) among subacts in recipes; (4) additional temporal constraints associated with particular action instances; and (5) temporal constraints in bids. The standard operations on STNs (e.g., determining whether a solution exists, propagating new constraints, and determining which constraints may be safely added) require only polynomial time.

This chapter summarizes the theory of Simple Temporal Networks and provides some extensions to that theory to facilitate the analysis in Chapter 5. The concepts in the first section derive, directly or indirectly, from Dechter et al. (1991). However, in some cases, the definitions and theorems have been recast in equivalent terms to facilitate their later use.

4.1 Existing Theory of Simple Temporal Networks

Definition 4.1 (Simple Temporal Network) *(Dechter, Meiri, and Pearl, 1991) A* Simple Temporal Network *$\mathcal{S}$ is a pair $(\mathcal{T}, \mathcal{C})$, where $\mathcal{T}$ is a set $\{t_0, t_1, \ldots, t_N\}$ of time-point variables and $\mathcal{C}$ is a finite set of binary constraints on those variables, each constraint having the form $t_j - t_i \leq \delta$, for some real number δ. The "variable" t_0 represents an arbitrary, fixed reference point on the time-line. (In this thesis, we fix t_0 to the value 0 and frequently refer to it as the* zero time-point variable*, or z.) The constraints in $\mathcal{C}$ are called the* explicit constraints *in $\mathcal{S}$.*

Definition 4.2 (STN Solution) *A* solution *to an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ is a complete set of variable assignments*

$$\{z = 0,\ t_1 = w_1,\ \ldots,\ t_N = w_N\},\ \text{each } w_i \in \mathbb{R}$$

that satisfies all the constraints in $\mathcal{C}$.

Definition 4.3 (Consistent STN) *An STN that has at least one solution is called* consistent.

Definition 4.4 (Equivalent STNs) *Two STNs $\mathcal{S}_1$ and $\mathcal{S}_2$ (over the same set of time point variables) are said to be* equivalent, *written $\mathcal{S}_1 \equiv \mathcal{S}_2$, if they admit identical solution sets.*

Unary constraints on time-point variables in an STN are represented as binary constraints involving z, as follows (where L and U are constants):

$$\begin{array}{rclcrclcrcl} L & \leq & t_i & \Leftrightarrow & 0 - t_i & \leq & -L & \Leftrightarrow & z - t_i & \leq & -L \\ t_i & \leq & U & \Leftrightarrow & t_i - 0 & \leq & U & \Leftrightarrow & t_i - z & \leq & U. \end{array}$$

Definition 4.5 (Trivial Constraints, Lean Constraint Set) *A constraint $t_j - t_i \leq \delta$ in $\mathcal{C}$ is called* trivial in $\mathcal{C}$ *if there exists a constraint $t_j - t_i \leq \delta'$ in $\mathcal{C}$ such that $\delta' < \delta$; otherwise, $t_j - t_i \leq \delta$ is called* non-trivial in $\mathcal{C}$. *A set $\mathcal{C}$ of constraints is called* lean *if it contains only non-trivial constraints.*

Proposition 4.6 *Each STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ is equivalent to a unique lean STN $\mathcal{S}_\ell = (\mathcal{T}, \mathcal{C}_\ell)$, where $\mathcal{C}_\ell \subseteq \mathcal{C}$.*

Proof The set $\mathcal{C}_\ell$ may be obtained by removing all trivial constraints from $\mathcal{C}$. Since removing a trivial constraint does not affect the set of solutions, $\mathcal{S}$ and $\mathcal{S}_\ell$ are equivalent. Uniqueness follows from the observation that a set contains only one copy of each element and thus there can exist at most one non-trivial constraint for each ordered pair (t_i, t_j) of time-points. ■

Definition 4.7 (Entailment) *Given sets $\mathcal{C}_1$ and $\mathcal{C}_2$ of constraints over a set $\mathcal{T}$ of time-points, the constraints in $\mathcal{C}_1$ are said to* entail *the constraints in $\mathcal{C}_2$ if any solution to the STN $(\mathcal{T}, \mathcal{C}_1)$ is necessarily a solution to the STN $(\mathcal{T}, \mathcal{C}_2)$.*

Definition 4.8 (Equivalent Constraint Sets) *Given sets $\mathcal{C}_1$ and $\mathcal{C}_2$ of constraints over a set $\mathcal{T}$ of time-points, if the constraints in $\mathcal{C}_1$ entail those in $\mathcal{C}_2$, and the constraints in $\mathcal{C}_2$ entail those in $\mathcal{C}_1$, then $\mathcal{C}_1$ and $\mathcal{C}_2$ are called* equivalent *constraint sets, in which case we write: $\mathcal{C}_1 \equiv \mathcal{C}_2$.*

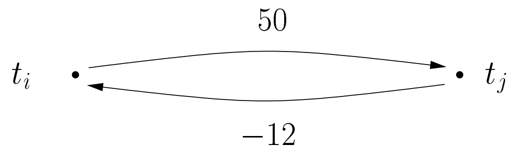
Definition 4.9 (Constraints Consistent with STN) *A set $\mathcal{C}'$ of constraints over the time-points in $\mathcal{T}$ is said to be* consistent *with the STN $(\mathcal{T}, \mathcal{C})$ if the STN $(\mathcal{T}, \mathcal{C} \cup \mathcal{C}')$ is consistent.*

Definition 4.10 (Distance Graph) *(Dechter, Meiri, and Pearl, 1991) The* distance graph *for an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ is a weighted, directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose vertices correspond to the time-points of $\mathcal{S}$ and whose directed edges correspond to the* lean *temporal constraints of $\mathcal{S}$, as follows:*

$$\mathcal{V} = \mathcal{T} \qquad \text{and} \qquad \mathcal{E} = \{(t_i, \delta, t_j) : (t_j - t_i \leq \delta) \in \mathcal{C}_\ell\}.$$

The value δ in an edge (t_i, δ, t_j) is called the length *of the edge.*

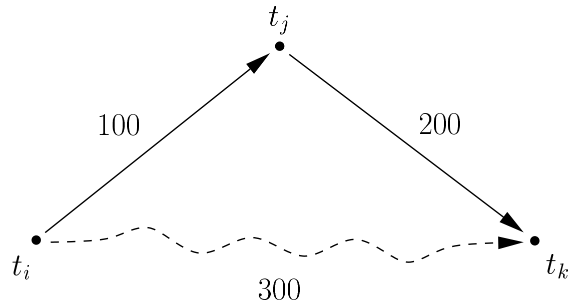
Thus, each *non-trivial* temporal constraint $t_j - t_i \leq \delta$ in an STN is represented in the corresponding distance graph by a directed edge from t_i to t_j with weight (or length) δ. For example, the constraints $t_j - t_i \leq 50$ and $t_i - t_j \leq -12$ would be represented by the edges shown below:



In an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$, the *explicit* constraints in $\mathcal{C}$ may give rise to additional *implicit* constraints.[1] For example, the explicit constraints $t_j - t_i \leq 100$ and $t_k - t_j \leq 200$ combine (using simple arithmetic) to entail the implicit constraint $t_k - t_i \leq 300$, as follows:

$$t_k - t_i \;=\; (t_k - t_j) + (t_j - t_i) \;\leq\; 200 + 100 \;=\; 300.$$

In graphical terms, the edges from t_i to t_j and t_j to t_k form a *path* of length 300 from t_i to t_k, shown as a squiggly arc below.



In general, each path corresponds to an implicit constraint on its endpoints.

Definition 4.11 (Path) *A* path *from t_i to t_j in the distance graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a sequence $P = (\tau_1, \ldots, \tau_m)$ such that:*

- $m \geq 1$*;*
- $\tau_1 = t_i$ *and* $\tau_m = t_j$*;*
- $\tau_1, \ldots, \tau_m \in \mathcal{V}$*; and*
- *for each $i \in \{1, \ldots, m-1\}$, there is an edge of the form $(\tau_i, \delta_i, \tau_{i+1})$ in $\mathcal{E}$.*

(If P is a path in the distance graph of an STN $\mathcal{S}$, then we may also say that P is a path in $\mathcal{S}$.) The length of P, denoted $|P|$, is the sum of the lengths of the edges constituting P:

$$|P| \;=\; \sum_{i=1}^{m-1} \delta_i$$

If $\tau_1 = \tau_m$, then P is called a loop.

[1]For convenience, an explicit constraint is taken to be a special case of an implicit constraint. Thus, the set of implicit constraints contains the set of explicit constraints.

Notice that the length of a path may be negative.

Definition 4.12 (Shortest Path) *A path P from t_i to t_j in a distance graph $\mathcal{G}$ is called a* shortest path *from t_i to t_j in $\mathcal{G}$ if:*

$$|P| \;=\; \min\,\{\, |P'| : P' \text{ is a path from } t_i \text{ to } t_j \text{ in } \mathcal{G} \,\}.$$

Proposition 4.13 *(Dechter, Meiri, and Pearl, 1991) If there exists a path P from t_i to t_j in the distance graph of an STN $\mathcal{S}$, then the implicit constraint, $t_j - t_i \leq |P|$, is entailed by the explicit constraints in $\mathcal{S}$. In other words, the length of P is an upper bound on the temporal difference, $t_j - t_i$, in any solution for $\mathcal{S}$.*

Proof For any path $P = (\tau_1, \ldots, \tau_m)$:

$$\begin{array}{rcccccccc} \tau_m - \tau_1 & = & (\tau_m - \tau_{m-1}) & + & (\tau_{m-1} - \tau_{m-2}) & + & \ldots & + & (\tau_2 - \tau_1) \\ & \leq & \delta_{m-1} & + & \delta_{m-2} & + & \ldots & + & \delta_1 \\ & = & |P|. \;\blacksquare & & & & & & \end{array}$$

Definition 4.14 (Temporal Distance) *(Dechter, Meiri, and Pearl, 1991)*[2] *The* temporal distance *from t_i to t_j in an STN $\mathcal{S}$ is the length of the* shortest path *from t_i to t_j in the distance graph $\mathcal{G}$ (or negative infinity if no such path exists):*

$$dist(t_i, t_j) \;=\; \text{glb}\,\{\, |P| : P \text{ is a path from } t_i \text{ to } t_j \text{ in } \mathcal{G} \,\}.$$

Since path lengths may be negative, temporal distances also may be negative. In fact, if there is a *loop* with negative path length, then some temporal distances will necessarily be $-\infty$. (However, as will be seen, a consistent STN cannot have any loops with negative path length.)

Definition 4.15 (Distance Matrix) *(Dechter, Meiri, and Pearl, 1991)*[3] *The* distance matrix *for an STN $\mathcal{S} = (\mathcal{T},\ \mathcal{C})$ is a matrix $\mathcal{D}$ each entry of which equals the temporal distance between the corresponding pair of time-points in $\mathcal{T}$:*

$$\mathcal{D}(i, j) \;=\; dist(t_i, t_j).$$

Thus, $\mathcal{D}$ is a $|\mathcal{T}|$-by-$|\mathcal{T}|$ matrix. Abusing notation slightly, we may write $\mathcal{D}(t_i, t_j)$ where t_i and t_j are time-point variables rather than indices. Notice that the fixed, zero-time-point variable z is associated with the zeroeth row and column of the distance matrix.

Proposition 4.16 *For any time-points t_i and t_j in an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ with distance matrix $\mathcal{D}$, the following implicit constraints are entailed by the constraints in $\mathcal{C}$:*

$$-\mathcal{D}(t_j, t_i) \;\leq\; t_j - t_i \;\leq\; \mathcal{D}(t_i, t_j).$$

[2]The concept of temporal distance, implicit in Dechter et al., is made explicit in Tsamardinos (2000).

[3]The concept of the distance matrix is implicit in Dechter et al. Tsamardinos (2001) uses the term *distance array*.

If the temporal *distance* $\mathcal{D}(t_i, t_j)$ is positive infinity, which happens if and only if there are no paths from t_i to t_j in the distance graph, then the temporal *difference* $t_j - t_i$ is unconstrained. If the temporal *distance* $\mathcal{D}(t_i, t_j)$ is negative infinity, which happens if and only if there is a loop in the distance graph having negative path length, then the temporal *difference* $t_j - t_i$ is impossibly constrained. Similar remarks apply to $\mathcal{D}(t_j, t_i)$.

Definition 4.17 (Strongest Implicit Constraint) *For any time-points t_i and t_j in an STN $\mathcal{S}$ with distance matrix $\mathcal{D}$, the implicit constraint, $t_j - t_i \leq \mathcal{D}(t_i, t_j)$, is called the* strongest implicit constraint *from t_i to t_j in $\mathcal{S}$.*

The definition of temporal distance in terms of shortest paths leads to the following property.

Proposition 4.18 (The Triangle Inequality) *(Tsamardinos, 2000) For any STN $\mathcal{S}$ with distance matrix $\mathcal{D}$, the following inequality holds among each triple of time-points, t_i, t_j and t_k:*

$$\mathcal{D}(t_i, t_k) \;\leq\; \mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, t_k).$$

Definition 4.19 (Tight Edge/Constraint) *(Morris and Muscettola, 2000) If an explicit constraint (or edge), $(t_j - t_i \leq \delta)$, in an STN $\mathcal{S}$ is such that $\delta = \mathcal{D}(t_i, t_j)$, then that constraint is said to be* tight *in $\mathcal{S}$.*[4]

Proposition 4.20 *Any subpath P' of a shortest path P is itself a shortest path. Hence, any edge along a shortest path must be tight.*

Lemma 4.21 *For an edge E of the form $t_j - t_i \leq \delta$, if E is* tight*, then the following inequalities necessarily hold:*

$$\mathcal{D}(t_i, z) - \mathcal{D}(t_j, z) \;\leq\; \delta \quad \textit{and} \quad \mathcal{D}(z, t_j) - \mathcal{D}(z, t_i) \;\leq\; \delta.$$

Proof The first inequality may be proven as follows:

$$\begin{array}{ll}
\mathcal{D}(t_i, z) \;\leq\; \mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, z) & \text{(Triangle Inequality)} \\
\mathcal{D}(t_i, z) \;\leq\; \delta + \mathcal{D}(t_j, z) & \text{(Since } E \text{ is a tight edge)} \\
\mathcal{D}(t_i, z) - \mathcal{D}(t_j, z) \;\leq\; \delta & \text{(Rearrange terms)}
\end{array}$$

The second inequality follows from applying the Triangle Inequality to the edge from z to t_j. ∎

Proposition 4.22 (Computing the Distance Matrix) *As observed by Dechter et al. (1991), since the entry $\mathcal{D}(t_i, t_j)$ is equal to the length of the shortest path from t_i to t_j in the graph $\mathcal{G}$, the distance matrix for a given STN may be computed from scratch in polynomial time, for example, using Floyd-Warshall's $O(|\mathcal{T}|^3)$ "all-pairs shortest-path" algorithm (Cormen, Leiserson, and Rivest, 1990).*

[4] Dechter et al. (1991) define a tightness relation among STNs that is altogether different from an edge being tight in an STN.

4.1.1 The d-graph and the d-STN

Although the distance graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ for an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ is an important analytical tool, another type of graph, called the *d-graph* of $\mathcal{S}$, is also very important. For example, Theorem 4.27 (below) shows that any set of variable assignments satisfying the constraints in the d-graph of $\mathcal{S}$ can always be extended to a solution for $\mathcal{S}$.

Whereas the edges in the distance graph correspond one-to-one to the (non-trivial) constraints in $\mathcal{C}$, the edges in the d-graph correspond one-to-one to the *shortest paths* in $\mathcal{S}$.[5] Thus, whereas the distance graph contains only $|\mathcal{C}_\ell|$ edges and only implicitly specifies temporal distances in $\mathcal{S}$ in terms of shortest paths, the d-graph contains $|\mathcal{T}|^2$ edges, each of which explicitly specifies a temporal distance in $\mathcal{S}$.

Definition 4.23 (d-Graph) *(Dechter, Meiri, and Pearl, 1991) The* d-graph *for an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ is a* complete, *directed graph $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$ whose vertices correspond to the time-points of $\mathcal{S}$ and whose edges correspond to the temporal* distances *among those time-points:*

$$\mathcal{V}_d = \mathcal{T} \quad \text{and} \quad \mathcal{E}_d = \{(t_i, \mathcal{D}(i,j), t_j) : t_i, t_j \in \mathcal{T}\}.$$

Corresponding to the d-graph $\mathcal{G}_d$ (of $\mathcal{S}$) is an STN $\mathcal{S}_d$ called the d-STN (of $\mathcal{S}$).

Definition 4.24 (d-STN) *(Dechter, Meiri, and Pearl, 1991)*[6] *The* d-STN *of an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ is an STN $\mathcal{S}_d = (\mathcal{T}, \mathcal{C}_d)$ whose time-points are the same as those of $\mathcal{S}$, but whose constraints correspond directly to the edges in the d-graph of $\mathcal{S}$ (and hence to the entries in the distance matrix of $\mathcal{S}$), as follows:*

$$\mathcal{C}_d \ = \ \{t_j - t_i \leq \mathcal{D}(i,j) : t_i, t_j \in \mathcal{T}\}.$$

Figure 4.1 illustrates the relationships among $\mathcal{S}$, its distance graph $\mathcal{G}$, its d-graph $\mathcal{G}_d$, and its d-STN $\mathcal{S}_d$. Notice that the d-graph of $\mathcal{S}$ is the distance graph of the d-STN of $\mathcal{S}$.

4.1.2 The Decomposability of a Consistent STN

Definition 4.25 (Relevant Constraints) *Given an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ and a non-empty set of time-points $\mathcal{T}' \subseteq \mathcal{T}$, the set of constraints in $\mathcal{C}$* relevant *to $\mathcal{T}'$ is denoted by $\mathcal{C}|_{\mathcal{T}'}$, where:*

$$\mathcal{C}|_{\mathcal{T}'} = \{(t_j - t_i \leq \delta) \in \mathcal{C} : t_i, t_j \in \mathcal{T}'\}.$$

Definition 4.26 (Locally Consistent Assignment) *(Dechter, Meiri, and Pearl, 1991) Given sets of time-points $\mathcal{T}' \subseteq \mathcal{T}$ and a set of constraints $\mathcal{C}$ over the time-points in $\mathcal{T}$, a set of variable assignments for the time-points in $\mathcal{T}'$ is called* locally consistent *with respect to $\mathcal{C}$ if those variable assignments satisfy the constraints in $\mathcal{C}|_{\mathcal{T}'}$.*

---

[5]Morris and Muscettola (2000) refer to the d-graph as the *all-pairs graph*.

[6]We use the term d-STN to highlight the connection between d-STNs and d-graphs. Dechter et al. do not use this term. Instead, they refer to what we are calling the d-STN as the *minimal network representation* for $\mathcal{S}$—minimal in the sense that it explicitly specifies the minimal *domain* for each temporal difference.

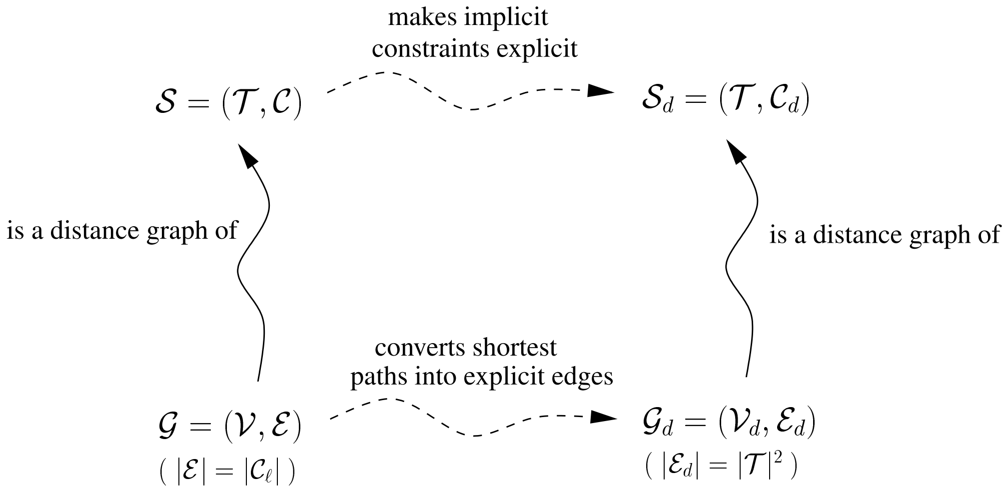


Figure 4.1: The relationships among $\mathcal{S}$, $\mathcal{G}$, $\mathcal{S}_d$ and $\mathcal{G}_d$

In general, a locally consistent assignment is not guaranteed to be extendible to a solution for an entire STN. For example, if $\mathcal{T}' = \{z, t_1\}$, then a locally consistent assignment for the variables in $\mathcal{T}'$ consists solely of an assignment of t_1 to a value that satisfies all of the constraints in $\mathcal{C}$ involving z and t_1 (i.e., all unary constraints on t_1). In general, such an assignment is not guaranteed to satisfy other constraints in $\mathcal{C}$, say, among t_1 and some other time-point t_2. However, the following theorem says that a locally consistent assignment with respect to $\mathcal{C}_d$ (i.e., the constraints in the d-STN of $\mathcal{S}$) can always be extended to a solution for $\mathcal{S}$.

Theorem 4.27 (Decomposability Relative to d-STN) *(Dechter, Meiri, and Pearl, 1991) Let $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ be a consistent STN. Let $\mathcal{C}_d$ be the constraints in the d-STN associated with $\mathcal{S}$. Then any set of variable assignments to time-points in any set $\mathcal{T}' \subseteq \mathcal{T}$ that is locally consistent with respect to $\mathcal{C}_d$ can be extended to a solution for $\mathcal{S}$.*

Corollary 4.28 *(Dechter, Meiri, and Pearl, 1991) Two STNs are equivalent if and only if their distance matrices are identical.*

Proposition 4.29 *(Dechter, Meiri, and Pearl, 1991) The distance matrices for $\mathcal{S}$ and $\mathcal{S}_d$ are identical (i.e., $\mathcal{D} = \mathcal{D}_d$) and, thus, $\mathcal{S} \equiv \mathcal{S}_d$.*

Theorem 4.30 *(Dechter, Meiri, and Pearl, 1991)*[7] *An STN $\mathcal{S}$ is consistent if and only if its corresponding distance graph $\mathcal{G}$ has no negative cycles (i.e., if and only if the path length around any loop is non-negative).*

[7] Dechter et al. cite others regarding this theorem (Shostak, 1981; Liao and Wong, 1983; Leiserson and Saxe, 1983).

Corollary 4.31 *An STN $\mathcal{S}$ is consistent if and only if all of the diagonal entries in its distance matrix are zero:*

$$\mathcal{D}(t_i, t_i) \ = \ 0, \text{ for all } t_i \in \mathcal{T}.$$

Proposition 4.32 *(Tsamardinos, 2000) Given Theorem 4.30, the following inequality holds among each pair of time-points t_i and t_j in a consistent STN:*

$$\mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, t_i) \geq 0.$$

Proposition 4.32 says that the length of any loop from t_i to t_j and back to t_i is non-negative.

4.1.3 Adding Constraints to an STN

Typically, adding a constraint to an STN causes some entries in the distance matrix to change.[8] The following theorem specifies which constraints can be added to an STN without threatening its consistency.

Theorem 4.33 *(Dechter, Meiri, and Pearl, 1991) For any time-points t_i and t_j in a consistent STN $\mathcal{S}$, the new constraint $t_j - t_i \leq \delta$ will not threaten the consistency of $\mathcal{S}$ if and only if δ satisfies $-\mathcal{D}(t_j, t_i) \leq \delta$. Furthermore, $\mathcal{S}$ has a solution in which $t_j - t_i = \sigma$ if and only if $\sigma \in [-\mathcal{D}(t_j, t_i), \mathcal{D}(t_i, t_j)]$.*

Corollary 4.34 *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be the distance matrices for the consistent STNs $\mathcal{S}_1 = (\mathcal{T}, \mathcal{C}_1)$ and $\mathcal{S}_2 = (\mathcal{T}, \mathcal{C}_2)$. Then the following are equivalent:*

- *The constraints in $\mathcal{C}_1$ entail the constraints in $\mathcal{C}_2$.*
- *$\mathcal{D}_1(t_i, t_j) \leq \mathcal{D}_2(t_i, t_j)$, for all $t_i, t_j \in \mathcal{T}$.*

Proof ($\Rightarrow$) Suppose the constraints in $\mathcal{C}_1$ entail those in $\mathcal{C}_2$. Let $t_i, t_j \in \mathcal{T}$ be arbitrary. By Theorem 4.33, there exists a solution $\mathcal{X}_1$ to $\mathcal{S}_1$ for which $t_j - t_i = \mathcal{D}_1(t_i, t_j)$. Since the constraints in $\mathcal{C}_1$ entail the constraints in $\mathcal{C}_2$ (cf. Definition 4.7), $\mathcal{X}_1$ must also be a solution for $\mathcal{S}_2$, which, by Proposition 4.16, implies that $t_j - t_i \leq \mathcal{D}_2(t_i, t_j)$. Thus,

$$\mathcal{D}_1(t_i, t_j) \ = \ t_j - t_i \ \leq \ \mathcal{D}_2(t_i, t_j).$$

($\Leftarrow$) Suppose $\mathcal{D}_1(t_i, t_j) \leq \mathcal{D}_2(t_i, t_j)$, for all $t_i, t_j \in \mathcal{T}$. Let $\mathcal{X}_1$ be any solution for $(\mathcal{T}, \mathcal{C}_1)$. Let E: $t_j - t_i \leq \delta$ be an arbitrary constraint in $\mathcal{C}_2$. Since $\mathcal{X}_1$ is a solution for $(\mathcal{T}, \mathcal{C}_1)$, we have that $t_j - t_i \ \leq \ \mathcal{D}_1(t_i, t_j)$. In addition, since E is a constraint in $\mathcal{C}_2$, we have that $\mathcal{D}_2(t_i, t_j) \ \leq \ \delta$. Putting these inequalities together with the assumption that $\mathcal{D}_1(t_i, t_j) \ \leq \ \mathcal{D}_2(t_i, t_j)$ yields the following:

$$t_j - t_i \ \leq \ \mathcal{D}_1(t_i, t_j) \ \leq \mathcal{D}_2(t_i, t_j) \ \leq \delta,$$

which implies that E is satisfied in $\mathcal{X}_1$. Since E was arbitrary in $\mathcal{C}_2$, $\mathcal{X}_1$ is a solution for $(\mathcal{T}, \mathcal{C}_2)$. Since $\mathcal{X}_1$ was arbitrary, the constraints in $\mathcal{C}_1$ entail those in $\mathcal{C}_2$. ∎

[8]When a new constraint is added to the network, rather than recomputing the distance matrix from scratch, it is more efficient to incrementally update the distance matrix using constraint propagation techniques (Chleq, 1995).

Corollary 4.35 *The quantity* $\mathcal{D}(t_i, t_j)+\mathcal{D}(t_j, t_i)$*, which specifies the length of the interval* $[-\mathcal{D}(t_j, t_i), \mathcal{D}(t_i, t_j)]$*, also specifies the maximum amount by which the strongest implicit constraint from* t_i *to* t_j *may be* tightened.

Corollary 4.35 says that the quantity $\mathcal{D}(t_i, t_j)+\mathcal{D}(t_j, t_i)$ specifies the maximum amount by which the constraints joining the time-points t_i and t_j (in either direction) may be tightened. Adding a constraint $t_j - t_i \leq \delta$ in the extreme case where $\delta = -\mathcal{D}(t_j, t_i)$ (recall Theorem 4.33), causes the *updated* distance matrix entries to satisfy:

$$-\mathcal{D}(t_j, t_i) \ = \ t_j - t_i \ = \ \mathcal{D}(t_i, t_j).$$

In such a case, the temporal difference $t_j - t_i$ is constrained to be fixed; equivalently, $\mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, t_i) = 0$.

Definition 4.36 (Rigidly Connected Time-Points) *(Tsamardinos, Muscettola, and Morris, 1998; Gerevini, Perini, and Ricci, 1996; Wetprasit and Sattar, 1998) The time-points* t_i *and* t_j *are said to be* rigidly connected *in an STN* $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ *if:*

$$\mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, t_i) = 0.$$

A set of time-points $\mathcal{T}_R \subseteq \mathcal{T}$ *forms a* rigidly-connected component *(or a* rigid component*) in* $\mathcal{S}$ *if each pair of time-points in* $\mathcal{T}_R$ *is rigidly connected in* $\mathcal{S}$*.*

4.1.4 Measures of Flexibility and Rigidity for an STN

If the time-points of an STN represent the execution times of tasks to which an agent is committed, then the more flexible the constraints in that STN, the greater the flexibility of that agent in, for example, integrating new tasks into its schedule. (Tasks subject to flexible constraints can be moved around to accommodate new tasks.) Thus, the flexibility of an STN measures one aspect of the *goodness* of an STN from the viewpoint of the agent. Similarly, the rigidity of an STN is a measure of the *badness* of an STN from the agent's viewpoint.[9]

The *relative flexibility* of the time-points t_i and t_j (Definition 4.37, below) specifies the maximum amount by which the current strongest implicit constraint on t_i and t_j may be strengthened (in either direction). The relative flexibility of the pair, t_i and t_j—which is the same as the relative flexibility of the pair t_j and t_i—is a number in the interval $[0, \infty)$. The relative flexibility is 0 if the time-points are rigidly connected, ∞ if their temporal difference is unconstrained. The *relative rigidity* of a pair of time-points (Definition 4.38, below) is (almost) the reciprocal of their relative flexibility. The *root-mean-square (RMS) rigidity* of an STN (Definition 4.38) is defined in terms of the relative rigidity among each pair of time-points in the STN.

[9]The measure of rigidity defined here is similar to the measure of rigidity defined in an earlier paper (Hunsberger, 2002b).

Definition 4.37 (Relative Flexibility) *Given time-points t_i and t_j in a consistent STN, the relative* flexibility *of t_i and t_j is the (non-negative) quantity:*

$$Flex(t_i, t_j) \;=\; \mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, t_i).$$

Definition 4.38 (Relative Rigidity, RMS Rigidity) *The* relative rigidity *of the pair of time-points t_i and t_j in a consistent STN is the quantity:*

$$Rig(t_i, t_j) \;=\; \frac{1}{1 + Flex(t_i, t_j)} \;=\; \frac{1}{1 + \mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, t_i)}.$$

The RMS rigidity *of a consistent STN $\mathcal{S}$ is the quantity:*

$$Rig(\mathcal{S}) \;=\; \sqrt{\frac{2}{|\mathcal{T}|(|\mathcal{T}| - 1)} \sum_{i<j} [Rig(t_i, t_j)]^2}\;.$$

From $Flex(t_i, t_j) \in [0, \infty)$, it follows that $Rig(t_i, t_j) \in [0, 1]$ and $Rig(\mathcal{S}) \in [0, 1]$. If t_i and t_j are part of a rigid component, then $Flex(t_i, t_j) = 0$ and $Rig(t_i, t_j) = 1$. Similarly, if $\mathcal{S}$ is completely rigid, then $Rig(\mathcal{S}) = 1$. At the opposite extreme, if $\mathcal{S}$ is completely unconstrained, then $Rig(\mathcal{S}) = 0$.

A pair of time-points may be rigidly connected to one another, yet not be fixed on the time-line. However, points that are rigidly connected to the zero time-point z, which is fixed on the time-line, are themselves necessarily fixed on the time-line.

Proposition 4.39 *A time-point t_i is rigidly connected to the zero time-point z if and only if t_i is constrained to be fixed. For example, $t_j - z = 10$ if and only if $t_j = 10$.*

4.1.5 Dealing with Rigid Components in an STN

Figure 4.2 shows an STN with two rigid components, $\{t_1, t_2, t_3\}$ and $\{t_4, t_5, t_6\}$. Several researchers (Tsamardinos, 2000; Gerevini, Perini, and Ricci, 1996; Wetprasit and Sattar, 1998) have presented algorithms for effectively *decoupling* rigid components from an STN.[10] For example, Tsamardinos (2000) presents an algorithm that:

(1) selects from each rigid component RC a single time-point t_{RC} that will represent the rigid component in the modified network;

(2) rearranges the edges in the network so that any edge in the modified network that interacts with a time-point in a rigid component RC does so by interacting with the representative t_{RC} of that component; and

[10]These researchers do not refer to this process as "decoupling" the rigid components from the network. Instead, they invariably refer to it as "collapsing" the rigid components down to single points. The word "decoupling" was used here to highlight the connection between these algorithms and the much more general temporal decoupling problems that are the subject of Chapter 5.

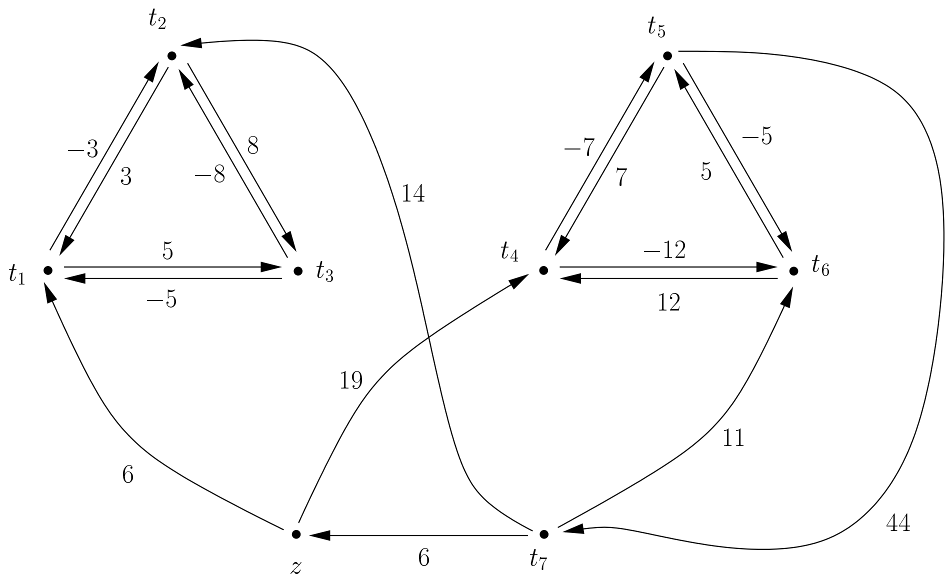


Figure 4.2: An STN with rigid components

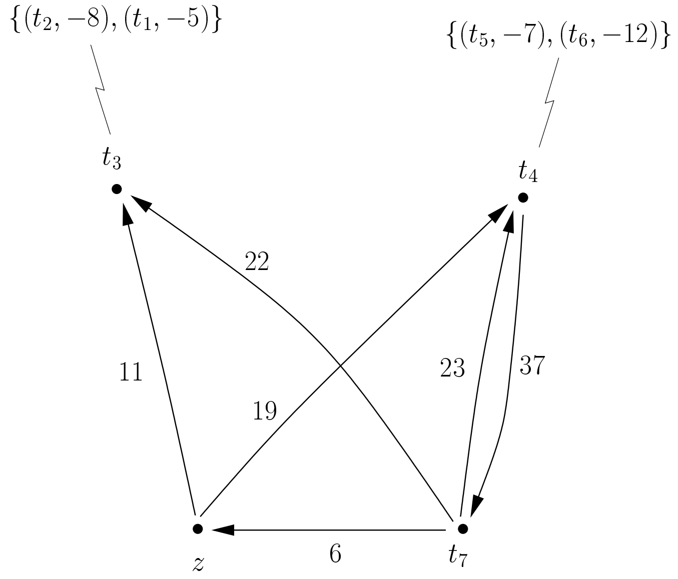


Figure 4.3: The STN from Figure 4.2 after collapsing its rigid components

(3) associates with each representative, additional information sufficient to reconstruct that representative's component.

When applied to the STN from Figure 4.2, the above algorithm generates the STN shown in Figure 4.3 (assuming that the time-points t_3 and t_4 are the chosen representatives). Notice that the algorithm has modified each edge involving a non-representative time-point in a rigid component to ensure that the edge's interface with the rigid component is through the component's representative. For example, the edge, $t_7 - t_5 \leq 44$, has been converted into the edge, $t_7 - t_4 \leq 37$, since t_4 is the representative for the rigid component containing t_5. Notice that the weight of the edge has been modified from 44 to 37, the difference of -7 being the distance from t_4 to t_5 in that component.

The additional information associated with each component's representative may take many forms. In Figure 4.3, each component's representative has a set of pairs associated with it, one pair for each non-representative time-point in the component, each offset specifying the distance from the representative to the non-representative time-point. For example, the pair $(t_2, -8)$ associated with the representative t_3 specifies that the distance from t_3 to t_2 is -8 in the rigid component represented by t_3.

4.2 Some Useful Extensions to the Theory of STNs

It is sometimes desirable (e.g., when transmitting a set of temporal constraints in a message) to represent an STN by an equivalent STN having the minimum number of explicit constraints. This section derives a canonical STN representation that is guaranteed to have the minimum number of explicit constraints. The canonical representation is constructed by first decoupling the rigid components as described in Section 4.1, and then removing all *dominated* constraints from the rest of the network. (A constraint is said to be dominated if removing it from the network does not change the set of solutions.)

This section also introduces the concept of a d-subnetwork, which is used in several key theorems in the next chapter.

Definition 4.40 (Relevant Distance Submatrix) *Let $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ be an STN with distance matrix $\mathcal{D}$. Let $\mathcal{T}'$ be a set of time-points such that $z \in \mathcal{T}' \subseteq \mathcal{T}$. The* distance submatrix *of* $\mathcal{D}$ relevant to $\mathcal{T}'$, *denoted $\mathcal{D}|_{\mathcal{T}'}$, is the $|\mathcal{T}'|$-by-$|\mathcal{T}'|$ matrix given by:*

$$\mathcal{D}|_{\mathcal{T}'}(t_i, t_j) \;=\; \mathcal{D}(t_i, t_j), \text{ for each } t_i, t_j \in \mathcal{T}'.$$

Notice that the indices of the two matrices typically do not correspond. Using variable names instead of integer indices to refer to the entries of the matrices sweeps this trivial problem under the rug.

Definition 4.41 (d-Subnetwork) *Given a consistent STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ and a set of time-points $\mathcal{T}'$ such that $z \in \mathcal{T}' \subseteq \mathcal{T}$, the* d-subnetwork *of $\mathcal{S}$ with respect to $\mathcal{T}'$ is the the STN $\mathcal{S}_d|_{\mathcal{T}'} = (\mathcal{T}', \mathcal{C}_d|_{\mathcal{T}'})$, where (combining Definitions 4.24 and 4.25):*

$$\mathcal{C}_d|_{\mathcal{T}'} = \{(t_j - t_i \leq \mathcal{D}(t_i, t_j)) : t_i, t_j \in \mathcal{T}'\}.$$

In other words, for each pair of time-points t_i and t_j in $\mathcal{T}'$, the d-subnetwork contains an explicit constraint of the form $t_j - t_i \leq \mathcal{D}(t_i, t_j)$. Notice that a shortest path from t_i to t_j in $\mathcal{S}$ may pass through points that are not in $\mathcal{T}'$. However, in the d-subnetwork, the *edge* from t_i to t_j has the same length as the shortest path from t_i to t_j in $\mathcal{S}$.

Lemma 4.42 *Let $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ be an STN with distance matrix $\mathcal{D}$. Let $\mathcal{T}'$ be an arbitrary set of time-points such that $z \in \mathcal{T}' \subseteq \mathcal{T}$. Then the distance matrix for the d-subnetwork $\mathcal{S}_d|_{\mathcal{T}'}$ is identical to the distance submatrix of $\mathcal{D}_d$ relevant to $\mathcal{T}'$. Thus, the notation $\mathcal{D}_d|_{\mathcal{T}'}$ is unambiguous.*

Proof Let $\mathcal{D}_{[\mathcal{S}_d|_{\mathcal{T}'}]}$ be the distance matrix for $\mathcal{S}_d|_{\mathcal{T}'}$. Since any solution to $\mathcal{S}$ must satisfy the constraints in $\mathcal{C}_d|_{\mathcal{T}'}$, the constraints in $\mathcal{C}$ entail the constraints in $\mathcal{C}_d|_{\mathcal{T}'}$. Thus, by Corollary 4.34, $\mathcal{D}(t_i, t_j) \leq \mathcal{D}_{[\mathcal{S}_d|_{\mathcal{T}'}]}(t_i, t_j)$, for all t_i and t_j in $\mathcal{T}'$. However, for each $t_i, t_j \in \mathcal{T}'$, the constraint $t_j - t_i \leq \mathcal{D}(t_i, t_j)$ is in $\mathcal{C}_d|_{\mathcal{T}'}$, which implies that the opposite inequality, $\mathcal{D}_{[\mathcal{S}_d|_{\mathcal{T}'}]}(t_i, t_j) \leq \mathcal{D}(t_i, t_j)$, also holds. Thus, $\mathcal{D}_{[\mathcal{S}_d|_{\mathcal{T}'}]}(t_i, t_j) = \mathcal{D}(t_i, t_j)$ holds for all t_i and t_j in $\mathcal{T}'$. Finally, since $\mathcal{D} = \mathcal{D}_d$ (by Proposition 4.29), we have that $\mathcal{D}_{[\mathcal{S}_d|_{\mathcal{T}'}]} = (\mathcal{D}_d)|_{\mathcal{T}'}$. Thus, the notation $\mathcal{D}_d|_{\mathcal{T}'}$ may be used to refer to both the submatrix of $\mathcal{D}_d$ with respect to $\mathcal{T}'$ and the distance matrix for the d-subnetwork of $\mathcal{S}$ with respect to $\mathcal{T}'$ (i.e., what has been called $\mathcal{D}_{[\mathcal{S}_d|_{\mathcal{T}'}]}$ in this proof). ■

In terms of d-subnetworks, Theorem 4.27 may be restated as follows:

> Let $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ be a consistent STN and $\mathcal{T}'$ a set of time-points such that $z \in \mathcal{T}' \subseteq \mathcal{T}$. Any solution to the d-subnetwork $\mathcal{S}_d|_{\mathcal{T}'}$ of $\mathcal{S}$ can be extended to a solution for $\mathcal{S}$.

We now provide a slight generalization of this theorem that will be quite useful in the next chapter.

Theorem 4.43 *Let $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ be a consistent STN. Let $\mathcal{C}'$ be any set of constraints over the time-points in $\mathcal{T}'$, where $z \in \mathcal{T}' \subseteq \mathcal{T}$. If the constraints in $\mathcal{C}'$ are consistent with $\mathcal{S}_d|_{\mathcal{T}'}$, then they are also consistent with $\mathcal{S}$.*

Proof Suppose that the constraints in $\mathcal{C}'$ are consistent with $\mathcal{S}_d|_{\mathcal{T}'} = (\mathcal{T}', \mathcal{C}_d|_{\mathcal{T}'})$. By Definition 4.9, this implies that the STN $\mathcal{S}' = (\mathcal{T}', \mathcal{C}_d|_{\mathcal{T}'} \cup \mathcal{C}')$ is consistent. Thus, there is a solution $\mathcal{X}'$ to $\mathcal{S}'$ (i.e., a set of assignments to the time-points in $\mathcal{T}'$ satisfying the constraints in $\mathcal{C}_d|_{\mathcal{T}'} \cup \mathcal{C}'$). In particular, the solution $\mathcal{X}'$ satisfies all the constraints in $\mathcal{C}_d|_{\mathcal{T}'}$. However, this implies that the solution $\mathcal{X}'$ also satisfies all of the constraints in $\mathcal{C}_d$ (since all of the constraints in $\mathcal{C}_d$ relevant to $\mathcal{T}'$ are contained in $\mathcal{C}_d|_{\mathcal{T}'}$). Thus, by Theorem 4.27, the set of variable assignments in $\mathcal{X}'$ must be extendible to a solution $\mathcal{X}$ for $\mathcal{S}$. Since the solution $\mathcal{X}'$ satisfies all the constraints in $\mathcal{C}'$, and since the constraints in $\mathcal{C}'$ only involve time-points in $\mathcal{T}'$, the solution $\mathcal{X}$ for $\mathcal{S}$ must also be a solution for $(\mathcal{T}, \mathcal{C} \cup \mathcal{C}')$. Thus, $(\mathcal{T}, \mathcal{C} \cup \mathcal{C}')$ is consistent (i.e., the constraints in $\mathcal{C}'$ are consistent with $\mathcal{S}$). ■

We now consider a notion of *dominance* among constraints in an STN. Our definition of dominance, our characterization of dominance in an STN without rigid components, and our algorithm for finding the *undominated constraints* in an STN are related to pre-existing work (Tsamardinos, 2000; Tsamardinos, Muscettola, and Morris, 1998). The key difference is that the definition of dominance in the pre-existing work is tied to the real-time execution of tasks in an STN, whereas ours is not.

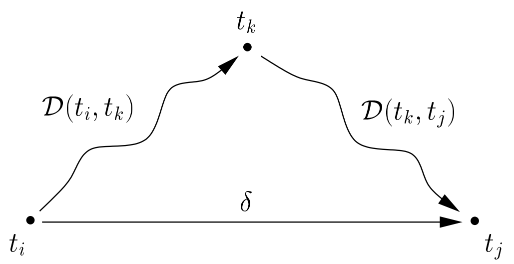
Definition 4.44 (Dominance) *An explicit constraint (or edge) $t_j - t_i \leq \delta$ is* dominated *in a lean STN $\mathcal{S}$ if removing that edge would result in no change to the distance matrix for $\mathcal{S}$.*

Theorem 4.45 (Characterizing Dominance) *Consider an edge E of the form $t_j - t_i \leq \delta$ in a consistent STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$.*

- *If $\mathcal{D}(t_i, t_j) < \delta$ (i.e., if E is not tight in $\mathcal{S}$), then E is dominated in $\mathcal{S}$.*
- *If $\delta = \mathcal{D}(t_i, t_j)$ (i.e., if E is tight in $\mathcal{S}$)* and there are no rigid components in $\mathcal{S}$, *then the edge $t_j - t_i \leq \delta$ is dominated in $\mathcal{S}$ if and only if there is some* other *point $t_k \in \mathcal{T}$ such that:*

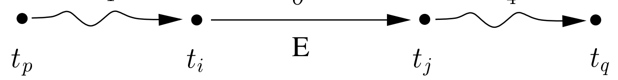
  $$\delta \ = \ \mathcal{D}(t_i, t_k) + \mathcal{D}(t_k, t_j),$$

  as illustrated below.



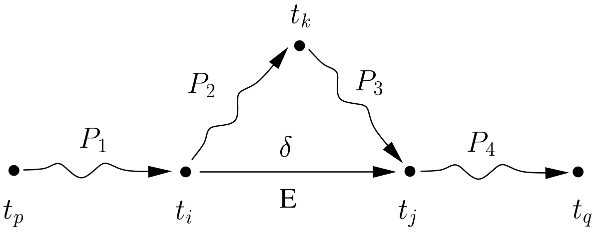
Proof For the first part, if E is not tight, then (by Proposition 4.20) E cannot lie along any shortest path. Hence, removing E could not cause a change to the distance matrix.

For the second part, suppose that $\mathcal{S}$ has no rigid components and that the *tight* edge E, $t_j - t_i \leq \delta$, is *not* dominated in $\mathcal{S}$. Then E must be part of some shortest path P from some t_p to some t_q such that removing E from $\mathcal{S}$ would cause $\mathcal{D}(t_p, t_q)$ to increase (i.e., the subsequent shortest path from t_p to t_q would be longer than P). P has the form shown below:



where the squiggly arcs denote (possibly empty) *paths*. (Without loss of generality, E may be assumed to occur only once on the path P.)

Suppose there exists some t_k such that $\mathcal{D}(t_i, t_k) + \mathcal{D}(t_k, t_j) = \delta$. Then there exist shortest paths P_2 (from t_i to t_k) and P_3 (from t_k to t_j), as shown below



where $|P_2| + |P_3| = \delta$.

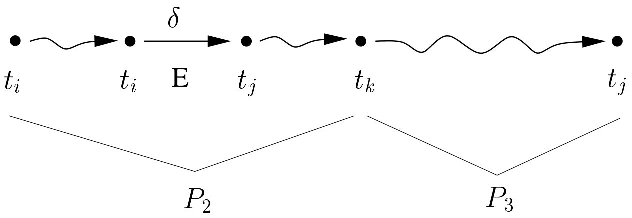
Let P' be the concatenation of the paths P_1, P_2, P_3 and P_4. Then $|P'|$ is given by:

$$|P'| \;=\; |P_1| + |P_2| + |P_3| + |P_4| \;=\; |P_1| + \delta + |P_4| \;=\; |P| \;=\; \mathcal{D}(t_p, t_q).$$

Thus, P' is also a shortest path from t_p to t_q.

E must lie along the path P' since otherwise removing E from $\mathcal{S}$ would not change the value of $\mathcal{D}(t_p, t_q)$. Since E does not lie along P_1 or P_4, it must lie along P_2 or P_3.

Suppose E lies along P_2. Then P_2 and P_3 look like this:



Thus,

$$\begin{aligned} & \delta &=& \; |P_2| + |P_3| \\ \Rightarrow\; & \delta &=& \; [\mathcal{D}(t_i, t_i) + \delta + \mathcal{D}(t_j, t_k)] + \mathcal{D}(t_k, t_j) \\ \Rightarrow\; & \delta &=& \; [0 + \delta + \mathcal{D}(t_j, t_k)] + \mathcal{D}(t_k, t_j) \\ \Rightarrow\; & 0 &=& \; \mathcal{D}(t_j, t_k) + \mathcal{D}(t_k, t_j) \end{aligned}$$

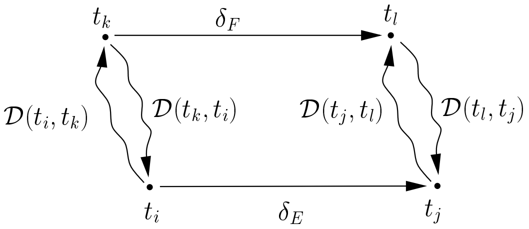
contradicting that $\mathcal{S}$ has no rigid components (cf. Definition 4.36). The case of E lying along P_3 results in a similar contradiction. Thus, the assumption that there exists some t_k such that $\mathcal{D}(t_i, t_k) + \mathcal{D}(t_k, t_j) = \delta$ must have been wrong.

On the other hand, if E is a tight, dominated edge and $\mathcal{S}$ has no rigid components, then removing E must not cause a change to the distance matrix of $\mathcal{S}$. Thus, there must exist some shortest path P from t_i to t_j such that $|P| = \delta$. Since the edge from t_i to t_j has been removed, P must contain some other point t_k. Thus, $\mathcal{D}(t_i, t_k) + \mathcal{D}(t_k, t_j) = \delta$. ∎

Theorem 4.46 *Suppose E is a dominated edge in a consistent STN $\mathcal{S}$ in which there are no rigid components. Then removing any set $\{E_1, E_2, \ldots, E_n\}$ of dominated edges from $\mathcal{S}$ will not change the property of E being a dominated edge in $\mathcal{S}$. Thus, the set of* undominated constraints *in $\mathcal{S}$ is a well-defined notion.*

Proof Let $E = (t_i, \delta_E, t_j)$ be a dominated edge in an STN $\mathcal{S}$ that has no rigid components. Suppose $F = (t_k, \delta_F, t_l)$ is some *other* dominated edge in $\mathcal{S}$, and that removing F would cause E to become undominated in $\mathcal{S}$. Since E and F are both dominated in $\mathcal{S}$, removing either edge alone would not cause a change in the distance matrix of $\mathcal{S}$. However, by assumption, removing F would cause E to become undominated. Thus, removing *both* E and F *would* cause a change to the distance matrix. Thus, removing only E would cause F to become undominated as well.

As a result, it must be that F is part of every shortest path from t_i to t_j that does not include E. Similarly, E is part of every shortest path from t_k to t_l that does not include F. This situation is pictured below.



Considering the paths (t_i, t_k, t_l, t_j) and (t_k, t_i, t_j, t_l), which are shortest paths, we get the following equations:

$$\begin{aligned} \mathcal{D}(t_i, t_k) + \delta_F + \mathcal{D}(t_l, t_j) &= \delta_E \\ \mathcal{D}(t_k, t_i) + \delta_E + \mathcal{D}(t_j, t_l) &= \delta_F. \end{aligned}$$

Adding these together and canceling like quantities yields:

$$[\mathcal{D}(t_i, t_k) + \mathcal{D}(t_k, t_i)] \; + \; [\mathcal{D}(t_l, t_j) + \mathcal{D}(t_j, t_l)] \; = \; 0.$$

Since both of the bracketed quantities are necessarily non-negative in a consistent STN, we get that both bracketed quantities must in fact be equal to 0, contradicting that $\mathcal{S}$ has no rigid components.

Finally, suppose that removing the finite *sequence* of dominated edges $E_1, E_2, \ldots, E_n$ caused the status of some remaining edge E to change from dominated to undominated. Let E_i be the *first* edge in that sequence such that E was dominated before E_i's removal, but undominated thereafter. The preceding argument shows that this cannot happen. ■

Definition 4.47 *The* undominated constraints *in an STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ are denoted $\mathcal{C}^u$.*

By Theorem 4.45, the following simple $O(|\mathcal{C}||\mathcal{T}|)$ algorithm will find the undominated constraints in an STN that has no rigid components: For each tight edge, $t_j - t_i \leq \delta$, check whether there is some t_k such that $\mathcal{D}(t_i, t_k) + \mathcal{D}(t_k, t_j) = \delta$. If there is no such t_k, then the edge is undominated.

Theorem 4.48 *Let $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ be an STN in which there are no rigid components; let $\mathcal{S}_d = (\mathcal{T}, \mathcal{C}_d)$ be its d-STN. Then the set $(\mathcal{C}_d)^u$ of undominated constraints in $\mathcal{C}_d$ is the unique, minimal set of constraints equivalent to the constraints in $\mathcal{C}$. (Here, "minimal" refers to the number of constraints in the set.)*

Proof *(Equivalence of $\mathcal{C}$ and $(\mathcal{C}_d)^u$)* By Proposition 4.29, $\mathcal{S}$ and $\mathcal{S}_d$ are equivalent (i.e., $\mathcal{S} \equiv \mathcal{S}_d$). Thus, $\mathcal{C} \equiv \mathcal{C}_d$. Furthermore, since $\mathcal{S}$ has no rigid components, neither does

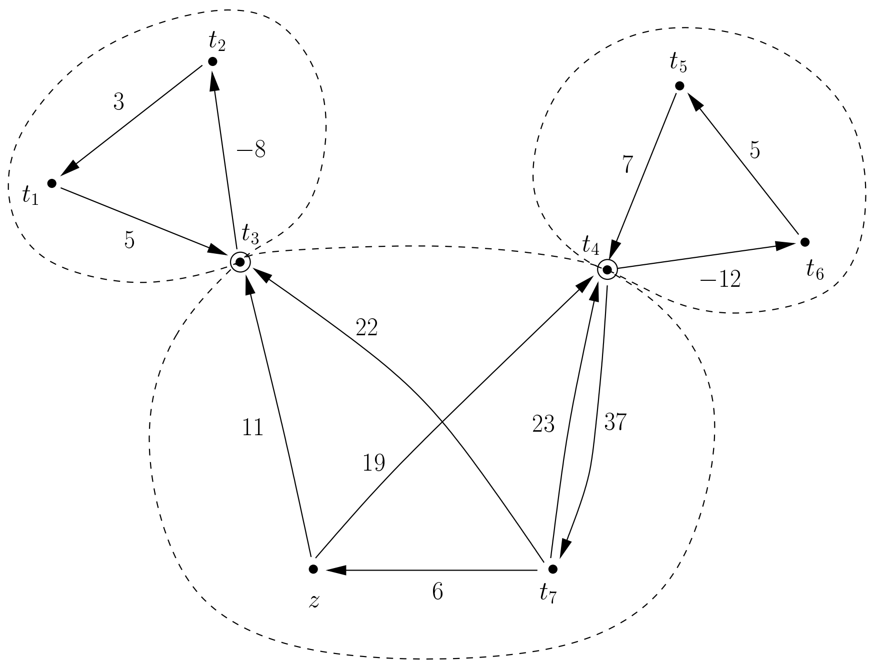


Figure 4.4: The STN from Figure 4.2 with its decoupled rigid components

$\mathcal{S}_d$. Thus, by Theorem 4.46, the set $(\mathcal{C}_d)^u$ of undominated constraints in $\mathcal{C}_d$ may be constructed by removing the dominated constraints from $\mathcal{C}_d$. Since removing dominated constraints does not affect the distance matrix (Definition 4.44), the distance matrices for $\mathcal{S}_d$ and $(\mathcal{T}, (\mathcal{C}_d)^u)$ are identical. Hence, by Corollary 4.28, $\mathcal{C}_d \equiv (\mathcal{C}_d)^u$. Thus, $(\mathcal{C}_d)^u \equiv \mathcal{C}$.

(Minimality of $(\mathcal{C}_d)^u$*)* Since removing any undominated constraint from $(\mathcal{C}_d)^u$ *would* change the distance matrix (Definition 4.44), and hence would cause the resulting constraint set to lose its equivalence with $\mathcal{C}$, the set $(\mathcal{C}_d)^u$ is a minimal constraint set equivalent to $\mathcal{C}$.

(Uniqueness) By Theorem 4.45, each edge $E = (t_j - t_i \leq \delta)$ in $(\mathcal{C}_d)^u$ constitutes the *only* shortest path from t_i to t_j in the fully-connected d-graph $\mathcal{G}_d$. Thus, replacing the edge E with some alternative shortest path from t_i to t_j would necessarily require changing the distance matrix. ∎

Theorem 4.48, above, specifies the minimum number of constraints needed to entail the constraints in a network having no rigid components. Theorem 4.53, below, provides the analogous result for networks that have rigid components. It shows that the minimum number of edges needed to entail a network having rigid components can be found by first decoupling the rigid components from the rest of the network, as described in Section 4.1, and then providing minimal sets of edges to represent each of the resulting subnetworks.

Figure 4.3 from Section 4.1 shows an STN after its rigid components have been collapsed down to single points. Figure 4.4 shows the same network, but with the rigid components shown as decoupled subnetworks, henceforth called *rc-subnetworks*. Notice that rc-subnetworks, unlike proper STNs (cf. Definition 4.1), need not include the zero time-point variable, z. Notice also that the edges in each rc-subnetwork have been arranged into a loop, which the following result shows to be the most concise representation for a rigid component.

Lemma 4.49 *If all of the time-points in a consistent STN $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ are rigidly connected and $|\mathcal{T}| > 1$, then $|\mathcal{C}| \geq |\mathcal{T}|$. Furthermore, there exists an equivalent STN $\mathcal{S}' = (\mathcal{T}, \mathcal{C}')$ such that $|\mathcal{C}'| = |\mathcal{T}|$.*

Proof Let $t_i \in \mathcal{T}$ be arbitrary. Since $|\mathcal{T}| > 1$, there exists some $t_j \in \mathcal{T}$ such that $t_i \neq t_j$. Since t_i and t_j are rigidly connected in $\mathcal{S}$, there is a shortest path in $\mathcal{S}$ from t_i to t_j. Thus, there is at least one edge leaving t_i. Since each edge originates at a single point and each time-point has at least one edge emanating from it, there must be at least $|\mathcal{T}|$ edges in $\mathcal{C}$.

Now suppose $\mathcal{T} = \{t_0, t_1, \ldots, t_n\}$. Let $\mathcal{C}'$ be the following set of constraints:

$$\{\ (t_1 - t_0 \leq \mathcal{D}(t_0, t_1)),\ \ldots,\ (t_n - t_{n-1} \leq \mathcal{D}(t_{n-1}, t_n)),\ (t_0 - t_n \leq \mathcal{D}(t_n, t_0))\ \},$$

where $\mathcal{D}$ is the distance matrix for $\mathcal{S}$. Let $\mathcal{S}' = (\mathcal{T}, \mathcal{C}')$. Let $t_i, t_j \in \mathcal{T}$ be arbitrary such that $i < j$. The only path from t_i to t_j in $\mathcal{S}'$ that does not contain a loop is the path, $P_{ij} = (t_i, t_{i+1}, \ldots, t_{j-1}, t_j)$. Thus, P_{ij} must be a shortest path in $\mathcal{S}'$. Similarly, the only path from t_j to t_i that does not contain a loop is the path, $P_{ji} = (t_j, t_{j+1}, \ldots, t_0, \ldots, t_{i-1}, t_i)$, which must therefore also be a shortest path. Next, notice that

$$|P_{ij}| = \mathcal{D}(t_i, t_{i+1}) + \ldots + \mathcal{D}(t_{j-1}, t_j),$$

which, by repeated application of the Triangle Inequality yields that $|P_{ij}| \geq \mathcal{D}(t_i, t_j)$. Similarly, $|P_{ji}| \geq \mathcal{D}(t_j, t_i)$.

Next, notice that

$$|P_{ij}| + |P_{ji}| = \mathcal{D}(t_0, t_1) + \mathcal{D}(t_1, t_2) + \ldots + \mathcal{D}(t_{n-1}, t_n) + \mathcal{D}(t_n, t_0).$$

Since each pair of time-points, (t_p, t_q), is rigidly connected in $\mathcal{S}$, it follows that $\mathcal{D}(t_p, t_q) = -\mathcal{D}(t_q, t_p)$ (cf. Definition 4.36). Thus,

$$\begin{aligned} -|P_{ij}| - |P_{ji}| &= -\mathcal{D}(t_0, t_1) - \mathcal{D}(t_1, t_2) - \ldots - \mathcal{D}(t_{n-1}, t_n) - \mathcal{D}(t_n, t_0) \\ &= \mathcal{D}(t_1, t_0) + \mathcal{D}(t_2, t_1) + \ldots + \mathcal{D}(t_n, t_{n-1}) + \mathcal{D}(t_0, t_n). \end{aligned}$$

Since the right-hand side of each of the above equations represents the length of a loop in $\mathcal{S}$, which is a consistent STN, each must be non-negative (cf. Theorem 4.30). Thus, $|P_{ij}| + |P_{ji}|$ must be equal to zero. Thus,

$$\begin{array}{ll} |P_{ij}| = -|P_{ji}| & \\ \Rightarrow\ |P_{ij}| \leq -\mathcal{D}(t_j, t_i), & \text{Since } |P_{ji}| \geq \mathcal{D}(t_j, t_i) \\ \Rightarrow\ |P_{ij}| \leq \mathcal{D}(t_i, t_j), & \text{Since } \mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, t_i) = 0 \\ \Rightarrow\ |P_{ij}| = \mathcal{D}(t_i, t_j), & \text{Since } \mathcal{D}(t_i, t_j) \leq |P_{ij}|. \end{array}$$

Similarly, $|P_{ji}| = \mathcal{D}(t_j, t_i)$. Since P_{ij} is a shortest path in $\mathcal{S}'$, $\mathcal{D}'(t_i, t_j) = |P_{ij}|$; hence $\mathcal{D}'(t_i, t_j) = \mathcal{D}(t_j, t_i)$. Similarly, $\mathcal{D}'(t_j, t_i) = \mathcal{D}(t_j, t_i)$. Since t_i and t_j were arbitrary in $\mathcal{T}$ with $i < j$, $\mathcal{S}' \equiv \mathcal{S}$. Since the number of edges in $\mathcal{C}'$ is equal to $|\mathcal{T}|$, the result is proven. ■

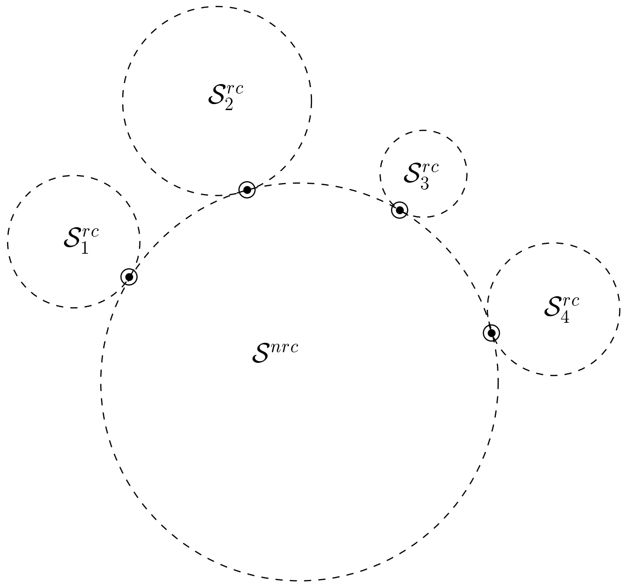


Figure 4.5: An STN with four decoupled rigid components

Definition 4.50 (rc-Subnetworks) *Let $\mathcal{S}$ be a consistent STN. Let $\mathcal{S}^*$ be* any *STN resulting from decoupling the rigid components of $\mathcal{S}$ as described in Section 4.1. Let*

$$\mathcal{S}_1^{rc} = (\mathcal{T}_1^{rc}, \mathcal{C}_1^{rc}), \quad \ldots, \quad \mathcal{S}_n^{rc} = (\mathcal{T}_n^{rc}, \mathcal{C}_n^{rc})$$

be the subnetworks in $\mathcal{S}^*$ corresponding to the decoupled rigid components; and let $\mathcal{S}^{nrc} = (\mathcal{T}^{nrc}, \mathcal{C}^{nrc})$ be the remaining subnetwork of $\mathcal{S}$ from which the rigid components have been decoupled. The subnetworks, $\mathcal{S}_1^{rc} = (\mathcal{T}_1^{rc}, \mathcal{C}_1^{rc}), \ldots, \mathcal{S}_n^{rc} = (\mathcal{T}_n^{rc}, \mathcal{C}_n^{rc})$, are called rc-subnetworks *of $\mathcal{S}^*$; $\mathcal{S}^{nrc}$ is called the* nrc-subnetwork *of $\mathcal{S}^*$.*

Figure 4.5 illustrates the notation from Definition 4.50. Notice that the nrc-subnetwork, $\mathcal{S}^{nrc}$, contains no rigid components, but that for each i, $\mathcal{T}^{nrc} \cap \mathcal{T}_i^{rc}$ is a singleton set containing the representative time-point for the rc-subnetwork $\mathcal{S}_i^{rc}$.

Proposition 4.51 *If P is a loop with length zero in a consistent STN $\mathcal{S}$, then the set of time-points occurring in P forms a rigid component in $\mathcal{S}$.*

Proof Let t_i and t_j be arbitrary time-points in P. Let P_{ij} be the subpath of P from t_i to t_j. Let P_{ji} be the subpath in P from t_j to t_i. (If t_i or t_j occur more than once in P, choose P_{ij} and P_{ji} such that they concatenate to form P.) Since replacing P_{ij} in P with a strictly shorter path would create a loop in $\mathcal{S}$ with negative path-length, which would contradict that $\mathcal{S}$ is consistent, P_{ij} must be a shortest path in $\mathcal{S}$. Similarly, P_{ji} must be a shortest path in $\mathcal{S}$. Thus,

$$0 \;=\; |P_{ij}| + |P_{ji}| \;=\; \mathcal{D}(t_i, t_j) + \mathcal{D}(t_j, t_i).$$

Hence, t_i and t_j are rigidly connected. Since t_i and t_j were arbitrary time-points in P, the result is proven. ∎

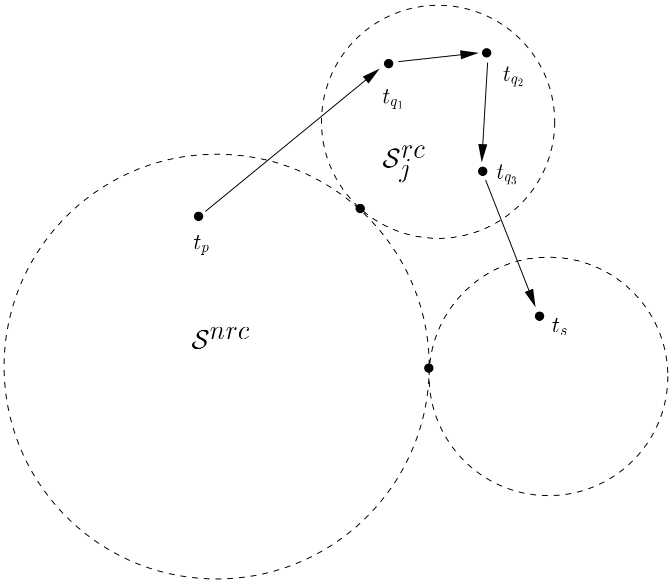


Figure 4.6: A sample path P'' from the proof of Theorem 4.53

Corollary 4.52 *If P is a shortest path from t_i to t_j in a consistent STN, where t_i and t_j are rigidly connected, then* all *of the time-points in P belong to the rigid component containing t_i and t_j.*

Proof Since P is a shortest path from t_i to t_j, $|P| = \mathcal{D}(t_i, t_j)$. Since t_i and t_j are rigidly connected, there is some shortest path P' from t_j to t_i such that $|P'| = \mathcal{D}(t_j, t_i) = -\mathcal{D}(t_i, t_j)$. But then concatenating P and P' forms a loop with path-length zero. Thus, by Proposition 4.51, all of the time-points in P, including t_i and t_j, belong to the same rigid component. ■

Theorem 4.53 *Let $\mathcal{S} = (\mathcal{T}, \mathcal{C})$ be a consistent STN. Let $\mathcal{S}_1^{rc}, \ldots, \mathcal{S}_n^{rc}$ and $\mathcal{S}^{nrc}$ be the rc-subnetworks of $\mathcal{S}$ (cf. Definition 4.50) such that each rc-subnetwork $\mathcal{S}_i^{rc}$ has the minimum number of edges (cf. Lemma 4.49). Let $\mathcal{S}' = (\mathcal{T}, \mathcal{C}')$ be an arbitrary STN that is equivalent to $\mathcal{S}$. Then*

$$|\mathcal{C}'| \;\geq\; |\mathcal{T}_1^{rc}| + \ldots + |\mathcal{T}_n^{rc}| + |(\mathcal{C}^{nrc})_d^u|.$$

Proof First, consider an arbitrary rigid component $\mathcal{T}_i^{rc}$ in $\mathcal{S}$. Let P be an arbitrary shortest path in $\mathcal{S}'$ both of whose endpoints are in $\mathcal{T}_i^{rc}$. By Corollary 4.52, all of the time-points in P are necessarily in $\mathcal{T}_i^{rc}$. As a result, the edges in $\mathcal{C}'$ that entail the constraints in the rigid components $\mathcal{T}_1^{rc}, \mathcal{T}_2^{rc}, \ldots, \mathcal{T}_n^{rc}$, necessarily fall into *disjoint sets*.

Next, let P' be a shortest path in $\mathcal{S}'$ both of whose endpoints are in $\mathcal{S}^{nrc}$. Suppose P' contains a subpath, $P'' = (t_p, t_{q_1}, t_{q_2}, \ldots, t_{q_k}, t_s)$, where $t_p \in \mathcal{T}^{nrc}$, $k \geq 1$, $t_{q_1}, t_{q_2}, \ldots, t_{q_k} \in \mathcal{T}_j^{rc}$, for some j, and $t_s \notin \mathcal{T}_j^{rc}$. Figure 4.6 illustrates such a path in the case where $k = 3$ and t_s happens to lie in some other rc-subnetwork. Let E_1 and E_2 be the first and last edges in P''. Then, E_1 and E_2 have the following form:

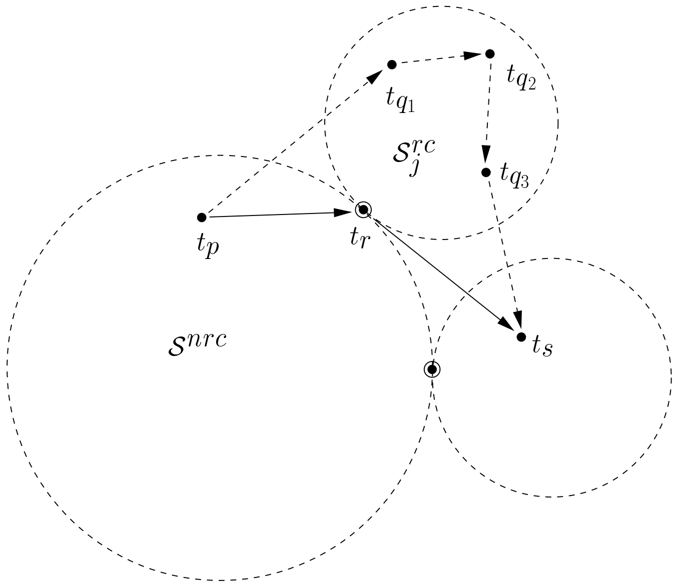


Figure 4.7: Modifying the constraint set $\mathcal{C}'$ in the proof of Theorem 4.53

$$E_1\colon t_{q_1} - t_p \leq \delta_1 \quad \text{and} \quad E_2\colon t_s - t_{q_k} \leq \delta_2.$$

If t_{q_1} is not the representative t_r of the rigid component, $\mathcal{T}^{rc}_j$, then E_1 may be replaced by the edge,

$$E_1'\colon t_r - t_p \leq (\delta_1 + \mathcal{D}(t_q, t_r)),$$

as described in Section 4.1. Similarly, if t_{q_k} is not the representative of the component, then E_2 may be replaced in $\mathcal{C}'$ by the edge,

$$E_2'\colon t_s - t_r \leq (\delta_2 - \mathcal{D}(t_q, t_r)).$$

Modifying the contents of the set $\mathcal{C}'$ in this way, as illustrated in Figure 4.7, does not change the set of solutions to the network; nor does it change the number of edges in $\mathcal{C}'$. However, the subpath P'' may be replaced (in P') by the subpath (t_p, t_r, t_s), without affecting the length of P'. However, t_p and t_r are time-points in $\mathcal{T}^{nrc}$. Furthermore, if t_s is not in $\mathcal{T}^{nrc}$, then the above argument may be applied to the subpath beginning at t_r. Thus, by repeated applications of the above argument, the original path P' may be converted to an equivalent path.

Similar arguments apply to any shortest path from a time-point in $\mathcal{T}^{nrc}$ to a non-representative time-point in some rigid component, and to any shortest path joining non-representative time-points from two different rigid components. In each case, any edge that joins a non-representative time-point from some rigid component to a time-point lying outside that rigid component is replaced by an edge involving the representative of that rigid component, as described in Section 4.1. Since doing so does not affect the intra-component edges in $\mathcal{C}'$, does not affect the set of solutions to the network, and does not change the number of edges in $\mathcal{C}'$, the net result is that the edges in the modified version of $\mathcal{C}'$ fall into disjoint sets: one set for each rigid component and one set for the time-points in $\mathcal{T}^{nrc}$. Since the minimum number of edges required to entail the constraints in each rigid component $\mathcal{T}_i^{rc}$ is $|\mathcal{T}_i^{rc}|$ (by Lemma 4.49), and since the minimum number of edges required to entail the constraints in $\mathcal{S}^{nrc}$ is $|(\mathcal{C}^{nrc})_d^u|$ (by Theorem 4.48), the result is proven. ■