# Dynamic Branching in Qualitative Constraint Networks via Counting Local Models

**Michael Sioutis**[1]
Faculty of Information Systems and Applied Computer Sciences, University of Bamberg, Germany
`https://msioutis.gitlab.io/`
michail.sioutis@uni-bamberg.de

**Diedrich Wolter**
Faculty of Information Systems and Applied Computer Sciences, University of Bamberg, Germany
`https://www.uni-bamberg.de/en/sme/team/diedrich-wolter/`
diedrich.wolter@uni-bamberg.de

## Abstract

We introduce and evaluate *dynamic branching* strategies for solving Qualitative Constraint Networks (QCNs), which are networks that are mostly used to represent and reason about spatial and temporal information via the use of simple qualitative relations, e.g., a constraint can be "Task $A$ is scheduled *after or during* Task $C$". In qualitative constraint-based reasoning, the state-of-the-art approach to tackle a given QCN consists in employing a backtracking algorithm, where the branching decisions during search are governed by the restrictiveness of the possible relations for a given constraint (e.g., *after* can be more restrictive than *during*). In the literature, that restrictiveness is defined a priori by means of static weights that are precomputed and associated with the relations of a given calculus, without any regard to the particulars of a given network instance of that calculus, such as its structure. In this paper, we address this limitation by proposing heuristics that dynamically associate a weight with a relation, based on the *count* of *local models* (or *local scenarios*) that the relation is involved with in a given QCN; these models are local in that they focus on triples of variables instead of the entire QCN. Therefore, our approach is adaptive and seeks to make branching decisions that preserve most of the solutions by determining what proportion of local solutions agree with that decision. Experimental results with a *random* and a *structured* dataset of QCNs of Interval Algebra show that it is possible to achieve up to 5 times better performance for structured instances, whilst maintaining non-negligible gains of around 20% for random ones.

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

**Keywords and phrases** Qualitative constraints, spatial and temporal reasoning, counting local models, dynamic branching, adaptive algorithm

**Digital Object Identifier** 10.4230/LIPIcs.TIME.2020.12

## 1 Introduction

Qualitative Spatial and Temporal Reasoning (QSTR) is a major field of study in AI that deals with the fundamental cognitive concepts of space and time in a human-like manner, via simple qualitative constraint languages [18, 8]. Such languages consist of abstract, qualitative, expressions like *inside*, *before*, or *north of* to spatially or temporally relate two or more objects to one another, without involving any quantitative information. Thus, QSTR offers tools for efficiently automating common-sense spatio-temporal reasoning and, hence, further boosts research to a plethora of application areas and domains that deal with spatio-temporal

[1] Corresponding author.

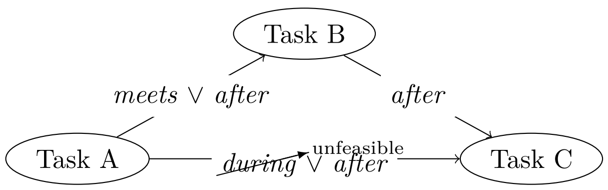


**Figure 1** The static weighting scheme in the literature dictates that relation *during* is less restrictive than relation *after* in general for the IA calculus and, hence, *during* should be preferred over *after* in branching decisions [39, Figure 9], but in the above simplified QCN *during* cannot appear in any solution; such schemes are defined for other calculi as well [13].

information, such as cognitive robotics [10], deep learning [17], visual explanation [37] and sensemaking [36], semantic question-answering [35], qualitative simulation [5], modal logic [21, 3, 20, 16, 11], temporal diagnosis [12], and stream reasoning [6, 14].

Qualitative spatial or temporal information may be modeled as a *Qualitative Constraint Network* (QCN), which is a network where the vertices correspond to spatial or temporal entities, and the arcs are labeled with qualitative spatial or temporal relations respectively. For instance $x \leq y$ can be a temporal QCN over $\mathbb{Z}$. Given a QCN $\mathcal{N}$, the literature is particularly interested in its *satisfiability problem*, which is the problem of deciding if there exists a spatial or temporal interpretation of the variables of $\mathcal{N}$ that satisfies its constraints, viz, a *solution* of $\mathcal{N}$. For instance, $x = 0 \wedge y = 1$ is one of the infinitely many solutions of the aforementioned QCN, and $x < y$ is the corresponding *scenario* that concisely represents all the cases where $x$ is assigned a lesser value than $y$. In general, for most widely-adopted qualitative calculi the satisfiability problem is NP-complete [9]. In the sequel, we will be using Interval Algebra (IA) [1] as an illustrative example of a qualitative calculus.

**Motivation & Contribution.** The state-of-the-art constraint-based approach for tackling a given QCN consists in employing a backtracking algorithm, where each branching decision during search is guided by the restrictiveness of the possible relations for a given constraint. Currently, that restrictiveness is defined a priori by means of entirely precomputed static weights that are associated with the relations of a given calculus. That static strategy has two major problems: it assumes a uniform use of relations in QCNs (as weights are computed by equally considering all the relations of a calculus); and it does not exploit any structure that may exist in QCNs (a relation that is used to form more than one constraints in a given QCN, which is typically the case, may exhibit different levels of restrictiveness among those constraints). A simple example of how this scheme can be problematic is detailed in Figure 1. In this paper, we address this limitation by proposing a *dynamic branching* mechanism via heuristics that dynamically associate a weight with a relation during search, based on the *count* of *local models*, i.e., scenarios pertaining to triples of variables, that the relation is involved with in a given QCN. This makes our approach similar to a counting-based one for CSPs [24], as it too is adaptive and it too seeks to make branching decisions that preserve most of the solutions by determining what proportion of local solutions agree with that decision. Further inspiration was drawn from a recent work in [32], where it was observed that a scenario of a QCN may often be constructed collectively by relations that appear in many scenarions individually, i.e., a scenario of a QCN may often be constructed by selecting the most popular relation for each constraint. Finally, through an evaluation with a random and a structured dataset of QCNs of IA, we show that we may achieve up to 5 times better performance for structured instances, and gains of about 20% for random ones.

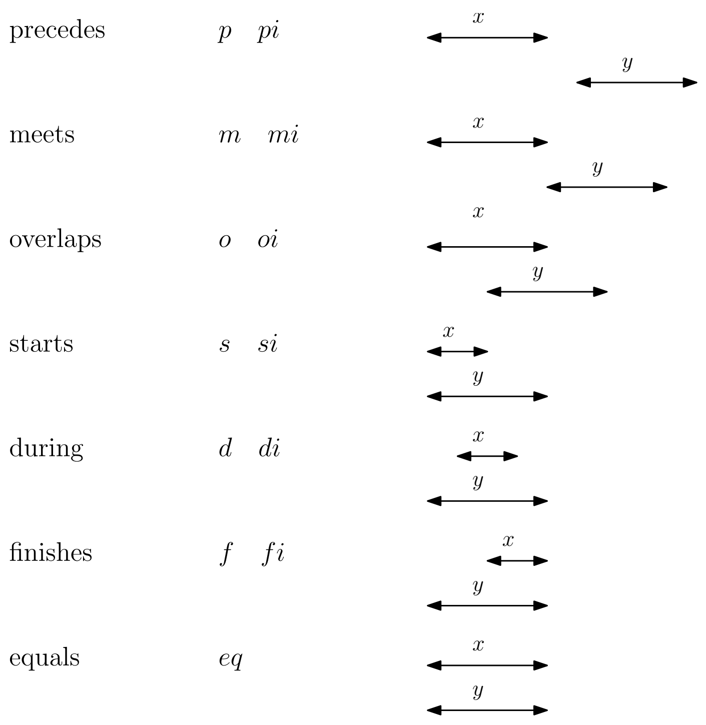


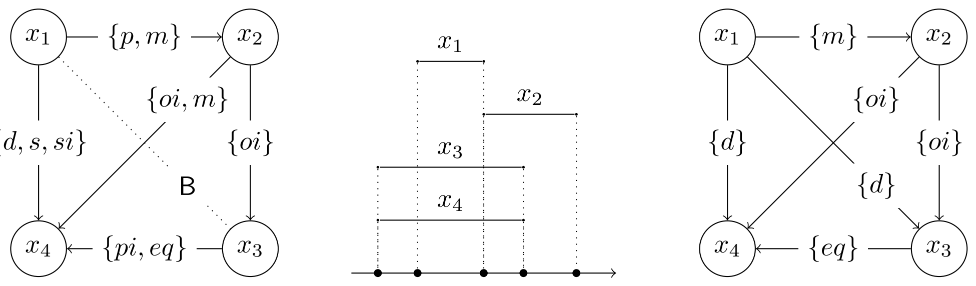
**Figure 2** The base relations of IA; $\cdot i$ denotes the converse of $\cdot$.

The rest of the paper is organized as follows. In Section 2 we give some preliminaries on QSTR. Next, in Section 3 we propose our dynamic approach, discuss some dynamic heuristics that are used internally, and present the related algorithms. Then, in Section 4 we evaluate our approach with random and structured QCNs of IA and comment on the outcome. Finally, in Section 5 we draw some conclusive remarks and give directions for future work.

## 2 Preliminaries

A binary qualitative spatial or temporal constraint language, is based on a finite set $\mathsf{B}$ of *jointly exhaustive and pairwise disjoint* relations, called the set of *base relations* [19], that is defined over an infinite domain $\mathsf{D}$. The base relations of a particular qualitative constraint language can be used to represent the definite knowledge between any two of its entities with respect to the level of granularity provided by the domain $\mathsf{D}$. The set $\mathsf{B}$ contains the identity relation $\mathsf{Id}$, 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, $2^{\mathsf{B}}$ represents the total set of relations. The set $2^{\mathsf{B}}$ is equipped with the usual set-theoretic operations of union and intersection, the converse operation, and the *weak composition* operation denoted by the symbol $\diamond$ [19]. 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., strongest) 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), introduced by Allen [1]. IA considers time intervals (as temporal entities) and the set of base relations $\mathsf{B} = \{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$ to



**(a)** A satisfiable QCN $\mathcal{N}$. **(b)** A solution $\sigma$ of $\mathcal{N}$. **(c)** A scenario $\mathcal{S}$ of $\mathcal{N}$

**Figure 3** Figurative examples of QCN terminology using IA.

encode knowledge about the temporal relations between intervals on the timeline, as depicted in Figure 2. Specifically, each base relation represents a particular ordering of the four endpoints of two intervals on the timeline, and *eq* is the identity relation Id.

Notably, most of the well-known and well-studied qualitative constraint languages, such as Interval Algebra [1] and RCC8 [25], are in fact *relation algebras* [9].

The problem of representing and reasoning about qualitative spatial or temporal information can be modeled as a *qualitative constraint network*, defined as follows:

▶ **Definition 1.** A *qualitative constraint network* (QCN) is a tuple $(V, C)$ where:

- $V = \{v_1, \ldots, v_n\}$ is a non-empty finite set of variables, each representing an entity of an infinite domain D;
- and $C$ is a mapping $C : V \times V \to 2^{\mathsf{B}}$ such that $C(v, v) = \{\mathsf{Id}\}$ for all $v \in V$ and $C(v, v') = (C(v', v))^{-1}$ for all $v, v' \in V$, where $\bigcup \mathsf{B} = \mathsf{D} \times \mathsf{D}$.

An example of a QCN of IA is shown in Figure 3a; for clarity, converse relations as well as Id loops are not mentioned or shown in the figure.

▶ **Definition 2.** Let $\mathcal{N} = (V, C)$ be a QCN, then:

- a *solution* of $\mathcal{N}$ is a mapping $\sigma : V \to \mathsf{D}$ such that $\forall (u, v) \in V \times V$, $\exists b \in C(u, v)$ such that $(\sigma(u), \sigma(v)) \in b$ (see Figure 3b);
- $\mathcal{N}$ is *satisfiable* 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 $C'(u, v) \subseteq C(u, v)$ $\forall u, v \in V$; if in addition $\exists u, v \in V$ such that $C'(u, v) \subset C(u, v)$, then $\mathcal{N}' \subset \mathcal{N}$;
- $\mathcal{N}$ is *atomic* iff $\forall v, v' \in V$, $C(v, v')$ is a *singleton relation*, i.e., a relation $\{b\}$ with $b \in \mathsf{B}$;
- a *scenario* $\mathcal{S}$ of $\mathcal{N}$ is an atomic satisfiable sub-QCN of $\mathcal{N}$ (see Figure 3c);
- the *constraint graph* of $\mathcal{N}$ is the graph $(V, E)$ where $\{u, v\} \in E$ iff $C(u, v) \neq \mathsf{B}$ and $u \neq v$;
- $\mathcal{N}$ is *trivially inconsistent*, denoted by $\emptyset \in \mathcal{N}$, iff $\exists v, v' \in V$ such that $C(v, v') = \emptyset$;
- $\mathcal{N}$ is the *empty* QCN on $V$, denoted by $\bot^V$, iff $C(u, v) = \emptyset$ for all $u, v \in V$.

Given a QCN $\mathcal{N} = (V, C)$ and $v, v' \in V$, we introduce the following operation that substitutes $C(v, v')$ with a relation $r \in 2^{\mathsf{B}}$ to produce a new, modified, QCN: $\mathcal{N}_{[v,v']/r}$ with $r \in 2^{\mathsf{B}}$ yields the QCN $\mathcal{N}' = (V, C')$, where $C'(v, v') = r$, $C'(v', v) = r^{-1}$ and $C'(u, u') = C(u, u')$ $\forall (u, u') \in (V \times V) \setminus \{(v, v'), (v', v)\}$.

We recall the definition of $\overset{\diamond}{_G}$-consistency [4] (cf [27]), which entails consistency for all triples of variables in a QCN that form triangles in an accompanying graph $G$, and is a basic and widely-used local consistency for reasoning with QCNs.

▶ **Definition 1.** *Given a* QCN $\mathcal{N} = (V, C)$ *and a graph* $G = (V, E)$*,* $\mathcal{N}$ *is said to be* $\overset{\diamond}{_G}$-consistent *iff* $\forall \{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$ *we have that* $C(v_i, v_j) \subseteq C(v_i, v_k) \diamond C(v_k, v_j)$*.*

We note here that if $G$ is complete, $\diamond_G$-consistency becomes identical to $\diamond$-consistency [27], and, hence, $\diamond$-consistency is a special case of $\diamond_G$-consistency. In the sequel, given a QCN $\mathcal{N} = (V, C)$ of some calculus and a graph $G = (V, E)$, we assume that $\diamond_G(\mathcal{N})$ is computable. This assumption holds for most widely-adopted qualitative calculi [9].

## 3 Approach

In qualitative constraint-based reasoning, the state-of-the-art approach to check the satisfiability of a given QCN $\mathcal{N}$, consists in splitting every relation $r$ that forms a constraint between two variables in $\mathcal{N}$ into a subrelation $r' \subseteq r$ that belongs to a set of relations $\mathcal{A}$ over which the QCN becomes tractable [29]. In particular, for most widely-adopted qualitative calculi [9], such *split* sets are either known or readily available [26], and tractability is then achieved via the use of some local consistency in backtracking fashion; after every refinement of a relation $r$ into a subrelation $r'$, the local consistency is enforced to know whether the refinement is valid or backtracking should occur and another subrelation should be chosen at an earlier point [29, Section 2]. One of the most essential and widely-used such local consistencies is $\diamond_G$-consistency, where $G$ is either the complete graph on the variables of $\mathcal{N}$ [27], or a *triangulation* (*chordal completion*) of the constraint graph of $\mathcal{N}$ [4].[2]

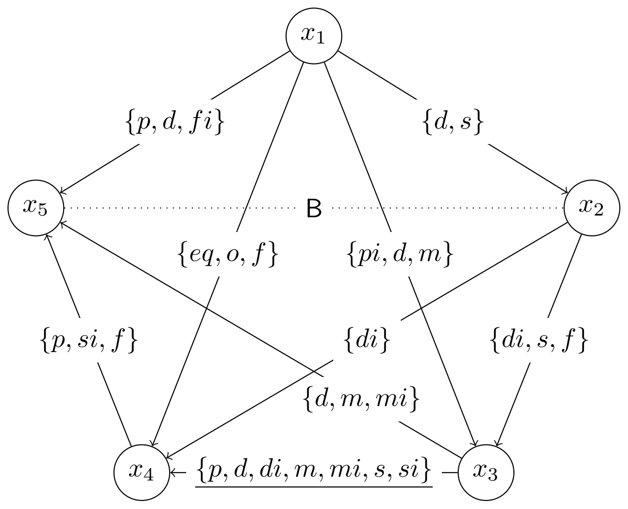
As an illustration, the subset $\mathcal{H}_{\mathsf{IA}}$ of the set of relations of Interval Algebra [23] is tractable for $\diamond_G$-consistency, i.e., $\diamond_G$-consistency is complete for deciding the satisfiability of any QCN defined over $\mathcal{H}_{\mathsf{IA}}$ with respect to a triangulation $G$ of its constraint graph [4]. That subset contains exactly those relations that are transformed to propositional Horn formulas when using the propositional encoding of Interval Algebra [23]. To further facilitate the reader, let us consider the constraint $C(x_3, x_4)$ in the QCN of Interva Algebra in Figure 4. The relation $\{mi, di, si, p, m, d, s\}$ that is associated with that constraint does not appear in the subset $\mathcal{H}_{\mathsf{IA}}$ and hence tractability is not guaranteed in general, but it can be split into subrelations $\{mi\}, \{di, si\}, \{p, m\}, \{d, s\}$ with respect to $\mathcal{H}_{\mathsf{IA}}$; each of those subrelations belongs to $\mathcal{H}_{\mathsf{IA}}$.

### Dynamic Selection of Subrelations via Counting Local Models

It is standard practice in the qualitative constraint-based reasoning community, and the constraint programming community in general, that, given a constraint of some QCN, a subrelation that is most likely to lead to a solution should be prioritized [39, 28]; in the context of finite-domain CSPs, this strategy is known as the *least-constraining value* heuristic [7].

Currently, the state of the art in qualitative constraint-based reasoning implements that selection strategy in a completely static manner. In particular, base relations of a calculus are assigned static weights a priori, and the overall weight that is associated with a subrelation corresponds to the sum of the weights of its base relations [39, 28]. In detail, a weight for a base relation is obtained by successively composing it with every possible relation and calculating the sum of the cardinalities of the results, which is then suitably scaled. Thus, the bigger the weight for a base relation is, the less restrictive that base relation is. For example, the weights of base relations $d$ and $s$ in Interval Algebra are 4 and 2 respectively and, consequently, the weight of relation {d,s} is $4 + 2 = 6$ [39, Figure 9].

[2] Please refer to [15] for the properties that are needed to exploit triangulations of QCNs in terms of tractability preservation.



| $b$ | $\#l$-models |
|---|---|
| $p$ | 6 |
| $d$ | 6 |
| $di$ | 7 |
| $m$ | 4 |
| $mi$ | 5 |
| $s$ | 3 |
| $si$ | 3 |

**Figure 4** Given the above QCN $\mathcal{N} = (V, C)$ of IA, a partition of $C(x_3, x_4)$ with respect to the subset $\mathcal{H}_{\mathsf{IA}}$ [23] is $\{\{mi\}, \{di, si\}, \{p, m\}, \{d, s\}\}$; for this QCN, heuristics *dynamic*_avg and *dynamic*_sum would prioritize relation $\{p, m\}$, and heuristics *static* [39], *dynamic*_max, and *dynamic*_min would prioritize relations $\{d, s\}$, $\{di, si\}$, and $\{mi\}$ respectively.

Two major problems with the aforementioned *static* strategy is that it assumes a uniform use of relations in QCNs (since weights are computed by equally considering all the relations of a calculus), and it does not exploit any structure that may be present in QCNs (a relation that is used to form more than one constraints in a given QCN, which is typically the case, may exhibit different levels of restrictiveness among those constraints).

In this paper, we propose the selection of subrelations to be *dynamic* and, in particular, based on the count of *local models* that the individual base relations of a subrelation are part of. Let $\mathcal{N}{\downarrow}_{V'}$, with $V' \subseteq V$, denote the QCN $\mathcal{N} = (V, C)$ restricted to $V'$, we formally define the notion of local models as follows:

▶ **Definition 3** (local models)**.** Given a QCN $\mathcal{N} = (V, C)$, a graph $G = (V, E)$, and a constraint $C(v, v')$ with $\{v, v'\} \in E$, the *local models* of a base relation $b \in C(v, v')$ are the scenarios $\mathcal{S} = (V', C')$ of $\mathcal{N}{\downarrow}_{V'}$, with $V' = \{v, v', u\}$, such that $\{v, u\}, \{u, v'\} \in E$ and $C(v, v') = \{b\}$.

Simply put, given a QCN $(V, C)$, a graph $G = (V, E)$, and a constraint $C(v, v')$ with $\{v, v'\} \in E$, we count how many times a given base relation $b \in C(v, v')$ participates in the scenarios of each triangle in $G$ that involves variables $v$ and $v'$, i.e., the local models from our perspective. In that sense, our approach can be seen as being similar to a counting-based one for CSPs [24], which, as our own method, formalizes a framework that is adaptive and seeks to make branching decisions that preserve most of the solutions by determining what proportion of local solutions agree with that decision. We devise the following strategies for choosing a subrelation from a given set of subrelations:

- **dynamic_f**: for each subrelation $r'$ find the f count of local models among each base relation $b \in r'$, where $\mathsf{f} \in \{\mathsf{max}, \mathsf{min}, \mathsf{avg}, \mathsf{sum}\}$, then choose the subrelation for which the highest such count was obtained.

In the context of counting local models, dynamic_max, dynamic_min, dynamic_avg, and dynamic_sum prioritize the subrelation with the best *most*, *least*, *on avegare*, and *in aggregate* supportive base relation respectively. At this point, let us revisit the QCN of

**Algorithm 1** Refinement($\mathcal{N}$, $G$, $\mathcal{A}$, f).

```
    in   : A QCN $\mathcal{N} = (V, C)$, a graph $G = (V, E)$, a subset $\mathcal{A} \subseteq 2^{\mathsf{B}}$, and a function
           $\mathsf{f} \in \{\mathsf{max}, \mathsf{min}, \mathsf{avg}, \mathsf{sum}\}$.
    out  : A refinement of $\mathcal{N}$ with respect to $G$ over $\mathcal{A}$, or $\bot^V$.
1   begin
2       $\mathcal{N} \leftarrow \overset{\diamond}{}_{G}(\mathcal{N})$;
3       if $\emptyset \in \overset{\diamond}{}_{G}(\mathcal{N})$ then
4           return $\bot^V$;
5       if $\forall \{v, v'\} \in E,\ C(v, v') \in \mathcal{A}$ then
6           return $\mathcal{N}$;
7       $(v, v') \leftarrow \{v, v'\} \in E$ such that $C(v, v') \notin \mathcal{A}$;
8       foreach $r \in$ dynamicSelection($\mathcal{N}$, $G$, $\mathcal{A}$, $(v, v')$, f) do
9           result $\leftarrow$ Refinement($\mathcal{N}_{[v,v']/r}$, $G$, $\mathcal{A}$, f);
10          if $result \neq \bot^V$ then
11              return $result$;
12      return $\bot^V$;
```

**Algorithm 2** dynamicSelection($\mathcal{N}$, $G$, $\mathcal{A}$, $(v, v')$, f).

```
    in   : A QCN $\mathcal{N} = (V, C)$, a graph $G = (V, E)$, a subset $\mathcal{A} \subseteq 2^{\mathsf{B}}$, a pair of variables
           $(v, v')$, and a function $\mathsf{f} \in \{\mathsf{max}, \mathsf{min}, \mathsf{avg}, \mathsf{sum}\}$.
    out  : A relation $r \in \mathcal{A}$.
1   begin
2       $counter \leftarrow$ hashTable();
3       foreach $r \in \{r_1, r_2, \ldots, r_n \in \mathcal{A} \mid \{r_1, r_2, \ldots, r_n\}$ is a partition of $C(v, v')\}$ do
4           $counter[r] \leftarrow \mathsf{f}\{\mathsf{localModels}(b, \mathcal{N}, G, (v, v')) \mid b \in r\}$;
5       while $counter$ is not empty do
6           $r \leftarrow counter$ key paired with the maximum value;
7           remove $r$ from $counter$;
8           yield $r$;
```

Interval Algebra in Figure 4, where the relation $\{mi, di, si, p, m, d, s\}$ that is associated with the constraint $C(x_3, x_4)$ is split into subrelations $\{mi\}, \{di, si\}, \{p, m\}, \{d, s\}$ with respect to subset $\mathcal{H}_{\mathsf{IA}}$. By viewing the table that lists the count of local models for each base relation in $C(x_3, x_4)$ on the right-hand side of the figure, the reader can verify that each strategy correctly prioritizes its subrelation of choice according to its objective; as a reminder, the weights associated with the static strategy detailed earlier are provided in [39, Figure 9].

## Tackling QCNs via Incorporating Dynamic Branching

For reference, a variation of the state-of-the-art backtracking algorithm for solving a QCN is provided in Algorithm 1, the main diffence to the one appearing in the literature [29, Section 2] being the use of dynamic selection of subrelations, in line 8, instead of selection based on static weights. Another difference is the use of a graph as a parameter, but, over the past few years, this has become a standard way of generalizing the original algorithm to exploit certain properties of a calculus that relate to graphs, see [34] and references therein.

The dynamic strategies that we described earlier are formally presented in Algorithms 2 and 3. In particular, in lines 2–4 of Algorithm 2 we calculate the count of local models for each base relation of each subrelation that pertains to a given constraint. This calculation

**Algorithm 3** localModels($b, \mathcal{N}, G, (v, v')$).

```
in   : A base relation $b$, a QCN $\mathcal{N} = (V, C)$, a graph $G = (V, E)$, and a pair of variables $(v, v')$.
out  : An integer.
1 begin
2     $count \leftarrow 0$;
3     foreach $u \in N_G(v) \cap N_G(v')$ do
4         foreach $(b', b'') \in C(v, u) \times C(u, v')$ do
5             if $b \in b' \diamond b''$ then
6                 $count \leftarrow count + 1$;
7     return $count$;
```

is performed via a call to Algorithm 3. After obtaining the count of models for each such base relation, we implement the chosen strategy by applying the respective function among $\{\mathsf{max}, \mathsf{min}, \mathsf{avg}, \mathsf{sum}\}$ on the results. Now, each subrelation is associated with a number, a dynamic weight, and in lines 5–8 the subrelation with the highest dynamic weight is prioritized each time there is a need for a new subrelation to be tried out in an assignment.

**Complexity Analysis.** Given the fact that for a calculus the number of its base relations, i.e., $|\mathsf{B}|$, can be viewed as a constant, Algorithm 2 calculates the count of local models for a base relation in a given constraint in linear time in the maximum degree of the graph $G$ that is used as a parameter; each subsequent prioritization of a subrelation based on those calculated counts (lines 5–8) takes constant time. In particular, given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, Algorithm 2 runs in $\Theta(\Delta(G))$ time. In practice, there was no noticable slowdown for the dataset that we consider in this paper (see Section 4), which is not surprising, as the search space for solving a QCN is $O(|\mathsf{B}|^{|E|})$ in general.

## 4 Evaluation

In this section we evaluate the proposed dynamic branching heuristics, as well as the state-of-the-art static branching strategy that appears in the literature, with respect to the fundamental reasoning problem of *satisfiability checking* of QCNs. Specifically, we explore the *efficiency* of the involved heuristics in determining the satisfiability of a given network instance when used in the standard backtracking algorithm (see Algorithm 1), and investigate their *fitness score* too, which is the difference *"% of times a heuristic* f *is the best choice"* − *"% of times a heuristic* f *is the worst choice"*; clearly, *fitness score* $\in [-100\%, 100\%]$. Finally, we also present results for two virtual portfolios of reasoners that always make the *best* and *worst* choice of a heuristic respectively for a given network instance.

**Technical specifications.** The evaluation was carried out on a computer with an Intel Core i7-8565U processor, 16 GB of RAM, and the Ubuntu 18.04.4 LTS x86_64 OS. All algorithms were coded in Python and run using the PyPy intepreter under version 5.10.0, which implements Python 2.7.13. Only one CPU core was used per run.

**Dataset.** We generated 7 000 *random* instances of Interval Algebra using model $\mathsf{H}(\mathsf{n} = 40, \mathsf{d})$ [22], 1 000 for each constraint graph degree value $\mathsf{d} \in \{9, 10, 11, 12, 13, 14, 15\}$ specifically, and 4 000 *structured* instances of Interval Algebra using model $\mathsf{BA}(\mathsf{n} = 80, \mathsf{m}, \mathsf{3CNF})$ [31],

**Table 1** Evaluation with random IA networks that were generated using model H(n = 40, d) [22]; $\frac{\text{median} \mid \text{average} \mid \text{maximum \# of visited nodes}}{\text{median} \mid \text{average} \mid \text{maximum CPU time}}$.

| d | *best* | *static* | *dynamic*_max | *dynamic*_min | *dynamic*_avg | *dynamic*_sum | *worst* |
|---|---|---|---|---|---|---|---|
| 9 | $\frac{62.0 \mid 61.8 \mid 95.0}{0.0s \mid 0.0s \mid 10.8s}$ | $\frac{78.0 \mid 79.4 \mid 413.0}{0.0s \mid \mathbf{0.0s} \mid 11.0s}$ | $\frac{107.0 \mid 112.0 \mid 258.0}{0.0s \mid 0.0s \mid 10.8s}$ | $\frac{66.0 \mid \mathbf{65.9} \mid 144.0}{0.0s \mid 0.0s \mid 10.9s}$ | $\frac{78.0 \mid 79.5 \mid 350.0}{0.0s \mid 0.0s \mid 10.8s}$ | $\frac{94.0 \mid 95.6 \mid 520.0}{0.0s \mid 0.0s \mid 10.9s}$ | $\frac{114.0 \mid 120.4 \mid 520.0}{0.0s \mid 0.0s \mid 11.0s}$ |
| 10 | $\frac{52.0 \mid 52.6 \mid 168.0}{0.0s \mid 0.0s \mid 10.9s}$ | $\frac{71.0 \mid 125.8 \mid 8.9k}{0.0s \mid 0.1s \mid 10.9s}$ | $\frac{85.0 \mid 101.3 \mid 2.0k}{0.0s \mid \mathbf{0.0s} \mid 10.9s}$ | $\frac{60.0 \mid \mathbf{89.9} \mid 2.9k}{0.0s \mid 0.0s \mid 11.0s}$ | $\frac{67.0 \mid 104.4 \mid 8.7k}{0.0s \mid 0.1s \mid 10.9s}$ | $\frac{81.0 \mid 129.6 \mid 6.6k}{0.0s \mid 0.1s \mid 10.9s}$ | $\frac{108.0 \mid 227.7 \mid 8.9k}{0.0s \mid 0.1s \mid 11.0s}$ |
| 11 | $\frac{59.0 \mid 679.3 \mid 388.0k}{0.0s \mid 0.5s \mid 4.4m}$ | $\frac{174.0 \mid 2.5k \mid 500.3k}{0.1s \mid 1.7s \mid 5.3m}$ | $\frac{143.5 \mid 2.1k \mid 388.0k}{0.1s \mid 1.4s \mid 4.4m}$ | $\frac{141.0 \mid \mathbf{1.7k} \mid 546.8k}{0.1s \mid \mathbf{1.2s} \mid 6.2m}$ | $\frac{133.0 \mid 2.1k \mid 619.7k}{0.1s \mid 1.4s \mid 6.9m}$ | $\frac{181.0 \mid 2.6k \mid 474.2k}{0.1s \mid 1.9s \mid 5.4m}$ | $\frac{682.0 \mid 5.0k \mid 619.7k}{0.4s \mid 3.4s \mid 6.9m}$ |
| 12 | $\frac{931.5 \mid 10.3k \mid 953.5k}{0.7s \mid 7.4s \mid 10.7m}$ | $\frac{4.4k \mid 24.6k \mid 958.6k}{3.1s \mid 16.8s \mid 10.7m}$ | $\frac{3.5k \mid 21.0k \mid 1.0m}{2.6s \mid 15.1s \mid 11.6m}$ | $\frac{3.1k \mid 18.7k \mid 1.0m}{2.3s \mid 13.5s \mid 12.0m}$ | $\frac{2.9k \mid \mathbf{18.6k} \mid 996.1k}{2.2s \mid \mathbf{13.3s} \mid 11.4m}$ | $\frac{4.7k \mid 24.6k \mid 953.5k}{3.5s \mid 17.4s \mid 10.8m}$ | $\frac{10.0k \mid 34.9k \mid 1.0m}{7.4s \mid 24.5s \mid 12.0m}$ |
| 13 | $\frac{2.0k \mid 7.6k \mid 214.9k}{1.6s \mid 5.9s \mid 2.7m}$ | $\frac{3.4k \mid 11.9k \mid 277.3k}{2.6s \mid 8.7s \mid 3.1m}$ | $\frac{3.2k \mid 11.2k \mid 302.8k}{3.0s \mid 9.8s \mid 4.1m}$ | $\frac{2.7k \mid \mathbf{10.8k} \mid 245.6k}{2.1s \mid \mathbf{8.2s} \mid 3.0m}$ | $\frac{2.9k \mid 10.9k \mid 272.8k}{2.3s \mid 8.4s \mid 3.2m}$ | $\frac{3.4k \mid 11.7k \mid 283.8k}{2.7s \mid 9.1s \mid 3.5m}$ | $\frac{4.6k \mid 15.7k \mid 302.8k}{3.7s \mid 12.3s \mid 4.1m}$ |
| 14 | $\frac{460.0 \mid 1.3k \mid 58.1k}{0.4s \mid 1.1s \mid 43.9s}$ | $\frac{720.0 \mid 1.9k \mid 64.9k}{0.5s \mid \mathbf{1.4s} \mid 47.5s}$ | $\frac{700.5 \mid 1.9k \mid 77.5k}{0.5s \mid 1.5s \mid 58.0s}$ | $\frac{584.5 \mid \mathbf{1.6k} \mid 64.9k}{0.6s \mid 1.6s \mid 1.0m}$ | $\frac{619.0 \mid 1.7k \mid 63.8k}{0.6s \mid 1.6s \mid 55.8s}$ | $\frac{708.5 \mid 2.0k \mid 106.0k}{0.6s \mid 1.6s \mid 1.3m}$ | $\frac{933.0 \mid 2.5k \mid 106.0k}{0.8s \mid 2.1s \mid 1.3m}$ |
| 15 | $\frac{113.0 \mid 220.7 \mid 2.6k}{0.1s \mid 0.2s \mid 11.2s}$ | $\frac{179.0 \mid 365.2 \mid 5.4k}{0.1s \mid 0.3s \mid 11.2s}$ | $\frac{168.5 \mid 350.2 \mid 4.4k}{0.1s \mid 0.3s \mid 11.3s}$ | $\frac{157.0 \mid \mathbf{285.7} \mid 2.8k}{0.1s \mid \mathbf{0.2s} \mid 11.4s}$ | $\frac{167.0 \mid 317.9 \mid 4.8k}{0.1s \mid 0.3s \mid 11.3s}$ | $\frac{176.0 \mid 360.2 \mid 5.9k}{0.1s \mid 0.3s \mid 11.3s}$ | $\frac{251.5 \mid 463.7 \mid 5.9k}{0.2s \mid 0.4s \mid 11.4s}$ |

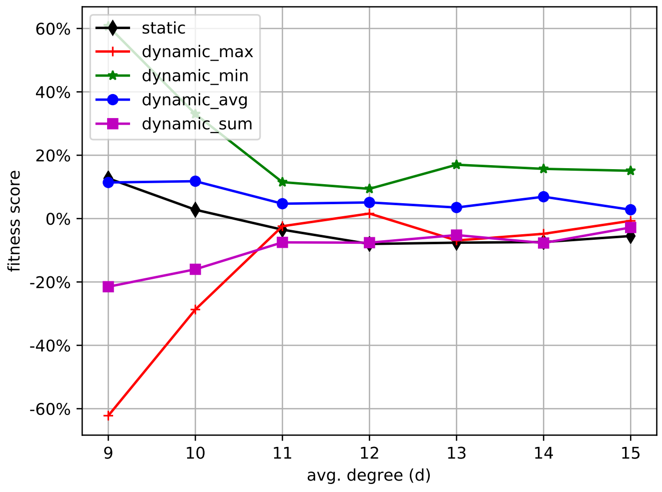


**Figure 5** Fitness score of each strategy for the instances of Table 1; *"% of times a heuristic* f *is the best choice" – "% of times a heuristic* f *is the worst choice".*

1 000 for each constraint graph *preferential attachment* [2] value $m \in \{4, 5, 6, 7\}$ specifically. In both of the aforementioned generation models, constraints were picked from the set of relations expressible in 3CNF when transformed into first-order formulae [22], in order to increase the branching factor in the search tree as much as possible. Finally, regarding the graphs that were given as input to our algorithms, the *maximum cardinality search* algorithm [38] was used to obtain triangulations of the constraint graphs of our QCNs.

**Results.** Regarding the generation model $H(n = 40, d)$, the main results are presented in Table 1. The dynamic strategies of *dynamic_*min and *dynamic_*avg are up to 20% faster on average than the *static* one in the phase transition of the tested instances.[3] Specifically, the phase transition covers mostly the case where $d = 12$, and a little less the case where $d = 13$. With respect to the rest of the dynamic heuristics, viz., *dynamic_*max and *dynamic_*sum, the results suggest that *dynamic_*max outperforms *static* by a small margin on average in the phase transition, whilst *dynamic_*sum almost mimics the performance of *static*, if not arguably being a little worse than *static* overall. This last finding informs us that relying too much on the number of base relations of a relation (viz., the cardinality of a relation) is a bad choice in general, i.e., it is better to focus on few base relations individually, where each one appears in many local models (quality), than on many base relations aggregately, where each one appears in few local models (quantity). The aforementioned results are depicted from a different perspective and complemented in Figure 5, where the fitness score for each heuristic is detailed. The superiority of heuristics *dynamic_*min and *dynamic_*avg among all strategies becomes even more so clear, and the marginal performance gains of *dynamic_*max, and *dynamic_*sum at times with respect to *static* in the phase transition are well-captured by their fitness scores too. Finally, at this point, it is interesting to observe the performance of the virtual portfolios of reasoners *best* and *worst*; as a reminder these always make the *best*

[3] Even though the improvement for this particular dataset may not seem that drastic, bear in mind that the instances of *this* dataset have little to no structure as their constraint graphs are regular graphs.

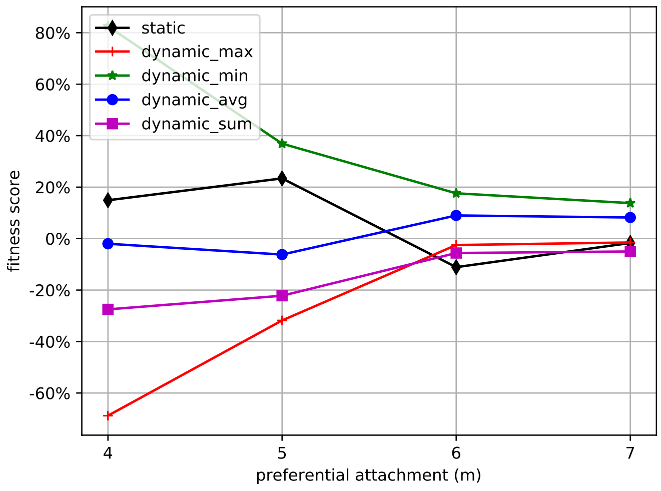


**Figure 6** Fitness factor of each strategy for the instances of Table 2; *"% of times a heuristic* f *is the best choice" – "% of times a heuristic* f *is the worst choice"*.

and *worst* choice of a heuristic respectively for a given network instance. The performance of portfolio *best* in particular allows us to be optimistic about future research in dynamic strategies, since it shows that there is still a lot of room for improvement. Specifically, research could be carried out both in terms of defining new dynamic strategies and in terms of devising selection protocols that choose among already existing strategies.

Regarding the generation model $\mathsf{BA}(\mathsf{n} = 80, \mathsf{m}, \mathsf{3CNF})$, the results are presented in Table 2 and Figure 6. Here, *dynamic*_min and *dynamic*_avg are up to 3 and 5 times faster on average respectively than the *static* one in the phase transition of the tested instances, which appears for $\mathsf{m} = 6$. The rest of the results are qualitative similar to the previous dataset.

Since the runtime distribution is heavy-tailed for both datasets, the interested reader may want to look into the $0.5^{\text{th}}$ percentile of most difficult instances pertaining to Tables 1 and 2 for each strategy, depicted in Figures 7 and 8 respectively in Appendix A.

## 5 Conclusion and Future Work

We introduced and evaluated *dynamic branching* strategies for solving QCNs via backtracking search, based on the *count* of *local models* (or *local scenarios*) that a possible relation for a given constraint is involved with in a considered QCN. Thus, we addressed a limitation in the state of the art in qualitative constraint-based reasoning, where the selection of a possible relation for a given contraint is dictated a priori by precomputed static weights, without any regard to the particulars of a given network instance of that calculus, such as its structure. Our approach is adaptive and seeks to make branching decisions that preserve most of the solutions by determining what proportion of local solutions agree with that decision. An evaluation with a *random* and a *structured* dataset of QCNs of Interval Algebra showed that up to 5 times better performance may be achieved for structured instances, whilst non-negligible gains of around 20% are maintained for random ones.

As this is a new approach, there are many directions for future work. In particular, we aim to devise selection protocols that choose among different strategies, as the performance of the virtual portfolio *best* in our evaluation, which always makes the *best* choice of a heuristic for

**Table 2** Evaluation with structured IA networks that were generated using model $\mathsf{BA}(\mathsf{n} = 80, \mathsf{m}, \mathsf{3CNF})$ [31]; $\frac{\text{median|average|maximum \# of visited nodes}}{\text{median|average|maximum CPU time}}$.

| m | *best* | *static* | *dynamic*_max | *dynamic*_min | *dynamic*_avg | *dynamic*_sum | *worst* |
|---|---|---|---|---|---|---|---|
| 4 | $\frac{144.0\|144.1\|177.0}{0.0s\|0.0s\|10.7s}$ | $\frac{166.0\|167.1\|257.0}{0.0s\|\mathbf{0.0s}\|10.8s}$ | $\frac{213.0\|216.0\|332.0}{0.0s\|0.0s\|10.8s}$ | $\frac{146.0\|\mathbf{146.3}\|320.0}{0.0s\|0.0s\|10.8s}$ | $\frac{178.0\|177.0\|231.0}{0.0s\|0.0s\|10.8s}$ | $\frac{198.0\|198.9\|321.0}{0.0s\|0.0s\|10.7s}$ | $\frac{219.0\|222.3\|332.0}{0.0s\|0.0s\|10.8s}$ |
| 5 | $\frac{113.0\|114.5\|550.0}{0.0s\|0.0s\|10.8s}$ | $\frac{128.0\|154.4\|1.0k}{0.0s\|0.1s\|10.8s}$ | $\frac{155.0\|173.3\|1.6k}{0.0s\|0.1s\|11.0s}$ | $\frac{124.0\|\mathbf{139.9}\|2.9k}{0.0s\|\mathbf{0.1s}\|10.9s}$ | $\frac{141.0\|159.8\|1.2k}{0.0s\|0.1s\|10.8s}$ | $\frac{151.0\|177.8\|1.2k}{0.0s\|0.1s\|10.9s}$ | $\frac{178.0\|226.3\|2.9k}{0.1s\|0.1s\|11.0s}$ |
| 6 | $\frac{121.0\|452.3\|72.5k}{0.1s\|0.6s\|1.7m}$ | $\frac{235.0\|4.7k\|1.3m}{0.3s\|5.4s\|23.7m}$ | $\frac{201.0\|7.4k\|3.8m}{0.2s\|8.1s\|1.1h}$ | $\frac{172.0\|1.4k\|460.4k}{0.2s\|1.8s\|8.8m}$ | $\frac{182.5\|\mathbf{933.4}\|139.6k}{0.2s\|\mathbf{1.2s}\|3.1m}$ | $\frac{223.0\|4.0k\|2.0m}{0.3s\|4.6s\|34.5m}$ | $\frac{337.5\|10.6k\|3.8m}{0.4s\|11.8s\|1.1h}$ |
| 7 | $\frac{34.0\|82.4\|3.4k}{0.0s\|0.1s\|11.1s}$ | $\frac{51.0\|178.9\|10.4k}{0.1s\|0.2s\|15.2s}$ | $\frac{50.0\|168.7\|17.9k}{0.1s\|0.2s\|27.7s}$ | $\frac{47.0\|\mathbf{110.8}\|3.4k}{0.1s\|\mathbf{0.2s}\|11.3s}$ | $\frac{49.0\|129.4\|5.0k}{0.1s\|0.2s\|11.2s}$ | $\frac{51.0\|206.6\|22.6k}{0.1s\|0.3s\|35.2s}$ | $\frac{74.5\|252.2\|22.6k}{0.1s\|0.4s\|35.2s}$ |

a given network instance, revealed that there is still a lot of room for improvement. Further, more sophisticated dynamic heuristics could be developed by going beyond triples of variables, which currently form the local models, and engaging larger parts of an instance. Finally, we would like to pair this approach with ongoing research on singleton consistencies [33, 30], and implement adaptive reasoners where the level of consistency checking during search would be adjusted according to the count of local models pertaining to a given constraint.

## References

**1** James F. Allen. Maintaining Knowledge about Temporal Intervals. *Commun. ACM*, 26(11):832–843, 1983. doi:10.1145/182.358434.

**2** Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science (New York, N.Y.)*, 286(5439):509–512, 1999. doi:10.1126/science.286.5439.509.

**3** Davide Bresolin, Dario Della Monica, Angelo Montanari, Pietro Sala, and Guido Sciavicco. Decidability and complexity of the fragments of the modal logic of Allen's relations over the rationals. *Inf. Comput.*, 266:97–125, 2019. doi:10.1016/j.ic.2019.02.002.

**4** Assef Chmeiss and Jean-Francois Condotta. Consistency of Triangulated Temporal Qualitative Constraint Networks. In *ICTAI*, pages 799–802, 2011. doi:10.1109/ICTAI.2011.125.

**5** Zhan Cui, Anthony G. Cohn, and David A. Randell. Qualitative Simulation Based on a Logical Formalism of Space and Time. In *AAAI*, pages 679–684, 1992. URL: http://www.aaai.org/Library/AAAI/1992/aaai92-105.php.

**6** Daniel de Leng and Fredrik Heintz. Qualitative Spatio-Temporal Stream Reasoning with Unobservable Intertemporal Spatial Relations Using Landmarks. In *AAAI*, pages 957–963, 2016. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12077.

**7** Rina Dechter. *Constraint processing*. Elsevier Morgan Kaufmann, 2003.

**8** Frank Dylla, Jae Hee Lee, Till Mossakowski, Thomas Schneider, André van Delden, Jasper van de Ven, and Diedrich Wolter. A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Comput. Surv.*, 50(1):7:1–7:39, 2017. doi:10.1145/3038927.

**9** Frank Dylla, Till Mossakowski, Thomas Schneider, and Diedrich Wolter. Algebraic Properties of Qualitative Spatio-temporal Calculi. In *COSIT*, pages 516–536, 2013. doi:10.1007/978-3-319-01790-7_28.

**10** Frank Dylla and Jan Oliver Wallgrün. Qualitative Spatial Reasoning with Conceptual Neighborhoods for Agent Control. *J. Intell. Robotic Syst.*, 48(1):55–78, 2007. doi:10.1007/s10846-006-9099-4.

**11** David Gabelaia, Roman Kontchakov, Ágnes Kurucz, Frank Wolter, and Michael Zakharyaschev. Combining Spatial and Temporal Logics: Expressiveness vs. Complexity. *J. Artif. Intell. Res.*, 23:167–243, 2005. doi:10.1613/jair.1537.

**12** Johann Gamper and Wolfgang Nejdl. Abstract temporal diagnosis in medical domains. *Artificial Intelligence in Medicine*, 10(3):209–234, 1997. doi:10.1016/S0933-3657(97)00393-X.

**13** Zeno Gantner, Matthias Westphal, and Stefan Wölfl. GQR-A Fast Reasoner for Binary Qualitative Constraint Calculi. In *AAAI Workshop on Spatial and Temporal Reasoning*, 2008.

**14** Fredrik Heintz and Daniel de Leng. Spatio-Temporal Stream Reasoning with Incomplete Spatial Information. In *ECAI*, pages 429–434, 2014. doi:10.3233/978-1-61499-419-0-429.

**15** Jinbo Huang. Compactness and its implications for qualitative spatial and temporal reasoning. In *KR*, 2012.

**16** Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. Spatial Logic + Temporal Logic = ? In *Handbook of Spatial Logics*, pages 497–564. Springer-Verlag, 2007. doi:10.1007/978-1-4020-5587-4_9.

**17** Nikhil Krishnaswamy, Scott Friedman, and James Pustejovsky. Combining Deep Learning and Qualitative Spatial Reasoning to Learn Complex Structures from Sparse Examples with Noise. In *AAAI*, pages 411–415, 2019.

18 Gérard Ligozat. *Qualitative Spatial and Temporal Reasoning*. ISTE. Wiley, 2013.

19 Gérard Ligozat and Jochen Renz. What Is a Qualitative Calculus? A General Framework. In *PRICAI*, pages 53–64, 2004. doi:10.1007/978-3-540-28633-2_8.

20 Antonio Morales, Isabel Navarrete, and Guido Sciavicco. A new modal logic for reasoning about space: spatial propositional neighborhood logic. *Ann. Math. Artif. Intell.*, 51(1):1–25, 2007. doi:10.1007/s10472-007-9083-0.

21 Emilio Muñoz-Velasco, Mercedes Pelegrín-García, Pietro Sala, Guido Sciavicco, and Ionel Eduard Stan. On coarser interval temporal logics. *Artif. Intell.*, 266:1–26, 2019. doi:10.1016/j.artint.2018.09.001.

22 Bernhard Nebel. Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class. *Constraints*, 1(3):175–190, 1997. doi:10.1007/BF00137869.

23 Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. *J. ACM*, 42(1):43–66, 1995. doi:10.1145/200836.200848.

24 Gilles Pesant, Claude-Guy Quimper, and Alessandro Zanarini. Counting-Based Search: Branching Heuristics for Constraint Satisfaction Problems. *J. Artif. Intell. Res.*, 43:173–210, 2012. doi:10.1613/jair.3463.

25 David A. Randell, Zhan Cui, and Anthony Cohn. A Spatial Logic Based on Regions and Connection. In *KR*, pages 165–176, 1992.

26 Jochen Renz. Qualitative Spatial and Temporal Reasoning: Efficient Algorithms for Everyone. In *IJCAI*, pages 526–531, 2007.

27 Jochen Renz and Gérard Ligozat. Weak Composition for Qualitative Spatial and Temporal Reasoning. In *CP*, pages 534–548, 2005. doi:10.1007/11564751_40.

28 Jochen Renz and Bernhard Nebel. Efficient Methods for Qualitative Spatial Reasoning. *J. Artif. Intell. Res.*, 15:289–318, 2001. doi:10.1613/jair.872.

29 Jochen Renz and Bernhard Nebel. Qualitative Spatial Reasoning Using Constraint Calculi. In *Handbook of Spatial Logics*, pages 161–215. Springer-Verlag, 2007. doi:10.1007/978-1-4020-5587-4_4.

30 Michael Sioutis. Just-In-Time Constraint-Based Inference for Qualitative Spatial and Temporal Reasoning. *KI*, 34(2):259–270, 2020. doi:10.1007/s13218-020-00652-z.

31 Michael Sioutis, Jean-François Condotta, and Manolis Koubarakis. An Efficient Approach for Tackling Large Real World Qualitative Spatial Networks. *Int. J. Artif. Intell. Tools*, 25:1–33, 2016. doi:10.1142/S0218213015500311.

32 Michael Sioutis, Zhiguo Long, and Tomi Janhunen. On Robustness in Qualitative Constraint Networks. In *IJCAI*, pages 448–455, 2020. To appear.

33 Michael Sioutis, Anastasia Paparrizou, and Jean-François Condotta. Collective singleton-based consistency for qualitative constraint networks: Theory and practice. *Theor. Comput. Sci.*, 797:17–41, 2019. doi:10.1016/j.tcs.2019.02.028.

34 Michael Sioutis, Yakoub Salhi, and Jean-François Condotta. Studying the use and effect of graph decomposition in qualitative spatial and temporal reasoning. *Knowl. Eng. Rev.*, 32:e4, 2017. doi:10.1017/S026988891600014X.

35 Jakob Suchan and Mehul Bhatt. Semantic Question-Answering with Video and Eye-Tracking Data: AI Foundations for Human Visual Perception Driven Cognitive Film Studies. In *IJCAI*, pages 2633–2639, 2016. URL: http://www.ijcai.org/Abstract/16/374.

36 Jakob Suchan, Mehul Bhatt, and Srikrishna Varadarajan. Out of Sight But Not Out of Mind: An Answer Set Programming Based Online Abduction Framework for Visual Sensemaking in Autonomous Driving. In *IJCAI*, pages 1879–1885, 2019. doi:10.24963/ijcai.2019/260.

37 Jakob Suchan, Mehul Bhatt, Przemyslaw Andrzej Walega, and Carl P. L. Schultz. Visual Explanation by High-Level Abduction: On Answer-Set Programming Driven Reasoning About Moving Objects. In *AAAI*, pages 1965–1972, 2018. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17303.

38 Robert E. 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. doi:10.1137/0213035.

39 Peter van Beek and Dennis W. Manchak. The design and experimental analysis of algorithms for temporal reasoning. *J. Artif. Intell. Res.*, 4(1):1–18, 1996.

## A Evaluation Figures

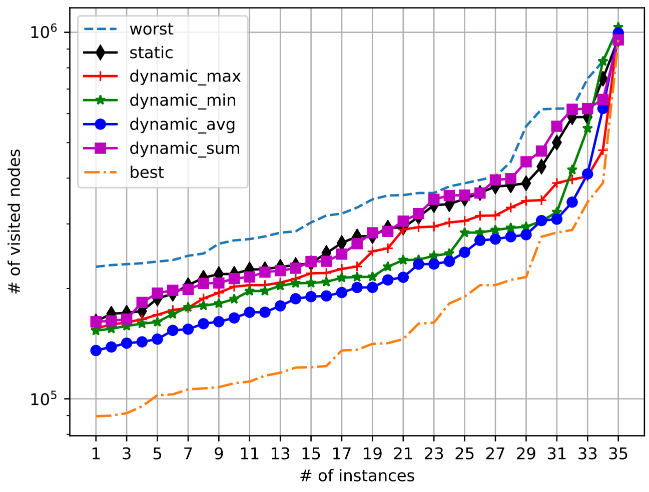


**Figure 7** Insight into the 0.5th percentile of most difficult instances of Table 1 for each strategy.

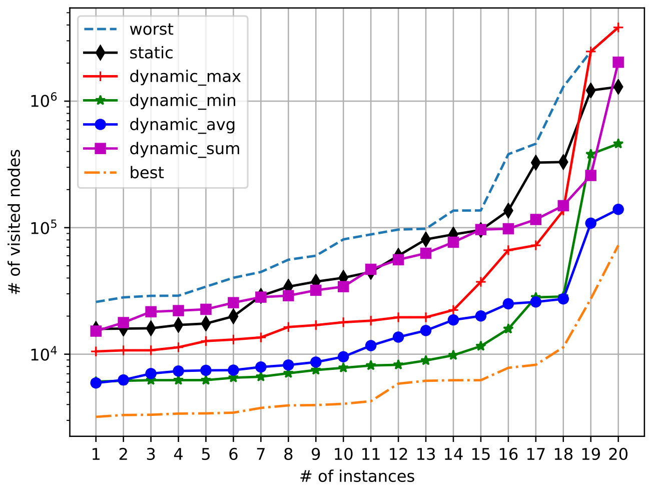


**Figure 8** Insight into the 0.5th percentile of most difficult instances of Table 2 for each strategy.