On Redundancy in Simple Temporal Networks[1]

Jae Hee Lee[2] and **Sanjiang Li**[3] and **Zhiguo Long**[4] and **Michael Sioutis**[5]

Abstract. The Simple Temporal Problem (STP) has been widely used in various applications to schedule tasks. For dynamical systems, scheduling needs to be efficient and flexible to handle uncertainty and perturbation. To this end, modern approaches usually encode the temporal information as an STP instance. This representation contains redundant information, which can not only take a significant amount of storage space, but also make scheduling inefficient due to the non-concise representation. In this paper, we investigate the problem of simplifying an STP instance by removing redundant information. We show that such a simplification can result in a unique minimal representation without loss of temporal information, and present an efficient algorithm to achieve this task. Evaluation on a large benchmark dataset of STP exhibits a significant reduction in redundant information for the involved instances.

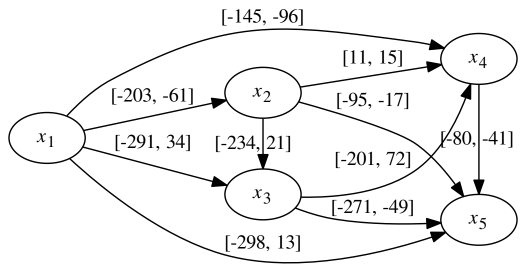
1 Introduction

The ability to reason about temporal information is necessary for intelligent agents that plan their actions to achieve their goals optimally. As such, temporal reasoning has been an active research area in Artificial Intelligence [6, 2, 16, 3, 17].

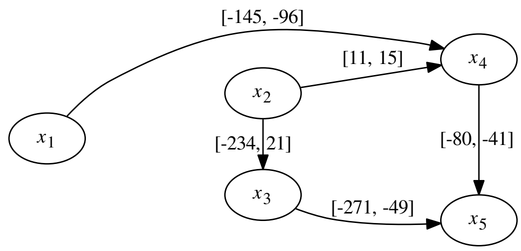
Among the different temporal representation frameworks, some of the most well-known ones are based on relations between time points or time intervals. Prominent examples of such representation frameworks include for example the Simple Temporal Problem [5], Allen's Interval Algebra [1], and Point Algebra [20]. In this paper, we will focus on the Simple Temporal Problem and are concerned with redundant information in instances of this problem.

The *Simple Temporal Problem* (STP) encodes the quantitative difference between two variables representing time points. An STP constraint $(x\ [a, b]\ y)$ associates an interval $[a, b]$ with variables x and y to represent the lower and upper bounds of the difference, i.e., $a \leq y - x \leq b$. An STP instance is called Simple Temporal Network (STN), and consists of a set of variables and a set of STP constraints involving those variables. This can be represented as a labelled directed graph (see Figure 1). A solution of an STN is an assignment of time points to the variables such that all of the corresponding STP constraints are satisfied by the assignment.

There are redundant constraints in STNs. Figure 1a shows an STN extracted from the job-shop scheduling problem benchmark dataset used in [14]. (In all of our examples, we note that we have modified the original STNs by reducing the bounds in their constraints to make the examples easier to follow; however, all qualitative properties of the STNs have been retained). We observe that the STN is densely



(a) An STN extracted from a job-shop problem instance.



(b) An STN obtained by removing all redundant constraints in the STN from Figure 1a. The two STNs are equivalent by Theorem 7.

Figure 1. Two equivalent STNs.

structured and contains redundant constraints that can be removed without affecting the set of solutions. For example, the constraint $(x_2\ [-95, -17]\ x_5)$ is redundant and can be removed, because combining the constraint between x_2 and x_4 and the constraint between x_4 and x_5 implies a tighter constraint, viz., $(x_2\ [-69, -26]\ x_5)$. Redundant constraints are not limited to the aforementioned trivial case; although the constraint between x_3 and x_4 cannot be directly inferred from any other two constraints, it can be inferred by combining more constraints (cf. Example 6 in Section 3). Identifying such redundant constraints efficiently is one of the main goals of this paper.

After identifying all redundant constraints in an STN, the question arises whether we can remove all of them, while maintaining the solution set unchanged. In this paper we characterize STNs that retain the solution sets after removing all redundant constraints. As an example, the STN in Figure 1a shares the same solution set with

[1] The authors have contributed equally.
[2] QCIS, FEIT, UTS, Australia, email: JaeHee.Lee@uts.edu.au
[3] QCIS, FEIT, UTS, Australia, email: Sanjiang.Li@uts.edu.au
[4] QCIS, FEIT, UTS, Australia, email: Zhiguo.Long@student.uts.edu.au
[5] CRIL, University of Artois, France, email: sioutis@cril.fr

the STN in Figure 1b that results from removing all redundant constraints of the STN in Figure 1a.

In the literature, the problem of identifying and removing redundant information has been discussed in the context of logic formulas [8, 10] as well as of qualitative spatial and temporal networks. For qualitative spatial and temporal networks, it was shown in [9] that an all-different network defined over a distributive subclass of relations in RCC8 [15] or Point Algebra [20] has a uniquely defined subset of non-redundant constraints that defines a network that is equivalent to the original network. Efficient algorithms have been developed to identify such subsets [9, 18].

For STP constraints, [11] and [4] identify another notion of redundancy, so-called *dominance*, for inferring the range of a variable. By identifying and removing such redundancy, a particular property of an STN, viz., the *dispatchability* [11], which is helpful for generating solutions of an STN online, may be retained. In this paper, on the other hand, we will show that, by removing redundant constraints, the structure of an STN is simplified, while the solution set remains the same. This could make other reasoning tasks involving STNs more efficient, e.g., speed up the generation of solutions of dynamic STNs.

The remainder of the paper is organised as follows. In Section 2, we recall some basic concepts and formally define redundant constraints. We then prove in Section 3 some properties of redundant constraints in STNs and use those properties to devise an algorithm for identifying and removing redundant constraints (Section 4). Section 5 evaluates our approach by using benchmark datasets. Finally, Section 6 concludes the paper.

2 Preliminaries

Definition 1. The *simple temporal problem* (STP) is a constraint satisfaction problem where each constraint is a set of linear inequalities of the form

$$a \leq y - x \leq b, \tag{1}$$

where a, b are constants and x, y are variables defined on a continuous domain representing time points. The constraint in (1) is abbreviated as $(x\ [a,b]\ y)$. As (1) is equivalent to $-b \leq x - y \leq -a$, we identify $(x\ [a,b]\ y)$ with $(y\ [-b,-a]\ x)$.

Algebraic operations on STP constraints are defined as follows. The *intersection* of two STP constraints defined over variables x, y yields a new constraint over x, y that represents the conjunction of the constraints. It is defined as $(x\ [a_1,b_1]\ y) \cap (x\ [a_2,b_2]\ y) := (x\ [c,d]\ y)$, where $[c,d] = [a_1,b_1] \cap [a_2,b_2]$.

The *composition* of an STP constraint over variables x, z and another STP constraint z, y yields a new STP constraint over x, y that is inferred from the other two constraints and is defined as $(x\ [a_1,b_1]\ z) \otimes (z\ [a_2,b_2]\ y) := (x\ [c,d]\ y)$, where interval $[c,d]$ is obtained by adding the corresponding bounds of the other two constraints, i.e., $c = a_1 + a_2$ and $d = b_1 + b_2$.

Note that for STP constraints, the composition and intersection are associative and, as noted in [5], composition distributes over non-empty intersection, i.e., $(x\ [a,b]\ y) \otimes \big((x\ [c,d]\ y) \cap (x\ [e,f]\ y)\big) = \big((x\ [a,b]\ y) \otimes (x\ [c,d]\ y)\big) \cap \big((x\ [a,b]\ y) \otimes (x\ [e,f]\ y)\big)$.

Definition 2. An instance of STP is called a *simple temporal network* (STN) and is a tuple (V, Γ), where $V = \{x_1, \ldots, x_n\}$ is a set of variables and Γ is a set of STP constraints defined over V.

Without loss of generality, we assume that all variables in V appear in Γ. Thus we will use Γ to refer to an STN and not explicitly mention V. Furthermore, we assume without loss of generality that there is only one constraint between x and y.

An STN naturally induces a graph in the following sense.

Definition 3. The *constraint graph* $G = (V, E)$ of an STN Γ is an undirected graph, where the set V of vertices consists of the variables in Γ and the set E of edges consists of unordered pairs of variables for which there is an STP constraint in Γ, i.e.,

$$E = \{\{x,y\} \mid x, y \in V, x \neq y, (x\ [a,b]\ y) \in \Gamma\}.$$

For simplicity, we write xy for an edge in place of $\{x,y\}$.

Let $G = (V, E)$ be the constraint graph of an STN Γ. For variables x, y with $xy \in E$, we write $\Gamma(x,y)$ (and equivalently $\Gamma(y,x)$) to refer to the constraint between x and y in Γ. Figure 1a is an illustration of the constraint graph of an STN. Note that we use a *directed* graph to illustrate a constraint graph, where for any edge $xy \in E$ with $\Gamma(x,y) = (x\ [a,b]\ y)$ there is exactly one directed edge (x,y) in the illustration with the corresponding interval $[a,b]$.

Definition 4. A *solution* of an STN Γ is an assignment of time points to the variables in Γ such that all constraints in Γ are satisfied. Given two STNs Γ and Γ' defined over the same set of variables we write $\Gamma' \models \Gamma$, if every solution of Γ' is a solution of Γ. If $\Gamma' \models \Gamma$ and $\Gamma \models \Gamma'$ then we say that they are equivalent and write $\Gamma' \equiv \Gamma$. We also write $\Gamma \models (x\ [a,b]\ y)$ if every solution of Γ satisfies $(x\ [a,b]\ y)$.

We observe that $\{(x\ [a,b]\ y)\} \models (x\ [c,d]\ y)$ if and only if $[a,b] \subseteq [c,d]$. Therefore, if $\{(x\ [a,b]\ y)\} \models (x\ [c,d]\ y)$, we will say that $(x\ [a,b]\ y)$ *refines* $(x\ [c,d]\ y)$, written as $(x\ [a,b]\ y) \subseteq (x\ [c,d]\ y)$. We call an STN Γ' a *refinement* of Γ, if Γ' and Γ are defined over the same set of variables and if for any constraint $(x\ [a,b]\ y) \in \Gamma$ there exists a constraint $(x\ [a',b']\ y) \in \Gamma'$ that refines $(x\ [a,b]\ y)$.

Definition 5 (Minimality)**.** Let Γ be a consistent STN and let x and y variables in Γ. Then an STP constraint $(x\ [a,b]\ y)$ is said to be *minimal* in Γ, if $\Gamma \models (x\ [a,b]\ y)$ and $(x\ [a,b]\ y)$ is the smallest constraint with respect to $\subseteq$.

A refinement Γ^{m} of Γ is said to be *the minimal network of* Γ, if for all $x, y \in V$, $x \neq y$ there is a constraint between x and y in Γ^{m} that is minimal in Γ.

Note that Γ^{m} is uniquely defined, and since $\Gamma^{\mathrm{m}} \models \Gamma$ and $\Gamma \models \Gamma^{\mathrm{m}}$, we have $\Gamma^{\mathrm{m}} \equiv \Gamma$. Figure 2a shows the minimal network of the STN Γ in Figure 1a. Note that it is also the minimal network of the STN Γ^{c} in Figure 1b.

Definition 6. Let $G = (V, E)$ be the constraint graph of an STN Γ. A sequence $P := (x_{i_0}x_{i_1}, x_{i_1}x_{i_2}, \ldots, x_{i_{k-1}}x_{i_k})$ of edges in E is called a *path* from x_{i_0} to x_{i_k}. The length of P is the number of edges in P and is denoted by $|P|$. A path P is called a *cycle*, if $x_{i_k} = x_{i_0}$. By $\Gamma(P)$ we denote the set $\{\Gamma(x_{i_0}, x_{i_1}), \Gamma(x_{i_1}, x_{i_2}), \ldots, \Gamma(x_{i_{k-1}}, x_{i_k})\}$ of constraints over P. The successive composition of constraints over P with respect to Γ is defined as

$$\bigotimes \Gamma(P) := \Gamma(x_{i_0}, x_{i_1}) \otimes \Gamma(x_{i_1}, x_{i_2}) \otimes \cdots \otimes \Gamma(x_{i_{k-1}}, x_{i_k}).$$

A concatenation of two paths $P_1 := (x_{i_0}x_{i_1}, \ldots, x_{i_{k-1}}x_{i_k})$ and $P_2 := (x_{i_k}x_{i_{k+1}}, \ldots, x_{i_{\ell-1}}x_{i_\ell})$, denoted by $P_1 + P_2$, is the path $(x_{i_0}x_{i_1}, \ldots, x_{i_{\ell-1}}x_{i_\ell})$. We write $\Pi(x, y, E)$ for the set of all paths from x to y on set E of edges.

Example 1. In Figure 1b $P = (x_1x_4, x_4x_2, x_2x_3)$ is a path of length three in $\Pi(x_1, x_3, E)$, and $C = (x_2x_4, x_4x_5, x_5x_3, x_3x_2)$ is a cycle in $\Pi(x_2, x_2, E)$. The composition of the constraints over P, $\bigotimes\Gamma(P)$, is then $(x_1[-145, -96]x_4) \otimes (x_4[-15, -11]x_2) \otimes (x_2[-234, 21]x_3)$. Note that $(x_4[-15, -11]x_2)$ is equivalent to $(x_2[11, 15]x_4)$.

The following lemma states that algebraic operations $\otimes$ and $\cap$ are sufficient to calculate the minimal network of an STN. More precisely, the minimal constraint between two variables x and y is equal to the intersection of all constraints between those variables obtained by composing the constraints over the paths in $\Pi(x, y, E)$.

Lemma 1. *Let Γ be an STN and E be the set of edges of the constraint graph of Γ. Let Γ^{m} be the minimal network of Γ. Then for all $x, y \in V, x \neq y$ we have*

$$\Gamma^{\mathrm{m}}(x, y) \;=\; \bigcap_{P \in \Pi(x,y,E)} \bigotimes \Gamma(P).$$

Proof. See [5, Section 3]. $\square$

For constraint satisfaction problems chordal (aka triangulated) constraint graphs have been identified as a class for which efficient algorithms exist [18]. In the following we provide a characterization of a chordal graph equivalent to that in [18], since it is more suited for the purpose of the paper.

Definition 7. (Chordal Graph) A graph $G = (V, E)$ is said to be *chordal* or *triangulated*, if for any edge $xy \in E$ and any path P from x to y on E with $|P| \geq 2$ there exist a $z \in V$ with $xz, zy \in E$ such that $P = P_1 + P_2$, where P_1 is a path from x to z and P_2 a path from z to y.

Example 2. A complete graph is chordal. The graph in Figure 1a is chordal as it is complete. We can make a non-chordal graph chordal by triangulating it. For example, the graph in Figure 1b is not chordal, but we can make it chordal by adding edge x_3x_4 and obtain the graph in Figure 2b.

In the following lemma we characterize a refinement (of an STN) whose constraint graph is chordal. Such a refinement has a constraint network that is less dense than that of the minimal network while sharing some nice properties with the minimal network.

Lemma 2. *Let Γ be a consistent STN and $G = (V, E)$ be its constraint graph. Further let $\Gamma^{\triangle}$ be a refinement of Γ such that*

1. *its constraint graph $G^{\triangle} = (V, E^{\triangle})$ is chordal with $E^{\triangle} \supseteq E$;*
2. *$\Gamma^{\triangle}(x, y) = \Gamma^{\mathrm{m}}(x, y)$ for all $xy \in E^{\triangle}$.*

Then $\Gamma^{\triangle}$ is equivalent to Γ.

Proof. We note that $\Gamma^{\mathrm{m}} \models \Gamma^{\triangle}$ and $\Gamma^{\triangle} \models \Gamma$. Then, because $\Gamma^{\mathrm{m}} \equiv \Gamma$, we have $\Gamma \models \Gamma^{\triangle}$ and $\Gamma^{\triangle} \models \Gamma$. Therefore $\Gamma \equiv \Gamma^{\triangle}$. $\square$

Example 3. The minimal network Γ^{m} of Γ satisfies trivially the conditions for $\Gamma^{\triangle}$, as its constraint graph is complete and thus chordal. The STN in Figure 2b has a chordal constraint graph that contains the constraint graph of the STN from Figure 1b. Its constraints are minimal.

The following characterization of $\Gamma^{\triangle}$ follows from its definition and Lemma 1.

Lemma 3. *Let Γ and $\Gamma^{\triangle}$, as well as E and $E^{\triangle}$, be specified as in Lemma 2. Then for all $x, y \in V$ with $xy \in E^{\triangle}$ we have*

$$\Gamma^{\triangle}(x, y) \;=\; \bigcap_{P \in \Pi(x,y,E)} \bigotimes \Gamma(P).$$

3 Redundancy in Simple Temporal Networks

In an STN some constraints can be redundant as they can be inferred from the other constraints of the STN.

Definition 8. Let Γ be an STN. Then $\Gamma(x, y)$ is said to be *redundant* in Γ, if $\Gamma \setminus \{\Gamma(x, y)\} \models \Gamma(x, y)$. We say Γ is *prime*, if it does not contain any redundant constraints. A subset Γ' of Γ is called a *prime subnetwork* of Γ, if Γ' is prime and equivalent to Γ. The set of non-redundant constraints in Γ, denoted by Γ^{c}, is the *core* of Γ.

Any prime subnetwork of Γ contains the core of Γ as a subset of constraints. If the core is equivalent to the original STN, then it would be the unique minimal subset of constraints that is equivalent to the original STN. However, the following simple example shows that it is not always the case.

Example 4. Suppose that the STN consists of the constraints $(x\ [0, 1]\ y)$, $(x\ [2, 3]\ z)$, and $(y\ [2, 2]\ z)$, which translate to $0 \leq y - x \leq 1$, $2 \leq z - x \leq 3$, and $z - y = 2$ respectively. Then, it is easy to see that $(x\ [0, 1]\ y)$ and $(x\ [2, 3]\ z)$ are both redundant because $(x\ [0, 1]\ y) = (x\ [2, 3]\ z) \otimes (z\ [-2, -2]\ y)$ and $(x\ [2, 3]\ z) = (x\ [0, 1]\ y) \otimes (y\ [2, 2]\ z)$. However, removing both $(x\,[0, 1]\,y)$ and $(x\,[2, 3]\,z)$ will leave the core as the single constraint $(y\ [2, 2]\ z)$, which obviously have a different solution set from the original STN.

On the other hand, for some other STNs the core is indeed equivalent to the original STN.

Example 5. The STN from Figure 1b is the core of the STN from Figure 1a. Moreover, these two STNs are equivalent, because the minimal network of both of them is the STN shown in Figure 2a.

We observe that in Example 4, the presence of constraint $(y[2, 2]z)$ makes the other two constraints dependent on each other with respect to their redundancy. These kind of STNs are *degenerated*.

Definition 9. Let Γ and $\Gamma^{\triangle}$, as well as E and $E^{\triangle}$, be specified as in Lemma 2. Γ is said to be *degenerated* if it is inconsistent or if there is $xy \in E^{\triangle}$ such that $\Gamma^{\triangle}(x, y) = (x\ [a, b]\ y)$ with $a = b$.

It is worth noting that if an STN is degenerated, say $(x\ [a, a]\ y)$, then we can easily fix it either by slightly adjusting the constraint with $(x\,[a - \varepsilon, a + \varepsilon]\,y)$, or by removing constraints involving y and updating the constraints $\Gamma(x, z)$ accordingly by $\Gamma(x, z) \cap \Gamma(x, y) \otimes \Gamma(y, z)$.

In the following, we will show that for non-degenerated STNs, the core will be the unique minimal subset of constraints that is equivalent to the original STN. To this end, we first characterize redundant constraints in an STN as the intersection of constraints over certain paths.

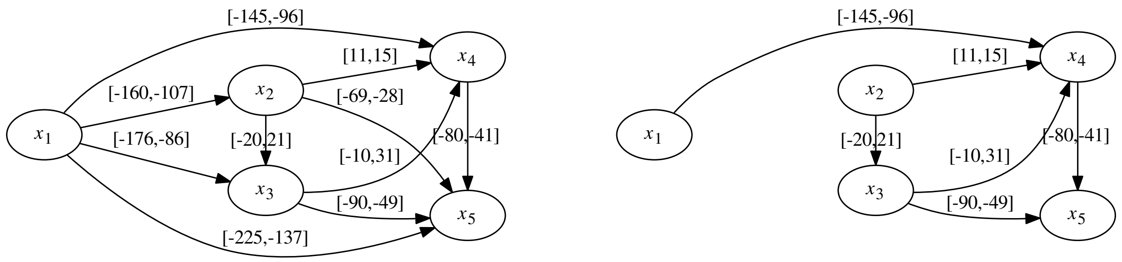
Lemma 4. *Let Γ be a consistent STN and $G = (V, E)$ its constraint graph. Then for all $x, y \in V$ with $xy \in E$ the following are equivalent:*

(i) $\Gamma(x, y)$ *is redundant in* Γ*;*

(ii) $\Gamma(x, y) \;\supseteq\; \bigcap_{P \in \Pi(x,y,E \setminus \{xy\})} \bigotimes \Gamma(P).$

Proof. (i) $\Rightarrow$ (ii). Let $\Gamma_0 := \Gamma \setminus \{\Gamma(x, y)\}$. Then, because $\Gamma(x, y)$ is redundant in Γ we have $\Gamma_0 \models \Gamma(x, y)$ thus $(\Gamma_0)^{\mathrm{m}} \models \Gamma(x, y)$, where $(\Gamma_0)^{\mathrm{m}}$ is the minimal network of Γ_0. Hence

$$\Gamma(x, y) \;\supseteq\; (\Gamma_0)^{\mathrm{m}}(x, y). \tag{2}$$



(a) The minimal network of the STNs from Figure 1a and from 1b.

(b) An STN whose constraint graph is chordal and contains the constraint graph of the STN from Figure 1b. It is equivalent to the STN from Figure 1b, and its constraints are minimal.

Figure 2. STNs equivalent to the STNs in Figure 1.

By Lemma 1 we know that

$$(\Gamma_0)^{\mathrm{m}}(x,y) = \bigcap_{P\in\Pi(x,y,E\setminus\{xy\})} \bigotimes \Gamma(P). \tag{3}$$

From (2) and (3) it follows that

$$\Gamma(x,y) \supseteq \bigcap_{P\in\Pi(x,y,E\setminus\{xy\})} \bigotimes \Gamma(P).$$

(ii) $\Rightarrow$ (i). For all paths between x and y on $E \setminus \{xy\}$, we have $\Gamma \models \Gamma(P)$ and thus $\Gamma \models \bigotimes\Gamma(P)$. Therefore

$$\Gamma \models \bigcap_{P\in\Pi(x,y,E\setminus\{xy\})} \bigotimes \Gamma(P).$$

By applying our assumption (ii) we then have $\Gamma \models \Gamma(x,y)$, and we showed (i). □

Example 6. The constraint $\Gamma(x_3,x_4) = (x_3[-201,72]x_4)$ in the STN Γ from Figure 1a is redundant in Γ, because $(x_3[-201,72]x_4) \supseteq (x_3[-10,31]x_4) = \Gamma(x_3,x_2)\otimes\Gamma(x_2,x_4)\cap\Gamma(x_3,x_5)\otimes\Gamma(x_5,x_4) \supseteq \bigcap_{P\in\Pi(x_3,x_4,E\setminus\{x_3y_4\})}\bigotimes\Gamma(P)$.

As noted before $\Gamma^{\triangle}$ shares some nice properties with the minimal network of Γ, thus allowing a characterization of the redundant constraints in $\Gamma^{\triangle}$.

Proposition 5. *Let Γ and $\Gamma^{\triangle}$, as well as E and $E^{\triangle}$, be specified as in Lemma 2. Then for all $x,y\in V$ with $xy\in E^{\triangle}$ the following are equivalent:*

(i) $\Gamma^{\triangle}(x,y)$ *is redundant in* $\Gamma^{\triangle}$
(ii) $\Gamma^{\triangle}(x,y) = \bigcap_{P\in\Pi(x,y,E^{\triangle}\setminus\{xy\})} \bigotimes \Gamma^{\triangle}(P)$
(iii) $\Gamma^{\triangle}(x,y) = \bigcap_{xz,zy\in E^{\triangle}} \Gamma^{\triangle}(x,z)\otimes\Gamma^{\triangle}(z,y)$

Proof. See Appendix A. □

Example 7. Consider the minimal network in Figure 2a as $\Gamma^{\triangle}$ for Γ in Figure 1a. Then we have $\Gamma^{\triangle}(x_3,x_4) = (x_3[-10,31]x_4) = (\Gamma^{\triangle}(x_3,x_1)\otimes\Gamma^{\triangle}(x_1,x_4)) \cap (\Gamma^{\triangle}(x_3,x_2)\otimes\Gamma^{\triangle}(x_2,x_4)) \cap (\Gamma^{\triangle}(x_3,x_5)\otimes\Gamma^{\triangle}(x_5,x_4))$.

There is a correspondence between redundant constraints in Γ and in $\Gamma^{\triangle}$.

Proposition 6. *Let Γ and $\Gamma^{\triangle}$, as well as E and $E^{\triangle}$, be specified as in Lemma 2. Furthermore, let Γ be not degenerated. Then for all $x,y\in V$ with $xy\in E^{\triangle}$ the following are equivalent:*

(i) $\Gamma(x,y)$ *is redundant in* Γ
(ii) $\Gamma^{\triangle}(x,y)$ *is redundant in* $\Gamma^{\triangle}$
(iii) $\Gamma^{\triangle}(x,y) = \bigcap_{\substack{P\in\Pi(x,y,E)\\ |P|\geq 2}} \bigotimes \Gamma(P)$

Proof. See Appendix A. □

Example 8. Taking the STN Γ in Figure 1a and the STN $\Gamma^{\triangle}$ in Figure 2a as an example of the above proposition, we can see the constraint $\Gamma(x_3,x_4) = (x_3[-201,72]x_4)$ is redundant in Γ while $\Gamma^{\triangle}(x_3,x_4) = (x_3[-10,31]x_4)$ is redundant in $\Gamma^{\triangle}$.

The following theorem affirms that, for non-degenerated STNs, the core resulting from removing redundant constraint simultaneously is the unique prime subnetwork of the STN.

Theorem 7. *Let Γ be a consistent and non-degenerated STN. Then the core Γ^{c} of Γ is equivalent to Γ and the unique prime subnetwork of Γ.*

Proof. The proof is similar to that for RCC8 constraints (see [9, Theorem 29]). Suppose $r_1, r_2, \ldots, r_n$ are the redundant constraints of Γ. Let $\Gamma_0 := \Gamma$ and $\Gamma_i := \Gamma_{i-1}\setminus\{r_i\}$ for $1\leq i\leq n$. Note that Γ_n is precisely Γ^{c}, the core of Γ.

Suppose k with $k<n$ is the largest integer such that $\Gamma_k\equiv\Gamma$. Let $r_{k+1} = \Gamma(x,y)$, thus $\Gamma(x,y)$ is redundant in Γ. Further, let $\Gamma^{\triangle}$ be as specified in Lemma 2. Then, on the one hand, we know by Proposition 6 that $\Gamma^{\triangle}(x,y)$ is redundant in $\Gamma^{\triangle}$. On the other hand, because $\Gamma^{\triangle}$ is a refinement of Γ_k that satisfies the conditions in Lemma 2 w.r.t. Γ_k, by Proposition 6 we also have that $\Gamma(x,y)$ is redundant in Γ_k. Therefore, $\Gamma_{k+1} = \Gamma_k\setminus\{r_{k+1}\}\equiv\Gamma_k\equiv\Gamma$, contradicting our assumption that $k<n$ is the largest integer such that Γ_k is equivalent to Γ. Therefore, $k=n$ and Γ^{c} is equivalent to Γ, thus it is a prime subnetwork of Γ which is uniquely defined. □

With this result, we can use the core as a simplification of the original STN to obtain a maximal reduction in the size of representation, without changing the semantic essence.

4 Efficient Algorithm for Calculating the Core

The result below follows directly from Proposition 6 and Proposition 5, and allows for efficient identification of redundant constraints in STNs.

Proposition 8. *Let Γ and $\Gamma^{\triangle}$, as well as E and $E^{\triangle}$, be specified as in Lemma 2. Furthermore, let Γ be not degenerated. Then, for all $x, y \in V$ with $xy \in E^{\triangle}$ the following are equivalent:*

(i) $\Gamma(x, y)$ *is redundant in* Γ*;*
(ii) $\Gamma^{\triangle}(x, y) = \bigcap_{xz,zy \in E^{\triangle}} \Gamma^{\triangle}(x, z) \otimes \Gamma^{\triangle}(z, y)$.

With the aid of Proposition 8, we propose Algorithm 1 to efficiently calculate the core of a non-degenerated STN Γ. In this algorithm, we first construct a chordal graph $G^{\triangle} = (V, E^{\triangle})$ by triangulating the constraint graph $G = (V, E)$ of Γ (line 3). Then, we construct $\Gamma^{\triangle}$ as specified in Lemma 2 by using a certain algorithm EstablishPPC that refines Γ appropriately (line 4). After that, we iteratively retrieve the core of Γ edge by edge (lines 5–11). By Proposition 8, to identify the redundancy of a constraint involving an edge $xy \in E$, we only need to check if the constraint involving the corresponding edge xy in $E^{\triangle}$ equals the intersection of the compositions of all paths of length two from x to y in $E^{\triangle}$ (lines 8–11).

Algorithm 1: Core(Γ)

Input : A non-degenerated STN Γ.
Output: Γ^{c}, the core of Γ.
1 $\Gamma^{c} \leftarrow \emptyset$;
2 $G = (V, E) \leftarrow$ the constraint graph of Γ;
3 $G^{\triangle} = (V, E^{\triangle}) \leftarrow$ Triangulate(G);
4 $\Gamma^{\triangle} \leftarrow$ EstablishPPC($\Gamma, G^{\triangle}$);
5 **while** $E \neq \emptyset$ **do**
6     $xy \leftarrow E$.pop() ;
7     $I \leftarrow (x\, [-\infty, \infty]\, y)$;
8   **foreach** z *with* $(x, z), (z, y) \in E^{\triangle}$ **do**
9         $I \leftarrow I \cap (\Gamma^{\triangle}(x, z) \otimes \Gamma^{\triangle}(z, y))$;
10   **if** $\Gamma^{\triangle}(x, y) \neq I$ **then**
11         $\Gamma^{c} \leftarrow \Gamma^{c} \cup \Gamma(x, y)$;
12 **return** Γ^{c}.

Example 9. To calculate the core of the STN Γ in Figure 1a, Algorithm 1 first triangulates the constraint graph $G = (V, E)$ of Γ. In this case, as G is complete, any triangulation of G is still G. Then, by calculating the minimal network, $\Gamma^{\triangle}$ is obtained, which is the STN shown in Figure 2a. For each edge $x_i x_j$ in E, the algorithm checks if the constraint $\Gamma^{\triangle}(x_i, x_j)$ coincides with the intersection $\bigcap_{x_i x_k, x_k x_j \in E^{\triangle}} \Gamma^{\triangle}(x_i, x_k) \otimes \Gamma^{\triangle}(x_k, x_j)$. The result shows that $\Gamma^{\triangle}(x_1, x_2), \Gamma^{\triangle}(x_1, x_3), \Gamma^{\triangle}(x_1, x_5), \Gamma^{\triangle}(x_2, x_5)$, and $\Gamma^{\triangle}(x_3, x_4)$ are such constraints. Therefore, we know that the corresponding constraints in Γ are redundant, and we obtain the core Γ^{c} in Figure 1b by removing them.

In what follows, we analyze the time complexity of Algorithm 1 in terms of the number of triangles t in the graph $G^{\triangle}$. Note that a sparsely structured graph can contain much fewer triangles than a complete graph of the same number of vertices. For each execution of the while loop from (lines 5–13), the algorithm checks the triangles in $G^{\triangle}$ that contain xy as an edge. Hence, the while loop checks each triangle in $G^{\triangle}$ at most three times and runs in $O(t + |E|)$ time. To obtain a sparse chordal graph $G^{\triangle}$ (line 4), we can use the *maximum cardinality search* algorithm [19] and a simple *fill-in* procedure [12]. In particular, the maximum cardinality search algorithm visits the vertices of G in an order such that, at any point, a vertex is visited that has the largest number of visited neighbours. Consequently, a vertex ordering α is produced. Then, the fill-in procedure considers the vertices in G one by one following the vertex ordering α, and, connects each pair of vertices in the neighbourhood of the vertex at hand with an undirected *fill* edge (if an edge is not already present). The obtained graph $G^{\triangle}$ is then a triangulation of G [7]. The maximum cardinality search algorithm has the nice property that it does not lead to any fill edges if the graph is already chordal. The entire operation of triangulating G in the aforementioned manner is linear in the size of $G^{\triangle}$, viz., $O(|V| + |E^{\triangle}|)$. Further, to construct $\Gamma^{\triangle}$ (line 6), we can use the state-of-the-art algorithm $\mathrm{P}^3\mathrm{C}$ [13], which runs in $O(t)$ time. Therefore, Algorithm 1 runs in $O(t + |V| + |E^{\triangle}|)$ time.

5 Evaluation

In this section, we evaluate the previous theoretical results using a large benchmark dataset comprising over a thousand of STNs of various nature. Note that by the analysis that took place in the previous section, the performance of Algorithm 1 is strongly dependent on the algorithm $\mathrm{P}^3\mathrm{C}$ for constructing $G^{\triangle}$ which has been shown to be very efficient in [13]. In light of that, and as we are more interested in obtaining results on redundancy reduction for STNs, we will not report on the computation time of our implementation.

Regarding the benchmark dataset, we employed the dataset of 1491 STNs used in the work of Planken et al. [14]. The basic properties of that dataset are presented in Table 1. More details about the nature of each dataset can be found in [14, Section 4]. The STNs vary from random scale-free networks and parts of the road network of New York City, to STNs generated from hierarchical task networks (HTNs) and job-shop scheduling problems. Note that Chordal-1 and Scale-free-1 contain STNs of a fixed number of variables, while Chordal-2 and Scale-free-2 contain STNs of a variant number of variables. It is worth noting that a small percentage of STNs are degenerated. To be exact, $100/1491$ of the STNs are degenerated. The number of degenerated STNs varies per nature of dataset. For example, only 5 out of the 400 STNs of the job-shop scheduling problems dataset (Job-shop) are degenerated. On the other hand, over a third of the STNs of the 1000-variable scale-free STNs dataset are degenerated. In any case, all degenerated STNs were easily and minimally repaired by introducing a small weight ε to each degenerated interval $[a, a]$, hence, modifying each such interval to $[a - \varepsilon, a + \varepsilon]$. We also observe that Chordal-1 and Chordal-2, which are STNs whose constraint graphs are chordal, are the densest ones, followed by the STNs of the job-shop scheduling problems dataset, viz., Job-shop and the scale-free networks datasets, viz., Scale-free-1 and Scale-free-2. On the contrary, STNs derived from the road network of New York City (New York), HTNs (HTN), and diamond-shaped networks (Diamonds) are extremely sparse, to the point of resembling trees, and are thus almost devoid of redundancy (as we will see in the fol-

Table 1. Results on redundancy reduction

dataset	#STNs	#deg.[a]	properties[b]			reduction[c]			
			$\lvert\mathbf{V}\rvert$	$\lvert\mathbf{E}\rvert$	$\overline{\mathbf{D}}$	$\boldsymbol{\mu}$	**min**	**max**	$\boldsymbol{\sigma}$
Chordal-1	250	18	1 000	75 840–499 490	0.532	97.43%	94.41%	98.99%	1.25
Chordal-2	130	11	214–3 125	22 788–637 009	0.509	96.92%	94.61%	98.27%	0.98
Scale-free-1	130	48	1 000	1 996–67 360	0.039	10.27%	0.10%	55.23%	13.09
Scale-free-2	160	18	250–1 000	2 176–3 330	0.025	3.30%	0.37%	17.93%	3.74
New York	170	0	108–3 906	113–6 422	0.006	0.01%	0.00%	0.19%	0.04
Diamonds	130	0	111–2 751	111–2 751	0.006	0.00%	0.00%	0.00%	0.00
Job-shop	400	5	17–1 321	32–110 220	0.142	49.45%	0.00%	73.69%	16.91
HTN	121	0	500–625	748–1 599	0.007	0.02%	0.00%	0.15%	0.04

[a] The number of degenerated STNs.
[b] $\lvert\mathbf{V}\rvert$,$\lvert\mathbf{E}\rvert$, and $\overline{\mathbf{D}}$: the number of vertices, the number of edges, and the average density of the constraint graphs of STNs respectively.
[c] $\boldsymbol{\mu}$, **min**, **max**, and $\boldsymbol{\sigma}$: the average, the minimal, the maximal, and the standard deviation value of reduction rate respectively.

lowing evaluation).

Regarding the results of our evaluation, Table 1 provides a detailed description of redundancy reduction per nature of dataset. Given the fact that chordal STNs are the densest ones, it is not surprising to obtain reduction rates lying between 94.41% to 98.99%, with a reduction rate of about 97% on average. With respect to the job-shop scheduling problems, we obtained an average reduction rate of 49.45%, while the reduction rate among the STNs reaches as high as 73.69%. Further, we obtain a 10.27% average reduction rate for scale-free STNs of a fixed number of variables, and a 3.3% reduction rate for scale-free STNs of a variant number of variables. Similar as the case of the job-scheduling problems, some of the STNs have less redundant constraints because they are already sparse. Finally, as expected due to their extreme sparseness, no significant reduction rate is obtained for any of the STNs derived from the road network of New York City, HTNs, and diamond-shaped networks. As an example, all constraints in the dataset of diamond-shaped networks are non-redundant. Finally, note that removing a constraint from an STN simplifies its constraint graph structure by eliminating the corresponding edge. Multiple graph related tasks, such as storing, querying, representing and reasoning, and pattern matching could benefit from the high reduction in the number of edges for several types of STNs.

As we strongly hinted earlier, the density of the constraint graph of a given STN correlates with the reduction that we can achieve in the number of constraints of that STN. The question arises of how big the correlation between densities and reduction rates really is, at least with respect to the dataset at hand. To answer this question, we use the *Spearman's rank correlation coefficient* [21], a non-parametric test that is used to measure the degree of association between two variables. It assesses how well the relationship between two variables x and y can be described using a monotonic function. Like other correlation coefficients, this one varies between -1 and $+1$, with 0 implying no correlation. Correlations of -1 or $+1$ imply an exact monotonic relationship. Positive correlations imply that as x increases, so does y. Negative correlations imply that as x increases, y decreases. By using the average density $\overline{D}$ and the average reduction rate μ as our variables, along with their respective values as provided in Table 1, we can obtain a Spearman's rank correlation coefficient of 0.99, which demonstrates that each of our variables is almost a perfect monotone function of the other. This results suggest that a strong correlation between densities and reduction rates exists, despite the different nature of the datasets.

6 Conclusion and Future Work

In this paper, we investigated the redundancy problem of STNs. In particular, we showed that for any non-degenerated STN Γ, there is a unique minimal subset Γ^c of its constraints (i.e., the core of Γ) such that all of the constraints in Γ^c are not redundant in Γ and Γ^c has the same set of solutions as Γ. We proposed an efficient algorithm to calculate the core, which runs in time linear in the number of triangles in a chordal graph that results from triangulating the constraint graph of Γ. Our experiments on benchmark datasets unveiled a large amount of redundant constraints , which suggests that the cores of the STNs achieve significant reductions of redundancies in practice. With respect to our dataset, we also identified two measures with a strong correlation between them. For future work, it would be interesting to investigate whether that statistical dependence between our measures will hold if more datasets of various nature were to be considered. Further, we would like to devise efficient reasoning algorithms based on the simplified representations to accomplish tasks involving dynamic and uncertain information and interactions between multi-agents.

A Proofs

Proposition 5 . *Let Γ and $\Gamma^{\triangle}$, as well as E and $E^{\triangle}$, be specified as in Lemma 2. Then for all $x, y \in V$ with $xy \in E^{\triangle}$ the following are equivalent:*

(i) $\Gamma^{\triangle}(x,y)$ *is redundant in* $\Gamma^{\triangle}$

(ii) $\Gamma^{\triangle}(x,y) = \bigcap_{P\in\Pi(x,y,E^{\triangle}\setminus\{xy\})} \bigotimes \Gamma^{\triangle}(P)$

(iii) $\Gamma^{\triangle}(x,y) = \bigcap_{xz,zy\in E^{\triangle}} \Gamma^{\triangle}(x,z) \otimes \Gamma^{\triangle}(z,y)$

Proof. (i) $\Rightarrow$ (ii). Suppose $\Gamma^{\triangle}(x,y)$ is redundant in $\Gamma^{\triangle}$. Then, by applying Lemma 4, we know that

$$\Gamma^{\triangle}(x,y) \supseteq \bigcap_{P\in\Pi(x,y,E^{\triangle}\setminus\{xy\})} \bigotimes \Gamma^{\triangle}(P).$$

On the other hand, $\Gamma^{\triangle}(x,y)$ is minimal in $\Gamma^{\triangle}$. Therefore

$$\Gamma^{\triangle}(x,y) \subseteq \bigcap_{P\in\Pi(x,y,E^{\triangle}\setminus\{xy\})} \bigotimes \Gamma^{\triangle}(P).$$

and we showed (ii).

(ii) ⇒ (iii). Let P be a path between x and y on $E^\triangle \setminus \{xy\}$. Then, because $G^\triangle$ is chordal, there exists a $z' \in V$ with $(x,z'),(z',y) \in E^\triangle$, such that P is a concatenation of paths P_1 and P_2, where P_1 is between x and z' and P_2 is between z' and y. Since $\Gamma^\triangle(x,z')$ and $\Gamma^\triangle(z',y)$ are minimal in $\Gamma^\triangle$, we have

$$\bigotimes \Gamma^\triangle(P_1) \;\supseteq\; \Gamma^\triangle(x,z')$$

and

$$\bigotimes \Gamma^\triangle(P_2) \;\supseteq\; \Gamma^\triangle(z',y).$$

Hence,

$$\begin{aligned}\bigotimes \Gamma^\triangle(P) &= \bigotimes \Gamma^\triangle(P_1) \otimes \bigotimes \Gamma^\triangle(P_2)\\ &\supseteq \Gamma^\triangle(x,z') \otimes \Gamma^\triangle(z',y)\\ &\supseteq \bigcap_{xz,zy\in E^\triangle} \Gamma^\triangle(x,z) \otimes \Gamma^\triangle(z,y).\end{aligned}$$

Because the preceding statement holds for any path P between x and y on $E^\triangle \setminus \{xy\}$, we have

$$\bigcap_{P\in\Pi(x,y,E^\triangle\setminus\{xy\})} \bigotimes \Gamma^\triangle(P) \;\supseteq\; \bigcap_{xz,zy\in E^\triangle} \Gamma^\triangle(x,z)\otimes\Gamma^\triangle(z,y)$$

and by applying our assumption (ii), we have

$$\Gamma^\triangle(x,y) \;\supseteq\; \bigcap_{xz,zy\in E^\triangle} \Gamma^\triangle(x,z)\otimes\Gamma^\triangle(z,y).$$

The other inclusion

$$\Gamma^\triangle(x,y) \;\subseteq\; \bigcap_{xz,zy\in E^\triangle} \Gamma^\triangle(x,z)\otimes\Gamma^\triangle(z,y)$$

follows from the minimality of $\Gamma^\triangle(x,y)$ in $\Gamma^\triangle$. Thus we showed (iii).

(iii) ⇒ (i). We first observe that for all $x,y,z \in V$ with $xy,xz,zy \in E^\triangle$

$$\Gamma^\triangle \setminus \{\Gamma^\triangle(x,y)\} \;\models\; \{\Gamma^\triangle(x,z),\Gamma^\triangle(z,y)\}$$

and

$$\{\Gamma^\triangle(x,z),\Gamma^\triangle(z,y)\} \;\models\; \Gamma^\triangle(x,z)\otimes\Gamma^\triangle(z,y)$$

and therefore

$$\Gamma^\triangle \setminus \{\Gamma^\triangle(x,y)\} \;\models\; \Gamma^\triangle(x,z)\otimes\Gamma^\triangle(z,y).$$

Thus we have

$$\Gamma^\triangle \setminus \{\Gamma^\triangle(x,y)\} \;\models\; \bigcap_{xz,zy\in E^\triangle} \Gamma^\triangle(x,z)\otimes\Gamma^\triangle(z,y)$$

By applying our assumption (iii) we then have

$$\Gamma^\triangle \setminus \{\Gamma^\triangle(x,y)\} \;\models\; \Gamma^\triangle(x,y)$$

and we showed (i). □

Non-degenerated STNs have the following property.

Lemma 9. *Let Γ and $\Gamma^\triangle$, as well as E and $E^\triangle$, be specified as in Lemma 2. Suppose Γ is not degenerated. Then there is a fixed $\varepsilon > 0$ such that for any cycle $C \in \Pi(x,x,E)$ we have that*

$$\bigotimes \Gamma(C) \;\supseteq\; (x\,[-\varepsilon,\varepsilon]\,x)$$

Proof. We first note that there exists an $\varepsilon > 0$ such that for any $x \in V$ and for any path $P \in \Pi(x,x,E^\triangle)$ of length 3, i.e., P is a triangle, we have $\bigotimes\Gamma(P) \supseteq (x\,[-\varepsilon,\varepsilon]\,x)$. This is because there are only finitely many triangles in $P^\triangle$ and the composition of non-degenerated constraints is again non-degenerate. Then, because $G^\triangle$ is chordal, for any cycle $C \in \Pi(x,x,E)$ with $xy \in C$ there exists a $z \in V$ with $xz,zy \in E^\triangle$ such that $C = P_1 + P_2 + yx$ with $P_1 \in \Pi(x,z,E)$ and $P_2 \in \Pi(z,y,E)$. Then

$$\begin{aligned}\bigotimes\Gamma(C) &= \bigotimes\Gamma(P_1)\otimes\bigotimes\Gamma(P_2)\otimes\Gamma(y,x)\\ &\supseteq \Gamma(x,z)\otimes\Gamma(z,y)\otimes\Gamma(y,x)\\ &\supseteq (x\,[-\varepsilon,\varepsilon]\,x)\end{aligned}$$

□

Example 10. The STN Γ_2 in Figure 2b is not degenerated. Consider the paths $C_1 = (x_2x_4)$, $C_2 = (x_4x_3, x_3x_5, x_5x_4)$, $C_3 = (x_4x_3, x_3x_2)$. Note that $C = C_1 + C_2 + C_3$ and C_2 are cycles. Then $\bigotimes\Gamma_2(C) = \bigotimes\Gamma_2(C_1) \otimes \bigotimes\Gamma_2(C_2)\otimes\bigotimes\Gamma_2(C_3) = (x_2[11,15]x_4)\otimes(x_4[-80,41]x_4)\otimes(x_4[-52,30]x_2) = (x_2[-121,86]x_2) \supseteq (x_2[-41,45]x_2) = \bigotimes\Gamma_2(C_1)\otimes\bigotimes\Gamma_2(C_3) \supseteq (x_2[-\varepsilon_C,\varepsilon_C]x_2)$, where we can choose $\varepsilon_C = 41$.

Proposition 6 . *Let Γ and $\Gamma^\triangle$, as well as E and $E^\triangle$, be specified as in Lemma 2. Furthermore, Γ is not degenerated. Then for all $x,y \in V$ with $xy \in E^\triangle$ the following are equivalent:*

(i) $\Gamma(x,y)$ *is redundant in* Γ
(ii) $\Gamma^\triangle(x,y)$ *is redundant in* $\Gamma^\triangle$
(iii) $\Gamma^\triangle(x,y) \;=\; \bigcap_{\substack{P\in\Pi(x,y,E)\\ |P|\geq 2}} \bigotimes\Gamma(P)$

Proof. (i) ⇒ (ii). Because $\Gamma^\triangle \models \Gamma$, we have

$$\Gamma^\triangle \setminus \{\Gamma^\triangle(x,y)\} \;\models\; \Gamma\setminus\{\Gamma(x,y)\},$$

and since, by our assumption, $\Gamma\setminus\{\Gamma(x,y)\} \equiv \Gamma$, we have

$$\Gamma^\triangle \setminus \{\Gamma^\triangle(x,y)\} \;\models\; \Gamma.$$

As $\Gamma^\triangle \equiv \Gamma$ by Lemma 2, we finally have

$$\Gamma^\triangle \setminus \{\Gamma^\triangle(x,y)\} \;\models\; \Gamma^\triangle,$$

and showed (ii).

(ii) ⇒ (iii). Suppose $\Gamma^\triangle(x,y)$ is redundant in $\Gamma^\triangle$. Then, by Proposition 5 we know that

$$\Gamma^\triangle(x,y) \;=\; \bigcap_{xz,zy\in E^\triangle} \Gamma^\triangle(x,z)\otimes\Gamma^\triangle(z,y).$$

By applying Lemma 3 we then have

$$\Gamma^\triangle(x,y)$$
$$= \bigcap_{xz,zy\in E^\triangle} \left(\bigcap_{P_1\in\Pi(x,z,E)} \bigotimes\Gamma(P_1)\right) \otimes \left(\bigcap_{P_2\in\Pi(z,y,E)} \bigotimes\Gamma(P_2)\right)$$

Since STP constraints are distributive (composition distributes over intersections) we have

$$\Gamma^\triangle(x,y) \;=\; \bigcap_{\substack{xz,zy\in E^\triangle\\ P_1\in\Pi(x,z,E)\\ P_2\in\Pi(z,y,E)}} \bigotimes\Gamma(P_1+P_2) \tag{4}$$

On the other hand, because $G^{\triangle}$ is chordal, P is a path between x and y on E with $|P| \geq 2$ if and only if there exists a $z \in V$ with $(x,z),(z,y) \in E^{\triangle}, z \neq x, z \neq y$, such that P is a concatenation of paths P_1 and P_2, where P_1 is between x and z and P_2 is between z and y. Hence we have

$$\bigcap_{\substack{P \in \Pi(x,y,E) \\ |P| \geq 2}} \bigotimes \Gamma(P) \; = \bigcap_{\substack{xz,zy \in E^{\triangle} \\ P_1 \in \Pi(x,z,E) \\ P_2 \in \Pi(z,y,E)}} \bigotimes \Gamma(P_1 + P_2) \tag{5}$$

From (4) and (5) it follows

$$\Gamma^{\triangle}(x,y) \; = \bigcap_{\substack{P \in \Pi(x,y,E) \\ |P| \geq 2}} \bigotimes \Gamma(P)$$

and we showed (iii).

(iii) $\Rightarrow$ (i): Suppose

$$\Gamma^{\triangle}(x,y) \; = \bigcap_{\substack{P \in \Pi(x,y,E) \\ |P| \geq 2}} \bigotimes \Gamma(P)$$

We first note that we can partition the set of paths between x and y on E with $|P| \geq 2$ into two subsets $\mathcal{P}_1$ and $\mathcal{P}_2$, where

$$\mathcal{P}_1 := \{P \in \Pi(x,y,E) \mid xy \in P, |P| \geq 2\},$$

$$\mathcal{P}_2 := \{P \in \Pi(x,y,E) \mid xy \notin P, |P| \geq 2\}.$$

Hence

$$\Gamma^{\triangle}(x,y) \; = \bigcap_{P \in \mathcal{P}_1 \dot{\cup} \mathcal{P}_2} \bigotimes \Gamma(P). \tag{6}$$

We note that $\mathcal{P}_1$ consists of paths that are of the form $C_1 + (xy)$, $(xy) + C_2$, or $C_1 + (xy) + C_2$, where C_1 and C_2 are cycles on E with $|C_1| \geq 2$ and $|C_2| \geq 2$. Then, since Γ is not degenerated, by Lemma 9, there is an $\varepsilon > 0$ such that

$$\bigotimes \Gamma(C_1) \; \supseteq \; (x\,[-\varepsilon, \varepsilon]\,x),$$

$$\bigotimes \Gamma(C_2) \; \supseteq \; (y\,[-\varepsilon, \varepsilon]\,y),$$

and we have, without loss of generality, for $P = C_1 + \{xy\} + C_2$

$$\begin{aligned}
\bigotimes \Gamma(P) &= \bigotimes \Gamma(C_1 + \{xy\} + C_2) \\
&= \bigotimes \Gamma(C_1) \; \otimes \; \Gamma(x,y) \; \otimes \; \bigotimes \Gamma(C_2) \\
&\supseteq (x\,[-\varepsilon, \varepsilon]\,x) \otimes \Gamma(x,y) \\
&= (x\,[a-\varepsilon, b+\varepsilon]\,y),
\end{aligned}$$

where $\Gamma(x,y) = (x\,[a,b]\,y)$. Hence,

$$\bigcap_{P \in \mathcal{P}_1} \bigotimes \Gamma(P) \; \supseteq \; (x\,[a-\varepsilon, b+\varepsilon]\,y). \tag{7}$$

Thus by (6) and (7) we have

$$\begin{aligned}
\Gamma^{\triangle}(x,y) &= \bigcap_{P \in \mathcal{P}_1 \dot{\cup} \mathcal{P}_2} \bigotimes \Gamma(P) \\
&= \bigcap_{P \in \mathcal{P}_1} \bigotimes \Gamma(P) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\
&\supseteq (x\,[a-\varepsilon, b+\varepsilon]\,y) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\
&\supseteq (x\,[a,b]\,y) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\
&= (x\,[a,b]\,y) \cap \bigcap_{P \in \mathcal{P}_1} \bigotimes \Gamma(P) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\
&= \Gamma(x,y) \; \cap \bigcap_{P \in \mathcal{P}_1 \dot{\cup} \mathcal{P}_2} \bigotimes \Gamma(P) \\
&= \Gamma(x,y) \; \cap \; \Gamma^{\triangle}(x,y) \\
&= \Gamma^{\triangle}(x,y)
\end{aligned}$$

Consequently we have

$$\begin{aligned}
&(x\,[a-\varepsilon, b+\varepsilon]\,y) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\
&\quad = (x\,[a,b]\,y) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P).
\end{aligned}$$

By setting $(x\,[c,d]\,y) := \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P)$ we then have

$$(x\,[a-\varepsilon, b+\varepsilon]\,y) \cap (x\,[c,d]\,y) = (x\,[a,b]\,y) \cap (x\,[c,d]\,y),$$

which yields $[\max(a - \varepsilon, c), \min(b + \varepsilon, d)] = [\max(a,c), \min(b,d)]$ which holds only if $a \leq c$ and $b \geq d$, i.e., $(x\,[a,b]\,y) \supseteq (x\,[c,d]\,y)$. Therefore

$$\Gamma(x,y) \supseteq \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P).$$

Since

$$\bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) = \bigcap_{P \in \Pi(x,y,E \setminus \{xy\})} \bigotimes \Gamma(P)$$

we have

$$\Gamma(x,y) \supseteq \bigcap_{P \in \Pi(x,y,E \setminus \{xy\})} \bigotimes \Gamma(P)$$

and by Lemma 4 we have that $\Gamma(x,y)$ is redundant in Γ. $\square$

REFERENCES

[1] James F. Allen, 'Maintaining knowledge about temporal intervals', *Commun. ACM*, **26**, 832–843, (1983).

[2] Roman Barták, Robert A. Morris, and K. Brent Venable, 'An introduction to constraint-based temporal reasoning', *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **8**(1), 1–121, (2014).

[3] John L. Bresina, Ari K. Jnsson, Paul H. Morris, and Kanna Rajan, 'Activity planning for the mars exploration rovers', in *ICAPS*, pp. 40–49, (2005).

[4] Patrick R. Conrad, Julie A. Shah, and Brian C. Williams, 'Flexible execution of plans with choice', in *ICAPS*, (2009).

[5] Rina Dechter, Itay Meiri, and Judea Pearl, 'Temporal constraint networks', *Artif. Intell.*, **49**(1-3), 61–95, (1991).

[6] Michael Fisher, Dov M. Gabbay, and L. Vila, *Handbook of temporal reasoning in artificial intelligence*, number 1 in Foundations of artificial intelligence, Elsevier, Amsterdam ; Boston, 1st edn., 2005.

[7] Delbert Fulkerson and Oliver Gross, 'Incidence matrices and interval graphs.', *Pacific J. Math.*, **15**(3), 835–855, (1965).

[8] Georg Gottlob and Christian G. Fermüller, 'Removing redundancy from a clause', *Artif. Intell.*, **61**(2), 263 – 289, (1993).

[9] Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both, 'On redundant topological constraints', *Artif. Intell.*, **225**, 51–76, (2015).

[10] Paolo Liberatore, 'Redundancy in logic I: CNF propositional formulae', *Artif. Intell.*, **163**(2), 203–232, (2005).

[11] Nicola Muscettola, Paul H. Morris, and Ioannis Tsamardinos, 'Reformulating temporal plans for efficient execution', in *KR*, pp. 444–452, (1998).

[12] Seymour Parter, 'The use of linear graphs in gauss elimination', *SIAM review*, **3**(2), 119–130, (1961).

[13] Léon Planken, Mathijs de Weerdt, and Roman van der Krogt, 'P^3C: A new algorithm for the simple temporal problem', in *ICAPS*, pp. 256–263, (2008).

[14] Léon Planken, Mathijs de Weerdt, and Roman van der Krogt, 'Computing all-pairs shortest paths by leveraging low treewidth', *J. Artif. Intell. Res.*, **43**, 353–388, (2012).

[15] David A. Randell, Zhan Cui, and Anthony G. Cohn, 'A spatial logic based on regions and connection', in *KR*, pp. 165–176, (1992).

[16] Francesca Rossi, Kristen Brent Venable, and Neil Yorke-Smith, 'Uncertainty in soft temporal constraint problems: A general framework and controllability algorithms forthe fuzzy case', *J. Artif. Intell. Res.*, 617–674, (2006).

[17] Julie A Shah and Brian C Williams, 'Fast dynamic scheduling of disjunctive temporal constraint networks through incremental compilation', in *ICAPS*, pp. 322–329, (2008).

[18] Michael Sioutis, Sanjiang Li, and Jean-François Condotta, 'Efficiently characterizing non-redundant constraints in large real world qualitative spatial networks', in *IJCAI*, pp. 3229–3235, (2015).

[19] Robert Endre Tarjan and Mihalis Yannakakis, 'Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs', *SIAM J. Comput.*, **13**(3), 566–579, (1984).

[20] Marc B. Vilain and Henry A. Kautz, 'Constraint propagation algorithms for temporal reasoning', in *AAAI*, pp. 377–382, (1986).

[21] Daniel Zwillinger and Stephen Kokoska, *CRC standard probability and statistics tables and formulae*, CRC Press, 1999.