Prime Scenarios in Qualitative Spatial and Temporal Reasoning

Yakoub Salhi
CRIL UMR 8188, Université d'Artois & CNRS, France

Michael Sioutis
LIRMM UMR 5506, Université de Montpellier & CNRS, France

Abstract

The concept of prime implicant is a fundamental tool in Boolean algebra, which is used in Boolean circuit design and, recently, in explainable AI. This study investigates an analogous concept in qualitative spatial and temporal reasoning, called prime scenario. Specifically, we define a prime scenario of a qualitative constraint network (QCN) as a minimal set of decisions that can uniquely determine solutions of this QCN. We propose in this paper a collection of algorithms designed to address various problems related to prime scenarios. The first three algorithms aim to generate a prime scenario from a scenario of a QCN. The main idea consists in using path consistency to identify the constraints that can be ignored to generate a prime scenario. The next two algorithms focus on generating a set of prime scenarios that cover all the scenarios of the original QCN: The first algorithm examines every branch of the search tree, while the second is based on the use of a SAT encoding. Our last algorithm is concerned with computing a minimum-size prime scenario by using a MaxSAT encoding built from countermodels of the original QCN. We show that this algorithm is particularly useful for measuring the robustness of a QCN. Finally, a preliminary experimental evaluation is performed with instances of Allen's Interval Algebra to assess the efficiency of our algorithms and, hence, also the difficulty of the newly introduced problems here.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming; Computing methodologies → Temporal reasoning; Computing methodologies → Spatial and physical reasoning

Keywords and phrases Spatial and Temporal Reasoning, Qualitative Constraints, Prime Scenario, Prime Implicant, Robustness Measurement

Digital Object Identifier 10.4230/LIPIcs.TIME.2023.5

Supplementary Material *Software*: `https://seafile.lirmm.fr/d/9c0cbd2cd0954252ab96/`

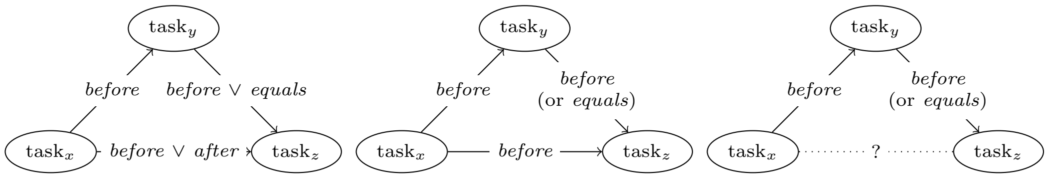
Funding *Michael Sioutis*: The work was partially funded by the Agence Nationale de la Recherche (ANR) for the "Hybrid AI" project that is tied to the chair of Dr. Sioutis, and the I-SITE program of excellence of Université de Montpellier that complements the ANR funding.

1 Introduction

The role of prime implicants is pivotal in various domains, including knowledge compilation [2, 5], Boolean circuit simplification [21, 22, 17], and diagnosis [7, 28]. Additionally, many recent research works have employed prime implicants to explain decisions by compiling machine learning classifiers into Boolean circuits [30, 9, 10, 11, 4].

Qualitative Spatial and Temporal Reasoning (QSTR) focuses on reasoning about space and time using qualitative human-like descriptions, e.g., x $\{$*is north of*$\}$ y, as opposed to quantitative ones [15]. QSTR is a rich symbolic AI framework concerned with studying various types of spatial and temporal relationships, such as the relative position of objects [14], the ordering and duration of events [1], and the mereotopology of regions [23]. By employing qualitative representations, QSTR allows modeling and reasoning about complex entities and phenomena in a more flexible and intuitive way without resorting to, often prohibitively expensive, numerical precision.

30th International Symposium on Temporal Representation and Reasoning (TIME 2023).
Editors: Alexander Artikis, Florian Bruse, and Luke Hunsberger; Article No. 5; pp. 5:1–5:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) A consistent plan as a QCN. **(b)** A scenario of the QCN. **(c)** A *prime scenario* of the QCN.

Figure 1 An illustration of the knowledge compilation notion of *prime scenario* of a qualitative constraint network (QCN) (see also Definition 4); a set of prime scenarios can form a *prime scenario cover* of a QCN, for such a cover, here, we only need to additionally consider the prime scenario in Figure 1c with task$_y$ $\{equals\}$ task$_z$ instead of task$_y$ $\{before\}$ task$_z$.

In this study, we introduce a novel notion, called prime scenario, that serves as the QSTR analogue of the notion of prime implicant. A prime scenario is defined as a minimal set of decisions that can only lead to solutions of the original qualitative constraint network (QCN); see Figure 1. While the notion of prime implicant shares similarities with that of prime scenarios, there are significant distinctions that hinder the direct application of prime implicant computation approaches to our context. Notably, prime scenarios are based on binary relations between variables, while prime implicants rely on truth values of variables. For instance, any literal entailed by a prime implicant belongs to that implicant; in contrast, singleton constraints entailed by prime scenarios do not have this property. To better grasp this point, consider the following constraints: x $\{before\}$ y, y $\{before, equals\}$ z, and x $\{before, after\}$ z (Figure 1a); although the two first constraints entail x $\{before\}$ z, this constraint does not belong to the prime scenario $\{x$ $\{before\}$ y, y $\{before\}$ $z\}$ (Figure 1c): it is redundant.

It is worth mentioning that our notion of prime scenario has some relation to that of prime sub-QCN introduced in [13]. Specifically, the constraints that are not included in the prime scenario are redundant when we require the instantiated part within the prime scenario. In particular, for every *atomic* QCN, the prime scenarios are the prime sub-QCNs. Intuitively, the difference between prime scenarios and prime sub-QCNs bears a resemblance to the difference between prime implicants and the formulas resulting from the elimination of redundant clauses in propositional formulas expressed in conjunctive normal form.

To illustrate the *motivation* behind our novel work here, consider the example of machine learning classifiers that can be compiled into QCNs, much like as in the ongoing research involving Boolean circuits that we mentioned in the beginning. In this case, the solutions correspond to positive decisions, while the remaining interpretations correspond to negative ones. To explain the decisions made by these classifiers, prime scenarios can be used in a similar way as prime implicants are used to explain decisions of classifiers compiled into Boolean circuits. In particular, a prime scenario that covers a solution can be seen as a sufficient reason behind the decision associated with this solution. What is more, the notions of prime scenario and prime scenario cover that we introduce here (Figure 1), form a step towards compiling QCNs and open new avenues for research in this field: Prime scenarios can be used in the context of compilation of spatio-temporal knowledge bases, and prime scenario covers would be a classical way to perform such compilations.

With regard to the discussion above, our main *contributions* are fivefold: (*i*) We define the notion of prime scenario of a QCN and propose three algorithms for computing it (Section 3); (*ii*) we introduce and study the related problem of prime scenario cover of a QCN and present two distinct algorithms for solving it, a constraint- and a SAT-based one (Section 4); (*iii*) we focus on obtaining a minimum-size prime scenario of a QCN and devise a countermodel-

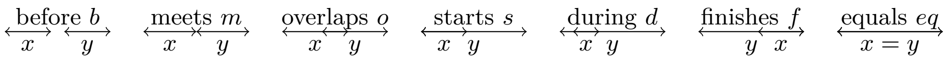


Figure 2 A representation of the 13 base relations b of IA, each one relating two potential intervals x and y as in $x\ b\ y$; the converse of b, i.e., b^{-1}, can be denoted by bi and is omitted in the figure.

based MaxSAT encoding to tackle this task, and (*iv*) we show how the minimum-size prime scenarios are useful for measuring the robustness of a QCN (Section 5); and finally (*v*) we experimentally evaluate all our algorithms and make our code available for any interested researcher to use (Section 6).

2 Preliminaries

A qualitative spatial or temporal constraint language is based on a finite set B of *jointly exhaustive and pairwise disjoint* relations, called *base relations*, and defined over an infinite domain D [15] (e.g., $\mathbb{R}$). The base relations of such a language can be used to represent the definite knowledge between any two of its entities (e.g., x *contains* y). The set B contains the identity relation Id, and is closed under the *converse* operation ($^{-1}$). Indefinite knowledge can be specified by a union of possible base relations, and is represented by the set containing them. Hence, 2^{B} represents the total set of relations. The set 2^{B} is equipped with the usual set-theoretic operations of union and intersection, the converse operation, and the *weak composition* operation, denoted by $\diamond$ [15]. For all $r \in 2^{\mathsf{B}}$, we have that $r^{-1} = \bigcup\{b^{-1} \mid b \in r\}$. The weak composition ($\diamond$) of two base relations $b, b' \in \mathsf{B}$ is defined as the smallest (i.e., most restrictive) relation $r \in 2^{\mathsf{B}}$ that includes $b \circ b'$, or, formally, $b \diamond b' = \{b'' \in \mathsf{B} \mid b'' \cap (b \circ b') \neq \emptyset\}$, where $b \circ b' = \{(x, y) \in \mathsf{D} \times \mathsf{D} \mid \exists z \in \mathsf{D}$ such that $(x, z) \in b \wedge (z, y) \in b'\}$ is the (true) composition of b and b'. For all $r, r' \in 2^{\mathsf{B}}$, we have that $r \diamond r' = \bigcup\{b \diamond b' \mid b \in r,\ b' \in r'\}$.

As an illustration, consider the well-known qualitative temporal constraint language of Interval Algebra (IA) [1]. IA considers time intervals (as temporal entities) and the set of base relations $\mathsf{B} = \{eq\ (= \mathsf{Id}),\ b,\ bi,\ m,\ mi,\ o,\ oi,\ s,\ si,\ d,\ di,\ f,\ fi\}$ to encode knowledge about the temporal relations between intervals on the real line, as described in Figure 2.

Finally, representing and reasoning about qualitative spatio-temporal information can be facilitated by a *qualitative constraint network (QCN)*; we recall the following definition:

▶ **Definition 1.** *A* qualitative constraint network (QCN) *is a tuple* (V, C) *where:*
- $V = \{v_1, \ldots, v_n\}$ *is a finite set of variables over some infinite domain* D *(e.g.,* $\mathbb{R}$*);*
- *and* C *is a mapping* $C : V \times V \rightarrow 2^{\mathsf{B}}$ *associating a relation with each pair of variables s.t.* $C(v, v) = \{\mathsf{Id}\}$ *for all* $v \in V$*, and* $C(v, v') = (C(v', v))^{-1}$ *for all* $v, v' \in V$*.*

For convenience, we often consider that the set of variables of a QCN consists of integers, and we use $[\![\mathcal{N}]\!]$ to denote the set $\{(i, j) \in V \times V : i < j\}$.

A QCN $\mathcal{N} = (V, C)$ is said to be *trivially inconsistent* iff $\exists v, v' \in V$ such that $C(v, v') = \emptyset$.

A *solution* of a QCN $\mathcal{N} = (V, C)$ is a mapping $\sigma : V \rightarrow \mathsf{D}$ such that $\forall v, v' \in V$, $\exists b \in C(v, v')$ such that $(\sigma(v), \sigma(v')) \in b$; $\mathcal{N}$ is said to be *consistent* iff it admits a solution.

A *sub-QCN* $\mathcal{N}'$ of $\mathcal{N}$, denoted by $\mathcal{N}' \subseteq \mathcal{N}$, is a QCN (V, C') such that, $\forall u, v \in V$, $C'(u, v) \subseteq C(u, v)$. (This term is also known as a *refined QCN* in the literature.)

A *scenario* of $\mathcal{N}$ is a consistent atomic sub-QCN $\mathcal{S}$ of $\mathcal{N}$, where a QCN $\mathcal{S} = (V, C')$ is *atomic* iff $\forall v, v' \in V$, $|C(v, v')| = 1$. To refer to the set of scenarios of $\mathcal{N}$, we employ the notation $\mathsf{Scenarios}(\mathcal{N})$.

Throughout the paper, we use the following notational conventions for a QCN $\mathcal{N} = (V, C)$:

- For two variables $v, v' \in V$, we use $\mathcal{N}[v, v']$ to denote the relation $C(v, v')$.
- For two variables $v, v' \in V$ and a relation $r \in 2^{\mathsf{B}}$, we use $v\ r\ v'$ to denote that $C(v, v') = r$ when there is no ambiguity about the considered QCN.
- For two variables $v, v' \in V$ and a relation $r \in 2^{\mathsf{B}}$, we use $\mathcal{N}_{[v,v']/r}$ to denote the result of substituting $C(v, v')$ with r in $\mathcal{N}$, i.e., $\mathcal{N}_{[v,v']/r}$ is the QCN (V, C') defined by $C'(v, v') = r$, $C'(v', v) = r^{-1}$ and, $\forall (u, u') \in (V \times V) \setminus \{(v, v'), (v', v)\}$, $C'(u, u') = C(u, u')$.

A *counter-scenario* of a QCN $\mathcal{N} = (V, C)$ is a consistent atomic QCN $\mathcal{S}$ over V that is *not* a scenario of $\mathcal{N}$, i.e., there exist $i, j \in V$ such that $\mathcal{S}[i, j] \nsubseteq \mathcal{N}[i, j]$. We denote the set of counter-scenarios of $\mathcal{N}$ as $\mathsf{CounterS}(\mathcal{N})$.

In general, there exists only one type of QCNs that do not admit any counter-scenario: those in which every constraint is *universal*, i.e., it contains all base relations. In such cases, we use $\mathcal{N}_{\top}$ to denote the universal QCN when the set of variables is assumed to be known, or to refer to this type of QCNs.

Given a set of variables V, we define a *q-assignment* over V as a partial function f from $\{(i, j) : i, j \in V \text{ and } i < j\}$ to B. We use $\mathcal{N}_V^f$ to denote the QCN (V, C) defined as follows:

- for each $(i, j) \in \mathsf{dom}(f)$, $C(i, j) = \{f(i, j)\}$; and
- for each $i, j \in V$ with $i < j$ and $(i, j) \notin \mathsf{dom}(f)$, $C(i, j) = \mathsf{B}$.

Given a QCN $\mathcal{N}$, we use $\mathsf{min}(\mathcal{N})$ to denote the equivalent *minimal* sub-QCN of $\mathcal{N}$ [26], i.e., the sub-QCN that contains only the feasible base relations of the original one.

It is important to note that in this paper, we focus on calculi with the following property:

▶ Note 2. For any q-assignment f over V, the closure of $\mathcal{N}_V^f$ under path consistency (with weak composition, or, equivalently, under *algebraic closure* [25]) yields *min*$(\mathcal{N}_V^f)$.

This property holds for many widely adopted qualitative calculi, such as IA [1] (mentioned earlier) and $\mathsf{RCC8}$ [23]; a fuller listing is provided in the proof of Theorem 2 in [16].

As a direct consequence of the aforementioned property, we also have that, for any q-assignment f over V, path consistency decides the consistency of $\mathcal{N}_V^f$.

Given a consistent atomic QCN $\mathcal{S} = (V, C)$, we say that a q-assignment f over V *covers* $\mathcal{S}$ if $\mathcal{S}$ is a scenario of $\mathcal{N}_V^f$.

In the sequel, we also represent a q-assignment as a set of expressions of the form $(i, j) \mapsto b$: f corresponds to the set $\{(i, j) \mapsto f(i, j) : (i, j) \in \mathsf{dom}(f)\}$.

3 Prime Scenarios

In this section, we introduce the concept of prime scenario, which can be thought of as analogous to that of prime implicant in propositional logic.

▶ **Definition 3** (Convergent Q-Assignment). *A* convergent q-assignment *(CQA) of a QCN $\mathcal{N} = (V, C)$ is a q-assignment π over V where (1) $\mathcal{N}_V^\pi$ is consistent, and (2) every scenario of $\mathcal{N}_V^\pi$ is a scenario of $\mathcal{N}$.*

Convergent q-assignments are similar in concept to implicants in propositional logic. Property 1 states that a CQA maintains consistency, and Property 2 says that a CQA cannot lead to a scenario that does not satisfy the original QCN. By virtue of this second property, $\pi(i, j) \in C(i, j)$ holds for every $(i, j) \in \mathsf{dom}(\pi)$.

▶ **Definition 4** (Prime Scenario). *A* prime scenario *of a QCN $\mathcal{N}$ is a convergent q-assignement π of $\mathcal{N}$ where for every $D \subsetneq \mathsf{dom}(\pi)$, $\pi|_D$ is not a convergent q-assignment.*

Algorithm 1 FINDONEPS_1($\mathcal{N}, \mathcal{S}$).

in : A QCN $\mathcal{N} = (V, C)$ and a complete scenario $\mathcal{S}$ of $\mathcal{N}$
out : A prime scenario π that covers $\mathcal{S}$

```
π ← {(i, j) ↦ b : (i, j) ∈ ⟦N⟧, b ∈ S[i, j], N[i, j] ≠ B};
for (i, j) ∈ ⟦N⟧ do
    N′ ← PathConsistency(N^π_V[i,j]/B);
    if N′ ⊆ N then
        π ← π|dom(π)\{(i,j)};
return π
```

In other words, a prime scenario is a CQA that has a minimal domain (w.r.t. set inclusion). We use $\mathsf{PSes}(\mathcal{N})$ to denote the set of prime scenarios of $\mathcal{N}$.

To distinguish between prime scenarios and standard scenarios more clearly, we will refer to the latter as complete scenarios.

▶ **Proposition 5.** *The problem of determining whether a q-assignment is a prime scenario of a QCN is tractable.*

Proof. We show that we can determine whether a q-assignment is a prime scenario by linearly applying the polytime procedure of path consistency. Let $\mathcal{N} = (V, C)$ be a QCN and π a q-assignment of $\mathcal{N}$. To determine whether π is a prime scenario, we first need to check that $\mathcal{N}^{\pi}_{V}$ is consistent, which can be done using path consistency (see Note 2 and the discussion after). Using, again, path consistency, we can determine whether every complete scenario of $\mathcal{N}^{\pi}_{V}$ is a complete scenario of $\mathcal{N}$ (see Note 2). Indeed, we only have to show $\mathcal{N}' \subseteq \mathcal{N}$, where $\mathcal{N}'$ is the result of applying path consistency on $\mathcal{N}^{\pi}_{V}$. Similarly, to show that π is minimal w.r.t. set inclusion, we can use path consistency to show that, for every $(i, j) \in \mathsf{dom}(\pi)$, $\mathcal{N}_{ij} \not\subseteq \mathcal{N}$, where $\mathcal{N}_{ij}$ is the result of applying path consistency on $\mathcal{N}^{\pi|_{\mathsf{dom}(\pi)\setminus\{(i,j)\}}}_{V}$. ◀

Let us recall that a prime implicant of a propositional formula is a minimal consistent conjunction of literals whose Boolean models are models of this formula. This definition clearly shows that prime implicants and prime scenarios are similar in concept. However, a closer examination reveals that there are significant differences between them, making the study of prime scenarios highly compelling and of great interest. First, prime scenarios are more complex structures by involving constraints and qualitative relations. Secondly, universal constraints, which are analogous to tautologies in the case of propositional logic, can be involved in prime scenarios, whereas tautologies can be simply ignored in prime implicant computation. Consider, for instance, the QCN $\mathcal{N}$ in Point Algebra PA [33] ($\mathsf{B} = \{<, =, >\}$) that corresponds to the following constraints: $i\{<, =, >\}j$, $j\{<, =, >\}k$ and $i\{<\}k$; we obtain that $\pi = \{(i, j) \mapsto <, (j, k) \mapsto <\}$ is a prime scenario of $\mathcal{N}$ even though the two involved constraints in π are universal in $\mathcal{N}$. Thirdly, unlike entailed literals in the case of prime implicants, the singleton constraints entailed from a prime scenario do not belong to it. The prime implicants benefit significantly from this advantage, as it enables the use of unit propagation to efficiently compute them.

Computing One Prime Scenario

The focus here is on the computation of a prime scenario that covers a given complete scenario. We propose three different algorithms that are centered around the idea of computing a prime scenario from a precomputed CQA.

Algorithm 2 FindOnePS_2($\mathcal{N}, \mathcal{S}$).

in : A QCN $\mathcal{N} = (V, C)$ and a complete scenario $\mathcal{S}$ of $\mathcal{N}$
out : A prime scenario π that covers $\mathcal{S}$
$\mathcal{N}' \leftarrow \mathcal{N}_\top$;
$P \leftarrow [\![\mathcal{N}]\!]$;
while $\mathcal{N}' \not\subseteq \mathcal{N}$ **do**
  Let $(i, j) \in P$ s.t. $|\mathcal{N}'[i, j]| > 1$ and $\mathcal{N}[i, j] \neq \mathsf{B}$;
  $\mathcal{N}' \leftarrow$ PathConsistency($\mathcal{N}'_{[i,j]/\mathcal{S}[i,j]}$);
  $P \leftarrow P \setminus \{(i, j)\}$
$\pi \leftarrow \{(i, j) \mapsto b : (i, j) \in [\![\mathcal{N}]\!] \setminus P, b \in \mathcal{S}[i, j]\}$;
for $(i, j) \in [\![\mathcal{N}]\!] \setminus P$ **do**
  $\mathcal{N}' \leftarrow$ PathConsistency($\mathcal{N}^{\pi}_{V\,[i,j]/\mathsf{B}}$);
  if $\mathcal{N}' \subseteq \mathcal{N}$ **then**
    $\pi \leftarrow \pi|_{\mathsf{dom}(\pi)\setminus\{(i,j)\}}$;
return π

Algorithm 3 FindOnePS_3($\mathcal{N}, \mathcal{S}$).

in : A QCN $\mathcal{N} = (V, C)$ and a complete scenario $\mathcal{S}$ of $\mathcal{N}$
out : A prime scenario π that covers $\mathcal{S}$
$\mathcal{N}' \leftarrow \mathcal{N}$;
$min \leftarrow 1$;
$max \leftarrow n$;
while $min \neq max$ **do**
  $v \leftarrow (max + min)/2$;
  $\mathcal{N}'' \leftarrow$ PathConsistency($\mathcal{N}'_{[i_1,j_1]/\mathcal{S}[i_1,j_1],\ldots,[i_v,j_v]/\mathcal{S}[i_v,j_v]}$);
  if $\mathcal{N}'' \subseteq \mathcal{N}$ **then**
    $max \leftarrow v$;
  else
    $min \leftarrow v + 1$;
    $\mathcal{N}' \leftarrow \mathcal{N}''$;
$\pi \leftarrow \{(i_k, j_k) \mapsto b_k : 1 \leq k \leq min, b \in \mathcal{S}[i, j]\}$;
for $k \in 1, \ldots, min$ **do**
  if *PathConsistency*($\mathcal{N}^{\pi}_{V\,[i_k,j_k]/B}$) $\subseteq \mathcal{N}$ **then**
    $\pi \leftarrow \pi|_{\mathsf{dom}(\pi)\setminus\{(i,j)\}}$;
return π

Algorithm 1 starts by obtaining a CQA from a given complete scenario: its domain corresponds to the set of non-universal constraints in the original QCN. It then iterates over this CQA, applying path consistency to determine if the domain can be reduced.

Algorithm 2 begins by constructing a more compact CQA compared to Algorithm 1. It achieves this by using a while loop, which adds a constraint at each iteration using the given complete scenario until it reaches a CQA. Then, similarly to algorithm 1, it uses a for loop to compute a prime scenario from the obtained CQA.

Algorithm 3 is described by fixing $\{(i, j) \in [\![\mathcal{N}]\!] : \mathcal{N}[i, j] \neq \mathsf{B}\} = \{(i_1, j_1), \ldots, (i_n, j_n)\}$. Similar to Algorithm 2, it starts by computing a CQA and then utilizes a for loop to obtain a prime scenario from the computed CQA. However, unlike Algorithm 2, Algorithm 3 incorporates a dichotomic search to compute a CQA, which might enable it to perform the search more efficiently.

Algorithm 4 COMPUTEPSCOVER($\mathcal{N}, \mathcal{N}', \pi$).

in : Two QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}'(V, C')$, and q-assignment π over V
out : A PS cover of $\mathcal{N}$ by assigning $\mathcal{N}_\top$ to $\mathcal{N}'$ and $\emptyset$ to π

1 $\mathcal{N}'' \leftarrow$ PATHCONSISTENCY($\mathcal{N}'$);
2 **if** $\exists(i,j) \in [\![\mathcal{N}]\!] \setminus \mathit{dom}(\pi), \mathcal{N}''[i,j] \cap \mathcal{N}[i,j] = \emptyset$ **then**
3 | **return** $\emptyset$;
4 **if** $\mathcal{N}'' \subseteq \mathcal{N}$ **then**
5 | **return** $\{$*FINDONEPS*$(\mathcal{N}, \pi)\}$;
6 Let $(i,j) \in [\![\mathcal{N}]\!] \setminus \mathsf{dom}(\pi)$ s.t. $\mathcal{N}''[i,j] \not\subseteq \mathcal{N}[i,j]$;
7 $R \leftarrow \emptyset$;
8 **for** $b \in \mathcal{N}''[i,j] \cap \mathcal{N}[i,j]$ **do**
9 | $R \leftarrow R \cup \{$FINDPSCOVER$(\mathcal{N}, \mathcal{N}''_{[i,j]/b}, \pi \cup \{(i,j) \mapsto b\})\}$;
10 **return** R

By employing three distinct algorithms, we can benefit from the advantages and the strength of each approach. Our experiments have revealed that these algorithms exhibit varying levels of accuracy and efficiency for specific instances. Note that the considered approaches are similar to some approaches used in propositional logic for computing prime implicants, prime implicates, and minimal unsatisfiable cores (e.g., see [29, 18, 8]).

4 Prime Scenario Cover

Prime implicant cover is a key knowledge compilation concept in the realm of Boolean circuit design, as it allows us to simplify complex Boolean functions: a function is represented as a disjunction of prime implicants that cover all its models. In this section, we investigate a similar concept in QSTR, called prime scenario cover.

We define a *prime scenario cover* of a QCN $\mathcal{N}$ as any set $\mathcal{C}$ of prime scenarios of $\mathcal{N}$ such that each complete scenario of $\mathcal{N}$ is covered by at least one element of $\mathcal{C}$.

A prime scenario cover provides a simplified representation of the original QCN. It can also be regarded as a compact representation of all complete scenarios of the initial QCN.

Computing A Prime Scenario Cover

We propose two distinct approaches for computing a prime scenario cover of a given QCN. The first approach considers every branch of the search tree to cover all scenarios, while the second is based on an encoding in the SAT problem.

Constraint-based Approach

Algorithm 4 generates a prime scenario cover by recursively exploring the search tree and including a prime scenario for each found CQA. To obtain a prime scenario cover, we need to invoke COMPUTEPSCOVER by assigning $\mathcal{N}_\top$ to $\mathcal{N}'$ and $\emptyset$ to π. The code in Lines 2–3 ensures that search-subtrees without any CQA are not considered. The code in Lines 4–5 generates a prime scenario from a found CQA using one of the approaches described previously. Finally, the code in Lines 6–9 selects a constraint in the current QCN to continue exploring the search tree by making new decisions.

Algorithm 5 ComputePSCover($\mathcal{N}$).

in : A QCN $\mathcal{N} = (V, C)$
out : A PS cover $\mathcal{C}$ of $\mathcal{N}$
1 $\mathcal{C} \leftarrow \emptyset$;
2 $\Phi \leftarrow \mathsf{SATEnc}(\mathcal{N})$;
3 **while** $SAT(\Phi)$ **do**
4 $\quad\pi \leftarrow$ FindOnePS($\mathcal{N}, \mathcal{S}_\omega$);
5 $\quad\mathcal{C} \leftarrow \mathcal{C} \cup \{\pi\}$;
6 $\quad\Phi \leftarrow \phi \wedge \bigvee_{(i,j)\in \mathsf{dom}(\pi)} \neg p_{ij}^{\pi(i,j)}$
7 **return** $\mathcal{C}$

SAT-based Approach

To define our second algorithm, we use a SAT encoding of the consistency problem [19, 35]. For every $(i, j) \in [\![\mathcal{N}]\!]$ and every $b \in \mathsf{B}$, we associate a distinct propositional variable p_{ij}^b. Then, we define the encoding $\mathsf{SATEnc}(\mathcal{N})$ as follows: (1) $\sum_{b\in C(i,j)} p_{ij}^b = 1$ for each $(i, j) \in [\![\mathcal{N}]\!]$; and (2) $\bigwedge_{\substack{b_1\in C(i,j)\\ b_2\in C(j,k)}} (p_{ij}^{b_1} \wedge p_{jk}^{b_2} \rightarrow \bigvee_{b_3\in(b_1\diamond b_2)\cap C(i,k)} p_{ik}^{b_3})$ for every $(i, j), (j, k) \in [\![\mathcal{N}]\!]$.

Note that the sum constraints in Formula (1) can be linearly encoded as CNF formulas in several ways (e.g., see [31]).

For every model ω of $\mathsf{SATEnc}(\mathcal{N})$, the associated complete scenario of $\mathcal{N}$, denoted $\mathcal{S}_\omega$, is defined as follows: for every $(i, j) \in [\![\mathcal{N}]\!]$, $\mathcal{S}_\omega[i, j] = \{b : \omega(p_{ij}^b) = 1\}$.

Algorithm 5 allows us to compute a prime scenario cover by ensuring that each newly found prime scenario covers at least one complete scenario that is not covered by the previously obtained prime scenarios. Indeed, in each iteration of the while loop, the computed complete scenario is not covered by the prime scenarios found in the previous iterations, thanks to the addition of blocking clauses in Line 6.

5 Minimum-Size Prime Scenarios

The minimum-size prime scenarios are those that have the smallest possible domains. We think that, like minimum-size prime implicants, minimum-size prime scenarios can be applied in various contexts. In this section, after describing our algorithm for computing minimum-size prime scenarios, we introduce a novel application by showing that these prime scenarios can be useful for analyzing and reasoning about robustness. Specifically, they can help us to define a robustness measure that provides insights into the number of critical constraints.

Computing a Minimum-Size Prime Scenario: PMaxSAT-based Approach

Given two QCNs $\mathcal{N}$ and $\mathcal{N}'$ over the same set of variables V, we use $\mathsf{comp}(\mathcal{N}, \mathcal{N}')$ to denote the set $\{(i, j) \mapsto b : (i, j) \in [\![\mathcal{N}]\!]$ and $b \in \mathcal{N}[i, j] \setminus \mathcal{N}'[i, j]\}$.

A *hitting set* is a subset of a collection of sets that intersects with every element in the collection. A hitting set is said to be *minimal* if it cannot be reduced in size without ceasing to be a hitting set.

The following theorem shows that all prime scenarios can be obtained from the minimal hitting sets of collections of sets built from the counter-scenarios.

▶ **Theorem 6.** *A q-assignment π is a prime scenario of $\mathcal{N}$ iff π is a minimal hitting set of $\mathcal{H} = \{\mathsf{comp}(\mathcal{N}, \mathcal{N}') : \mathcal{N}' \in \mathsf{CounterS}(\mathcal{N})\}$ and $\mathcal{N}_V^\pi$ is consistent.*

Algorithm 6 $\textsc{MinimumSizePS}(\mathcal{N})$.

in : A QCN $\mathcal{N} = (V, C)$
out : A minimum-size prime scenario of $\mathcal{N}$
Let $\mathcal{S}_0$ an arbitrary counter-scenario of $\mathcal{N}$;
$\mathcal{H} \leftarrow \{\mathsf{comp}(\mathcal{N}, \mathcal{S}_0)\}$;
while *true* **do**
  $\pi \leftarrow \textsc{getHS}(\mathsf{MaxSATMH}(\mathcal{H}, \mathcal{N}))$;
  $\mathcal{N}' \leftarrow \textsc{PathConsistency}(\mathcal{N}^{\pi}_{V})$;
  if $\mathcal{N}' \subseteq \mathcal{N}$ **then**
    return π
  Let $\mathcal{S}$ be an arbitrary scenario of $\mathcal{N}'$ where $\mathcal{S}[i, j] \not\subseteq \mathcal{N}[i, j]$ for some $(i, j) \in [\![\mathcal{N}]\!]$;
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathsf{comp}(\mathcal{N}, \mathcal{S})\}$;

Proof. First, we prove the "if" part. Let π be a q-assignment such that $\mathcal{N}^{\pi}_{V}$ is consistent and π is a minimal hitting set of $\mathcal{H}$. We assume for the sake of contradiction that $\mathcal{N}^{\pi}_{V}$ is satisfied by a counter-scenario $\mathcal{N}'$ of $\mathcal{N}$. This implies that $\pi \cap \mathsf{comp}(\mathcal{N}, \mathcal{N}') = \emptyset$. However, this contradicts the assumption that π is a hitting set of $\mathcal{H}$. Therefore, π must be a CQA of $\mathcal{N}$. To prove that π is a prime scenario, we must show that its domain is minimal w.r.t. set inclusion. This follows directly from the fact that π is a minimal hitting set of $\mathcal{H}$. Indeed, any proper subset π' of π does not hit at least one element of $\mathcal{H}$, which means that $\mathcal{N}^{\pi'}_{V}$ is satisfied by at least one counter-scenario of $\mathcal{N}$. Consequently, π is a prime scenario of $\mathcal{N}$.

Now, we move to the "only if" part. Let π be a prime scenario of $\mathcal{N}$. Suppose that there is counter-scenario $\mathcal{N}'$ of $\mathcal{N}$ s.t. $\pi \cap \mathsf{comp}(\mathcal{N}, \mathcal{N}') = \emptyset$. Thus $\mathcal{N}'$ is a complete scenario of $\mathcal{N}^{\pi}_{V}$, which leads to a contradiction. Therefore, π is a hitting set of $\mathcal{H}$. Just as in the "if" part, the minimality of π as a hitting set is implied by its minimality as a CQA. ◀

To some extent, Theorem 6 is similar to the minimal hitting set duality between prime implicants and prime implicates in the case of propositional logic [24, 27, 20].

Our algorithm generates candidate solutions by utilizing a Partial MaxSAT encoding to compute specific minimal hitting sets. We denote this encoding by $\mathsf{MaxSATMH}(\mathcal{H}', \mathcal{N})$, where $\mathcal{N} = (V, C)$ is a QCN and $\mathcal{H}' \subseteq \{\mathsf{comp}(\mathcal{N}, \mathcal{N}') : \mathcal{N}' \in \mathsf{CounterS}(\mathcal{N})\}$. In addition to the variables used to define the $\mathsf{SATEnc}(\mathcal{N})$ encoding, described in Section 4, we associate a distinct propositional variable q^b_{ij} with every $(i, j) \mapsto b \in \bigcup \mathcal{H}'$. The hard part of $\mathsf{MaxSATMH}(\mathcal{H}', \mathcal{N})$ corresponds to the conjunction of $\mathsf{SATEnc}(\mathcal{N})$ and the following formulas: (1) $\bigvee_{(i,j)\mapsto b\in e} q^b_{ij}$ for each $e \in \mathcal{H}$; and (2) $q^b_{ij} \to p^b_{ij}$ for each $(i, j) \mapsto b \in \bigcup \mathcal{H}'$.

Formula (1) guarantees that each solution of the encoding hits all elements of $\mathcal{H}'$, and Formula (2) forces the truth values of the variables representing a complete scenario of $\mathcal{N}$ to match those of the variables of the form q^b_{ij}.

The soft part of $\mathsf{MaxSATMH}(\mathcal{H}', \mathcal{N})$ corresponds to the set of unit clauses $\{\neg q^b_{ij} : (i, j) \mapsto b \in \bigcup \mathcal{H}'\}$. This allows us to minimize the size of the hitting set.

Given a solution ω of $\mathsf{MaxSATMH}(\mathcal{H}', \mathcal{N})$, its associated q-assignment is $\pi_\omega = \{(i, j) \mapsto b \in \bigcup \mathcal{H}' : \omega(q^b_{ij}) = 1\}$. Clearly, π_ω is one of the smallest hitting sets of $\mathcal{H}'$ such that $\mathcal{N}^{\pi_\omega}_{V}$ is consistent and covers a scenario of $\mathcal{N}$.

Theorem 6 shows that every minimum-size prime scenario π of $\mathcal{N}$ is a minimum-size hitting set of $\mathcal{H} = \{\mathsf{comp}(\mathcal{N}, \mathcal{N}') : \mathcal{N}' \in \mathsf{CounterS}(\mathcal{N})\}$ where (1) $\mathcal{N}^{\pi}_{V}$ is consistent, and (2) every complete scenario of $\mathcal{N}^{\pi}_{V}$ is a complete scenario of $\mathcal{N}$. Consequently, if π is one of the smallest hitting sets of a subset $\mathcal{H}' \subseteq \mathcal{H}$ that satisfies Properties 1 and 2, then π is a minimum-size prime scenario of $\mathcal{N}$. This is because every hitting set of $\mathcal{H}$ is also a hitting set of $\mathcal{H}'$. Algorithm 6 uses this property to generate a minimum-size prime scenario. In each iteration of the while loop, Algorithm 6 employs the encoding $\mathsf{MaxSATMH}(\mathcal{H}', \mathcal{N})$ to compute π, one of the smallest hitting sets that satisfies Property 1 (Line 4). It then uses path consistency to check whether π satisfies also Property 2 (Lines 5–6). If π satisfies both properties, then π is a minimum-size prime scenario and is returned; otherwise, the algorithm adds an element obtained from a new counter-scenario of $\mathcal{N}$ to the collection of sets $\mathcal{H}$. In the worst case, all counter-scenarios of $\mathcal{N}$ will be considered in $\mathcal{H}$, and this necessarily allows the algorithm to obtain a minimum-size prime scenario.

Algorithm 6 shares some similarities with the approach used in [6] for solving the MaxSAT problem. This approach leverages the duality between minimal correction subsets and minimal unsatisfiable subsets.

An Application of Minimum-Size Prime Scenarios: Robustness Measure

Now, we demonstrate one possible use of minimum-size prime scenarios in reasoning about robustness in QCNs, cf. [32] and [34]. With respect to our terminology here, QCN robustness refers to the ability of a QCN to withstand *perturbations*, i.e., eliminations of base relations, without needing to transform counter-scenarios into scenarios: the scenarios that result after perturbation are also scenarios of the original QCN. In other words, a robust QCN can maintain its consistency when facing perturbations. Although certain robustness notions have been studied in [32] and [34], robustness measures that can be used to compare different QCNs with one another have not been formalized or introduced; in fact, those notions only compare the different scenarios (or refined QCNs) with one another of a single QCN.

We define a robustness measure as a function from the set of QCNs to positive real numbers. Our robustness measure, denoted R_{PS}, is defined as follows:

$$R_{PS}(\mathcal{N}) = max\{|[\![\mathcal{N}]\!]| - |\mathsf{dom}(\pi)| : \pi \in \mathsf{PSes}(\mathcal{N})\}$$

where $max\ \emptyset = 0$. For consistent QCNs, we clearly have $R_{PS}(\mathcal{N}) = |[\![\mathcal{N}]\!]| - min\{|\mathsf{dom}(\pi)| : \pi \in \mathsf{PSes}(\mathcal{N})\}$; It follows that R_{PS} can be computed from any minimum-size prime scenario.

Our measure captures the fact that the robustness increases by decreasing the number of the constraints that we need to instantiate to get a complete scenario of the given QCN.

To formally establish the suitability of our robustness measure, we present a result that lists interesting properties that can be considered as necessary for any robustness measure.

▶ **Proposition 7.** *The following properties are satisfied:*
1. *for any inconsistent QCN $\mathcal{N}$, $R_{PS}(\mathcal{N}) = 0$;*
2. *$R_{PS}(\mathcal{N}_\top) = |[\![\mathcal{N}_\top]\!]|$;*
3. *for all two QCNs $\mathcal{N}$ and $\mathcal{N}'$ with $\mathsf{Scenarios}(\mathcal{N}) = \mathsf{Scenarios}(\mathcal{N}')$, $R_{PS}(\mathcal{N}) = R_{PS}(\mathcal{N}')$;*
4. *for all two QCNs $\mathcal{N}$ and $\mathcal{N}'$ with $\mathsf{Scenarios}(\mathcal{N}) \subseteq \mathsf{Scenarios}(\mathcal{N}')$, $R_{PS}(\mathcal{N}) \leq R_{PS}(\mathcal{N}')$.*

Proof. Property 1 holds since every inconsistent QCN does not admit any prime scenario. Property 2 follows from the fact that $\pi = \emptyset$ is a prime scenario of $\mathcal{N}_\top$. The fact that the QCNs having the same complete scenarios have also the same prime scenarios leads to Property 3. Property 4 stems from the observation that $\mathsf{PSes}(\mathcal{N}) \subseteq \mathsf{PSes}(\mathcal{N}')$ holds when $\mathsf{Scenarios}(\mathcal{N}) \subseteq \mathsf{Scenarios}(\mathcal{N}')$. ◀

The first two properties state that the minimum robustness value is associated with inconsistent QCNs, while the maximum value corresponds to QCNs where all relations are trivial, viz., $\mathcal{N}_\top$. The third property ensures that identical complete scenarios lead to the same robustness value. The last property guarantees that the robustness value does not decrease as more complete scenarios are considered.

Table 1 Assessing the performance of obtaining (minimum) prime scenarios, the format being $\frac{\text{min} \mid \text{avg.}(\mu) \mid \text{max prime index}}{\text{min} \mid \text{avg.}(\mu) \mid \text{max \# of oracle calls}}$(# of timeouts); a timeout occurs after 1 200*s*, and it is important to note that the oracle calls for the FindOnePS variants concern the application of path consistency, whereas the ones for MinimumSizePS the solving of a Partial MaxSAT instance.

d	FindOnePS_1	FindOnePS_2	FindOnePS_3	MinimumSizePS
9	$\frac{0.2 \mid 0.3 \mid 0.4}{45 \mid 45.0 \mid 45}$	$\frac{0.2 \mid \mathbf{0.29} \mid 0.36}{26 \mid \mathbf{38.24} \mid 52}$	$\frac{0.2 \mid 0.3 \mid 0.38}{34 \mid 45.11 \mid 50}$	$\frac{0.2 \mid 0.26 \mid 0.31}{0.9k \mid 2.2k \mid 4.2k}$(34)
8	$\frac{0.23 \mid 0.34 \mid 0.45}{40 \mid 40.0 \mid 40}$	$\frac{0.23 \mid \mathbf{0.33} \mid 0.43}{28 \mid \mathbf{39.02} \mid 54}$	$\frac{0.23 \mid 0.34 \mid 0.45}{23 \mid 41.03 \mid 45}$	$\frac{0.23 \mid 0.29 \mid 0.35}{1.5k \mid 2.9k \mid 5.7k}$(45)
7	$\frac{0.29 \mid 0.4 \mid 0.66}{35 \mid \mathbf{35.0} \mid 35}$	$\frac{0.29 \mid \mathbf{0.39} \mid 0.66}{26 \mid 39.98 \mid 60}$	$\frac{0.29 \mid 0.4 \mid 0.57}{27 \mid 37.39 \mid 40}$	$\frac{0.26 \mid 0.33 \mid 0.46}{1.7k \mid 3.4k \mid 5.3k}$(64)
6	$\frac{0.3 \mid 0.47 \mid 0.6}{30 \mid \mathbf{30.0} \mid 30}$	$\frac{0.3 \mid 0.46 \mid 0.6}{26 \mid 39.60 \mid 54}$	$\frac{0.33 \mid \mathbf{0.46} \mid 0.63}{21 \mid 32.89 \mid 34}$	$\frac{0.3 \mid 0.38 \mid 0.47}{2.7k \mid 4.1k \mid 5.5k}$(85)
5	$\frac{0.4 \mid 0.57 \mid 0.76}{25 \mid \mathbf{25.0} \mid 25}$	$\frac{0.4 \mid 0.57 \mid 0.76}{28 \mid 37.92 \mid 46}$	$\frac{0.4 \mid \mathbf{0.57} \mid 0.8}{23 \mid 28.3 \mid 29}$	$\frac{0.36 \mid 0.45 \mid 0.56}{2.2k \mid 4.3 \mid 6.2k}$(88)
4	$\frac{0.5 \mid 0.69 \mid 0.85}{20 \mid \mathbf{20.0} \mid 20}$	$\frac{0.5 \mid \mathbf{0.69} \mid 0.85}{24 \mid 34.1 \mid 40}$	$\frac{0.5 \mid 0.7 \mid 0.9}{21 \mid 23.57 \mid 24}$	$\frac{0.45 \mid 0.52 \mid 0.55}{3.6k \mid 5.6k \mid 7.0k}$(97)
3	$\frac{0.67 \mid \mathbf{0.83} \mid 1.0}{15 \mid \mathbf{15.0} \mid 15}$	$\frac{0.67 \mid \mathbf{0.83} \mid 1.0}{22 \mid 28.14 \mid 30}$	$\frac{0.67 \mid 0.84 \mid 1.0}{16 \mid 17.96 \mid 18}$	$\frac{0.6 \mid 0.63 \mid 0.67}{5.0k \mid 5.0k \mid 5.0k}$(98)

Table 2 Assessing the performance of obtaining prime scenario covers, the format being avg. # of oracle calls; it is important to note that the oracle calls for ComputePSCover concern the application of path consistency, whereas the ones for ComputePSCover(SAT) the solving of a SAT instance, and that avg. cover size = avg. # of oracle calls of ComputePSCover(SAT) − 1 (each oracle call in line 3 of Algorithm 5 computes a prime scenario in the cover, minus the last one).

	$d = 9$	8	7	6	5	4	3
ComputePSCover	0.2*k*	0.3*k*	0.5*k*	1.0*k*	2.3*k*	3.0*k*	3.5*k*
ComputePSCover(SAT)	**16.05**	**25.04**	**56.21**	**0.1k**	**0.4k**	**0.7k**	**1.0k**

6 Experimentation

In this section, we perform a *preliminary* evaluation to assess the efficiency of our algorithms and, hence, also the difficulty of the introduced problems that they tackle. Our expectation is that: the FindOnePS variants should run really fast as they involve a number of path consistency applications that is linear to the number of constraints of a QCN, the ComputePSCover variants should run comparatively quite slower as they explore the search space of a QCN and mirror model counting algorithms, and the MinimumSizePS algorithm should be the slowest of all as it is not only dealing with finding a prime scenario for each of the exponentially many scenarios of a QCN, but one that is minimal too (there are many possibilities for a single scenario).

Dataset, Measures, & Setup

To be able to have results that are comparable between fast polytime methods (the FindOnePS variants) and methods for hard optimization problems (the MinimumSizePS algorithm), we consider QCNs of IA of 10 variables with a maximum of 2 base relations per non-universal constraint, for every avg. degree $d \in (9, 8, \ldots, 3)$ of their constraint graphs

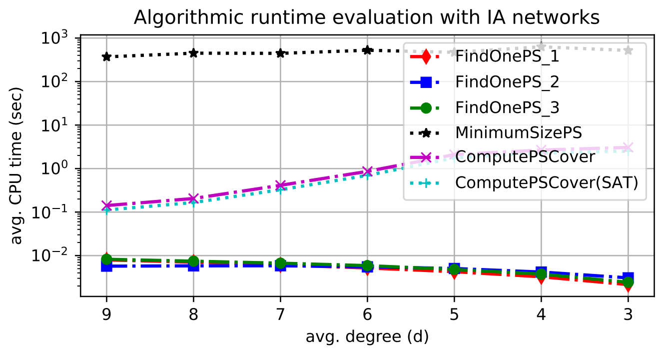


Figure 3 Assessing the runtime of our algorithms for the problems pertaining to prime scenarios.

(i.e., going from complete graphs to sparse ones). Specifically, we generate two arbitrary IA scenarios that we then proceed to unify; then, we create all the QCNs that result by considering one sub-graph of the initially complete constraint graph for every degree d in the aforementioned range, each with an avg. degree d. We consider 100 QCNs with an initially complete constraint graph, each yielding 6 more (sparser ones), hence a total of 700 QCNs. The size of the networks is relatively consistent with what has been used in the literature for similar optimization problems in order to present results that are as complete as possible (e.g., [3]), see also Table 1; in addition, a QCN of IA of n variables enumerates $O(2^{n \cdot \log n})$ scenarios (qualitative solutions) [12], which translates to roughly 10 billion scenarios in our case.

All of the used measures are clear and intuitive, with the exception of *prime index*: this is the ratio of the # of non-universal constraints in a prime scenario to the # of non-universal constraints in the original QCN and, thus, takes values in $(0, 1]$. Clearly, the denser the network, the more opportunities there are to obtain a low measure of this type.

For the experiments we used an Intel®Core®CPU i7-12700H @ 4.70GHz, 16 GB of RAM, and the Ubuntu Linux 22.04 LTS OS. All coding/running was done in Python 3.10.6; the code is available at: `https://seafile.lirmm.fr/d/9c0cbd2cd0954252ab96/`.

Results & Remarks

The results are shown in Tables 1 and 2 and Figure 3, and confirm our expectations; we detail as follows. Regarding (minimum) prime scenario computation, the polytime FindOnePS variants are extremely fast, and among those variants the simpler FindOnePS_1 has the best performance overall; in the case of computing a prime scenario that is also minimal, we can see that MinimumSizePS can reduce the min, avg., and max prime index values, but at a huge cost as the number of scenarios that this algorithm has to consider becomes detrimental to its runtime performance (see # of timeouts in Table 1 and runtime in Figure 3 in particular). Regarding prime scenario cover computation, the constraint-based and the SAT-based ComputePSCover algorithms perform very similarly, with the SAT variant, viz., ComputePSCover(SAT), performing better overall with respect to runtime performance (see Figure 3 in particular); here, we must note that we did not find any notable differences in the size of the covers that these algorithms computed (the same result applies to both, see the caption of Table 2), even though such differences may exist in general.

7 Conclusion and Perspectives

We introduced the novel notion of *prime scenario* to QSTR, which is analogous to that of prime implicant in the case of classical logic. In sum, we made five major contributions: first, we described three methods for computing one prime scenario; secondly, we presented two methods for computing a prime scenario cover, which is a set of prime scenarios that cover all the scenarios of a given QCN; thirdly, we proposed a method for computing a minimum-size prime scenario and, fourthly, demonstrated how this notion can be used to reason about robustness; and, fifthly, we experimentally evaluated all our algorithms and made our code available for any interested researcher to use. Our study opens up new perspectives by revealing previously unexplored ways to extend the notion of prime implicants to QSTR. Specifically, it sheds light on the possible use of prime scenarios to explain the decisions made by classifiers compiled into QCNs, in the same way as prime implicants [30, 9, 10, 11, 4], and opens new avenues for research in the field of knowledge compilation in the context of QSTR.

References

1 James F. Allen. Maintaining Knowledge about Temporal Intervals. *Commun. ACM*, 26:832–843, 1983.

2 Marco Cadoli and Francesco M. Donini. A Survey on Knowledge Compilation. *AI Commun.*, 10:137–150, 1997.

3 Jean-François Condotta, Issam Nouaouri, and Michael Sioutis. A SAT Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In *KR*, 2016.

4 Adnan Darwiche and Auguste Hirth. On the (Complete) Reasons Behind Decisions. *J. Log. Lang. Inf.*, 32:63–88, 2023.

5 Adnan Darwiche and Pierre Marquis. A Knowledge Compilation Map. *J. Artif. Intell. Res.*, 17:229–264, 2002.

6 Jessica Davies and Fahiem Bacchus. Solving MAXSAT by Solving a Sequence of Simpler SAT Instances. In *CP*, 2011.

7 Johan de Kleer, Alan K. Mackworth, and Raymond Reiter. Characterizing Diagnoses and Systems. *Artif. Intell.*, 56:197–222, 1992.

8 Fred Hemery, Christophe Lecoutre, Lakhdar Sais, and Frédéric Boussemart. Extracting MUCs from Constraint Networks. In *ECAI*, 2006.

9 Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. Abduction-Based Explanations for Machine Learning Models. In *AAAI*, 2019.

10 Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. On Relating Explanations and Adversarial Examples. In *NeurIPS*, 2019.

11 Yacine Izza and João Marques-Silva. On Explaining Random Forests with SAT. In *IJCAI*, 2021.

12 Peter Jonsson and Victor Lagerkvist. An initial study of time complexity in infinite-domain constraint satisfaction. *Artif. Intell.*, 245:115–133, 2017.

13 Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both. On redundant topological constraints. *Artif. Intell.*, 225:51–76, 2015.

14 Gerard Ligozat. Reasoning about cardinal directions. *J. Vis. Lang. Comput.*, 9:23–44, 1998. `doi:10.1006/jvlc.1997.9999`.

15 Gérard Ligozat. *Qualitative Spatial and Temporal Reasoning*. Iste Series. Wiley, 2011.

16 Zhiguo Long and Sanjiang Li. On Distributive Subalgebras of Qualitative Spatial and Temporal Calculi. In *COSIT*, 2015.

17 E. J. McCluskey Jr. Minimization of Boolean Functions*. *Bell System Technical Journal*, 35:1417–1444, 1956.

18 Luigi Palopoli, Fiora Pirri, and Clara Pizzuti. Algorithms for Selective Enumeration of Prime Implicants. *Artif. Intell.*, 111:41–72, 1999.

19 Duc Nghia Pham, John Thornton, and Abdul Sattar. Modelling and solving temporal reasoning as propositional satisfiability. *Artif. Intell.*, 172:1752–1782, 2008.

20 Alessandro Previti, Alexey Ignatiev, António Morgado, and João Marques-Silva. Prime Compilation of Non-Clausal Formulae. In *IJCAI*, 2015.

21 W. V. Quine. The Problem of Simplifying Truth Functions. *The American Mathematical Monthly*, 59:521–531, 1952.

22 W. V. Quine. A Way to Simplify Truth Functions. *The American Mathematical Monthly*, 62:627–631, 1955.

23 David A. Randell, Zhan Cui, and Anthony Cohn. A Spatial Logic Based on Regions and Connection. In *KR*, 1992.

24 Raymond Reiter. A Theory of Diagnosis from First Principles. *Artif. Intell.*, 32:57–95, 1987.

25 Jochen Renz and Gérard Ligozat. Weak Composition for Qualitative Spatial and Temporal Reasoning. In *CP*, 2005.

26 Jochen Renz and Bernhard Nebel. Qualitative Spatial Reasoning Using Constraint Calculi. In *Handbook of Spatial Logics*, pages 161–215. Springer, 2007.

27 Ron Rymon. An Se-Tree-Based Prime Implicant Generation Algorithm. *Ann. Math. Artif. Intell.*, 11:351–366, 1994.

28 Stefan Schlobach, Zhisheng Huang, Ronald Cornet, and Frank van Harmelen. Debugging Incoherent Terminologies. *J. Autom. Reason.*, 39:317–349, 2007.

29 Robert Schrag. Compilation for Critically Constrained Knowledge Bases. In *AAAI*, 1996.

30 Andy Shih, Arthur Choi, and Adnan Darwiche. A Symbolic Approach to Explaining Bayesian Network Classifiers. In *IJCAI*, 2018.

31 Carsten Sinz. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. In *CP*, 2005.

32 Michael Sioutis, Zhiguo Long, and Tomi Janhunen. On Robustness in Qualitative Constraint Networks. In *IJCAI*, 2020.

33 Marc Vilain, Henry Kautz, and Peter van Beek. Constraint Propagation Algorithms for Temporal Reasoning: A Revised Report. In *Readings in Qualitative Reasoning About Physical Systems*, pages 373–381. Morgan Kaufmann, 1990.

34 Jan Wehner, Michael Sioutis, and Diedrich Wolter. On Robust Vs Fast Solving of Qualitative Constraints. In *ICTAI*, 2021.

35 Matthias Westphal and Stefan Wölfl. Qualitative CSP, Finite CSP, and SAT: Comparing Methods for Qualitative Constraint-based Reasoning. In *IJCAI*, 2009.