A Simple Decomposition Scheme For Large Real World Qualitative Constraint Networks*

Michael Sioutis and **Yakoub Salhi** and **Jean-François Condotta**
Université Lille-Nord de France, Artois
CRIL-CNRS UMR 8188
Lens, France
Email: {sioutis,salhi,condotta}@cril.fr

Abstract

We improve the state-of-the-art in checking the satisfiability of large real world qualitative constraint networks (QCNs), by exploiting the loosely connected structure of their underlying graphs. We propose a simple decomposition scheme that retrieves the smaller QCNs that correspond to the biconnected component subgraphs of the underlying graph of a given large QCN, and show that our approach is sound for a qualitative constraint language that has a particular constraint property for atomic QCNs, namely, patchwork. Experimental evaluation shows that state-of-the-art reasoners can significanlty benefit from adopting this approach.

Introduction

Qualitative Reasoning is a major field of study in Artificial Intelligence, particularly in Knowledge Representation, that deals with continuous aspects of the physical world, such as space and time, using qualitative information. This field has gained a lot of attention during the last few years as it extends to a plethora of areas and domains that include ambient intelligence, dynamic GIS, cognitive robotics, and spatiotemporal design (Hazarika 2012).

Qualitative knowledge can be modelled as a qualitative constraint network (QCN), which can be seen as the infinite-domain variant of a Constraint Satisfaction Problem (CSP) (Dechter 2003). For instance, we can have infinitely many time points or temporal intervals on the time line and infinitely many regions in a two or three dimensional space. In this context, an emphasis has been made in recent literature on the satisfiability problem of *large real world* QCNs (Nikolaou and Koubarakis 2014; Sioutis and Condotta 2014; Sioutis 2014). The satisfiability problem is deciding if there exists a solution of a given QCN. Large real world datasets have already been, and are to be, offered by the Semantic Web community and scale up to millions of nodes (Nikolaou and Koubarakis 2014; Sioutis and Condotta 2014). Further, there is an ever increasing interest in coupling qualititave reasoning techniques with linked open data that are constantly being made available and are expected to encode a huge amount of qualitative constraints (Koubarakis et al. 2011). Thus, there is a real emergent need for scalable implementations of algorithms that can tackle large real world QCNs efficiently (Koubarakis et al. 2011; Nikolaou and Koubarakis 2014; Sioutis and Condotta 2014; Sioutis 2014).

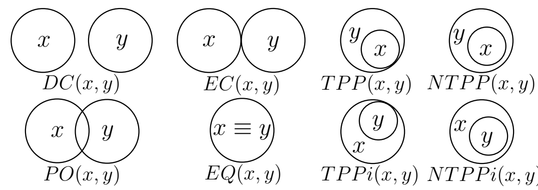

Figure 1: The RCC8 constraint language

In this paper, we propose a simple decomposition scheme that exploits the loosely connected structure of large real world QCNs. In particular, we extract the smaller QCNs that correspond to the biconnected component subgraphs of the underlying graph of a given large QCN, and show that the overall approach is sound for a constraint language that has patchwork (Lutz and Milicic 2007) for atomic QCNs. Intuitively, patchwork ensures that the combination of two satisfiable QCNs that completely agree on the constraints between their common variables continues to be satisfiable. Our approach allows for reasoning with the smaller biconnected QCNs completely separately, in a parallel or serial fashion, which, as our experimentation suggests, significantly decongests search when solving non-tractable QCNs.

Preliminaries

A (binary) qualitative temporal or spatial constraint language is based on a finite set B of *jointly exhaustive and pairwise disjoint* (JEPD) relations defined on an infinite domain D (Ladkin and Maddux 1994), called the set of base relations. The set of base relations B of a particular qualitative constraint language can be used to represent the definite knowledge between any two entities with respect to the given level of granularity. B contains the identity relation Id, and is closed under the converse operation ($^{-1}$). Indefinite knowledge can be specified by unions of possible base relations, and is represented by the set containing them. Hence, 2^{B} represents the total set of relations. 2^{B} is equipped with the usual set-theoretic operations (union and intersection),

*The work was funded by a PhD grant from Université d'Artois and region Nord-Pas-de-Calais.

the converse operation, and the weak composition operation denoted by $\diamond$ (Renz and Ligozat 2005). As illustration, the set of base relations of RCC8 (Randell, Cui, and Cohn 1992) is the set $\{dc, ec, po, tpp, ntpp, tppi, ntppi, eq\}$, where eq is the identity relation. These eight relations represent the binary topological relations between *regions* that are non-empty regular subsets of some topological space, as depicted in Figure 1 (for the $2D$ case). We can capture such qualitative knowledge using qualitative constraint networks (QCNs), which are defined as follows.

Definition 1 *A* QCN *is a pair* $\mathcal{N} = (V, C)$ *where:* V *is a non-empty finite set of variables;* C *is a mapping that associates a relation* $C(v, v') \in 2^{\mathsf{B}}$ *to each pair* (v, v') *of* $V \times V$*.* C *is such that* $C(v, v) = \{\mathsf{Id}\}$ *and* $C(v, v') = (C(v', v))^{-1}$*.*

In what follows, given a QCN $\mathcal{N} = (V, C)$ and $v, v' \in V$, $\mathcal{N}[v, v']$ will denote the relation $C(v, v')$. Given two QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V', C')$, $\mathcal{N} \cup \mathcal{N}'$ denotes the QCN $\mathcal{N}'' = (V'', C'')$ where $V'' = V \cup V'$, $\mathcal{N}''[u, v] = \mathcal{N}''[v, u] = \mathsf{B}\ \forall (u, v) \in (V \setminus V') \times (V' \setminus V)$, $\mathcal{N}''[u, v] = \mathcal{N}[u, v] \cap \mathcal{N}'[u, v]\ \forall u, v \in V \cap V'$, $\mathcal{N}''[u, v] = \mathcal{N}[u, v]$ $\forall (u, v) \in (V \times V) \setminus (V' \times V')$, and $\mathcal{N}''[u, v] = \mathcal{N}'[u, v]$ $\forall (u, v) \in (V' \times V') \setminus (V \times V)$. Given a QCN $\mathcal{N} = (V, C)$, $\mathcal{N}$ is said to be *trivially inconsistent* iff $\exists v, v' \in V$ with $\mathcal{N}[v, v'] = \emptyset$. A *solution* of $\mathcal{N}$ is a mapping σ defined from V to the domain D such that for every pair (v, v') of variables in V, $(\sigma(v), \sigma(v'))$ can be described by $\mathcal{N}[v, v']$, i.e., there exists a base relation $b \in \mathcal{N}[v, v']$ such that the relation defined by $(\sigma(v), \sigma(v'))$ is b. A *sub*-QCN $\mathcal{N}'$ of $\mathcal{N}$, denoted by $\mathcal{N}' \subseteq \mathcal{N}$, is a QCN (V, C') such that $\mathcal{N}'[v, v'] \subseteq \mathcal{N}[v, v']\ \forall v, v' \in V$. If b is a base relation, $\{b\}$ is a singleton relation. An *atomic* QCN is a QCN where each constraint is a singleton relation. A *scenario* $\mathcal{S}$ of $\mathcal{N}$ is an atomic satisfiable sub-QCN of $\mathcal{N}$. The constraint graph of a QCN $\mathcal{N} = (V, C)$ is the graph (V, E), denoted by $G(\mathcal{N})$, for which we have that $(v, v') \in E$ iff $\mathcal{N}[v, v'] \neq \mathsf{B}$.

Definition 2 *A* QCN $\mathcal{N}$ *is* satisfiable *iff it admits a solution.*

Checking the satisfiability of a QCN is $\mathcal{NP}$-hard in general for the most well-known and interesting calculi such as RCC8 (Renz and Nebel 1999) and IA (Nebel and Bürckert 1995). However, there exist maximal tractable subclasses $\mathcal{A} \subseteq 2^{\mathsf{B}}$ containing all singleton relations for which the satisfiability problem becomes tractable through the use of $\diamond$*-consistency*. A QCN $\mathcal{N}$ is $\diamond$*-consistent* or *closed under weak composition* iff $\forall v, v', v'' \in V$ we have that $\mathcal{N}[v, v'] \subseteq \mathcal{N}[v, v''] \diamond \mathcal{N}[v'', v']$. Given a QCN $\mathcal{N} = (V, C)$, $\diamond$-consistency can be applied in $O(|V|^3)$ time. We have that not trivially inconsistent and $\diamond$-consistent QCNs defined on a maximal tractable subclass $\mathcal{A} \subseteq 2^{\mathsf{B}}$ containing all singleton relations are satisfiable. For example, the maximal tractable subclasses for RCC8 and IA are $\hat{\mathcal{H}}_8, \mathcal{C}_8$, and $\mathcal{Q}_8$ (Renz and Nebel 2001), and $\mathcal{H}_{\mathsf{IA}}$ (Nebel 1997), respectively.

We now recall the definition of the *patchwork* property that was originally introduced in (Lutz and Milicic 2007), in the context of qualitative constraint languages.

Definition 3 *A constraint language has patchwork, if for any finite satisfiable* QCN*s* $\mathcal{N} = (V, C)$ *and* $\mathcal{N}' = (V', C')$ *defined in this language such that* $\forall u, v \in V \cap V'$ *we have that* $\mathcal{N}[u, v] = \mathcal{N}'[u, v]$*, network* $\mathcal{N} \cup \mathcal{N}'$ *is satisfiable.*

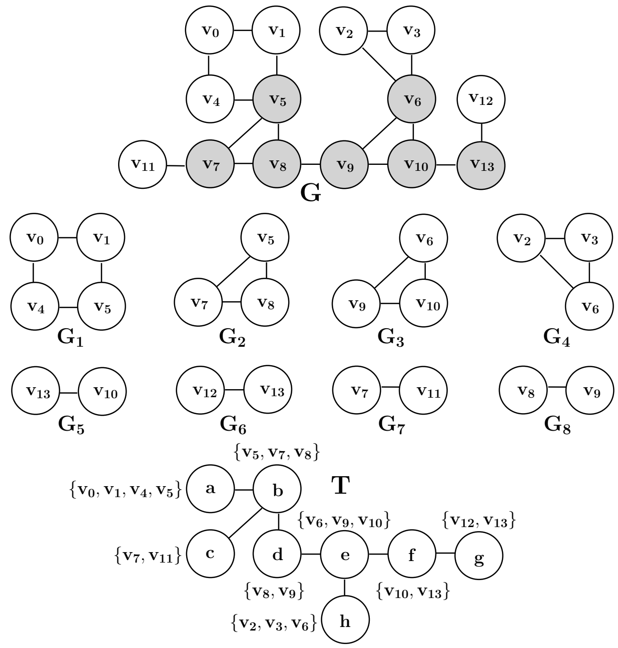

Figure 2: A graph G (top) with its biconnected components (middle) and its tree decomposition (bottom)

Intuitively, patchwork ensures that the combination of two satisfiable constraint networks that completely agree on the constraints between their common variables continues to be satisfiable. From (Lutz and Milicic 2007), we have that patchwork holds for atomic QCNs of RCC8 and IA. Huang further showed that $\diamond$-consistent QCNs defined on one of the maximal tractable subclasses $\hat{\mathcal{H}}_8, \mathcal{C}_8$, and $\mathcal{Q}_8$ for RCC8, and $\mathcal{H}_{\mathsf{IA}}$ for IA, have patchwork (Huang 2012). In what follows in this paper, the former result will be sufficient.

A Simple Decomposition Algorithm

In this section we present a simple decomposition scheme that is based on exploiting the sparse and loosely connected structure of real world qualitative costraint networks, which have been of high interest in recent literature (Nikolaou and Koubarakis 2014; Sioutis and Condotta 2014; Sioutis 2014).

First, we recall a definition from (Dechter 2003) regarding biconnected graphs.

Definition 4 *A connected graph* $G = (V, E)$ *is said to have an* articulation vertex u *if there exist vertices* v *and* v' *such that all paths connecting* v *and* v' *pass through* u*. A graph that has an articulation vertex is called* separable*, and one that has none is called* biconnected*. A maximal subgraph with no articulation vertices is a* biconnected component*.*

Intuitively, an articulation vertex is any vertex whose removal increases the number of connected components in a given graph. From (Dechter 2003) we also have the following property.

Property 1 *Any graph* G *has a* tree decomposition *(Diestel 2012)* $(T, \{X_1, \ldots, X_n\})$ *where* $X_i \subseteq V(G)$ *induces a biconnected component of* G *for every* $i \in \{1, \ldots, n\}$*.*

Algorithm 1: Decomposer($\mathcal{N}$)

```
   in     : A QCN N = (V, C).
   output : χ, a collection of biconnected QCNs.
1  begin
2      S ← {g | g ∈ BCSubgraphs(G(N)); and |V(g)| > 2};
3      χ ← ∅;
4      while S ≠ ∅ do
5          g ← S.pop();
6          V_g ← V(g);
7          C_g ← map({((v, v') : (B if v ≠ v' else
           {Id})) | v, v' ∈ V_g});
8          foreach (v, v') ∈ E(g) do
9              C_g(v, v') ← C(v, v');
10         χ ← χ ∪ {(V_g, C_g)};
11     return χ;
```

Let us now view the discussed notions in an example.

Example 1 *Figure 2 depicts a graph G, along with its biconnected components, and its tree decomposition. Vertices in grey are the articulation vertices of G. The tree decomposition comprises a tree T and a cluster X_i for every node i of that tree, e.g., $X_a = \{v_0, v_1, v_4, v_5\}$.*

We can obtain the following proposition.

Proposition 1 *Let $\mathcal{N}$ be a* QCN *defined on a language that has patchwork for atomic* QCN*s, and let $\{G_1, \ldots, G_k\}$ be the biconnected components of its constraint graph $G(\mathcal{N})$. Then, $\mathcal{N}$ is satisfiable iff $\mathcal{N}_i$ is satisfiable for every $i \in \{1, \ldots, k\}$, where $\mathcal{N}_i$ is the* QCN *that corresponds to G_i.*

Proof By Property 1, the constraint graph $G(\mathcal{N})$ has a tree decomposition $(T, \{X_1, \ldots, X_k\})$, where cluster X_i induces G_i for every $i \in \{1, \ldots, k\}$. We can also infer by Definition 4, that $\forall i, j \in \{1, \ldots, k\}$ with $i \neq j$, $V(G_i) \cap V(G_j)$ contains at most one vertex u (an articulation vertex). If $\mathcal{N}_i$ is satisfiable for every $i \in \{1, \ldots, k\}$, we will have by Definition 2 a solution σ_i, that in its turn will yield a scenario (i.e., an atomic satisfiable QCN) S_i, for every $i \in \{1, \ldots, k\}$. For any possible solutions, and, thus, any possible scenarios, we will have that $\mathcal{S}_i$ and $\mathcal{S}_j$ will always agree on the single unary constraint that is defined by a single vertex $u \in V(G_i) \cap V(G_j)$, as by Definition 1 we have that $\mathcal{S}_i[u, u] = \mathcal{S}_j[u, u] = \{\mathsf{Id}\}\ \forall u \in V(G(\mathcal{N}))$. By Definition 3, we can apply patchwork to patch together all atomic QCNs $\mathcal{S}_i$ for every $i \in \{1, \ldots, k\}$ in a tree-like manner and, thus, derive the satisfiability of $\mathcal{N}$. If $\mathcal{N}$ is satisfiable, then, clearly, $\mathcal{N}_i$ will be satisfiable for every $i \in \{1, \ldots, k\}$. $\dashv$

It is important to note that the proof of Proposition 1 is based on tree decompositions whose nodes correspond to clusters that share at most one vertex with one another. In any other case, the QCNs induced by the clusters need not only be satisfiable, but not trivialy inconsistent and $\diamond$-consistent, as it is done in (Sioutis and Koubarakis 2012) and (Chmeiss and Condotta 2011) for RCC8 and IA respectively. A simple algorithm for obtaining a collection of QCNs that correspond to the biconnected components of the constraint graph of a given QCN is shown in Algorithm 1. Function BCSubgraphs(G) in line 2 returns the biconnected components of a graph $G = (V, E)$ and has a complexity of $O(|E|)$ (Dechter 2003), which dominates the overall complexity of the algorithm. Note that in line 2 we also keep only the components of order > 2, as we need to have a network of at least 3 vertices to perform $\diamond$-consistency. In what follows, we always consider components of order > 2.

network	# of nodes	# of edges	avg. degree
`nuts`	2 236	3 176	2.84
`adm1`	11 762	44 833	7.62
`gadm1`	42 750	159 600	7.47
`gadm2`	276 728	590 443	4.27
`adm2`	1 733 000	5 236 270	6.04

Table 1: Characteristics of real RCC8 networks

network	# of components	max order	median order	min order
`nuts`	64	52	8	3
`adm1`	5	11 666	30	3
`gadm1`	166	19 864	6	3
`gadm2`	2 285	2 371	18	3
`adm2`	2 889	22 808	579	4

Table 2: Biconnected components of real RCC8 networks

Experimentation

In this section we are concerned with the dataset of real RCC8 network instances that was originally introduced in (Nikolaou and Koubarakis 2014). However, we do not consider the accompanying reasoner of (Nikolaou and Koubarakis 2014) in the evaluation to follow, as it has been found to perform very poorly (Sioutis 2014) and it is also not sound (Sioutis, Salhi, and Condotta 2015).

The characteristics of these networks are presented in Table 1. As it can be seen, the networks vary in size, but they are all relatively sparse. This comes as no surprise, as many real world networks seem to present a scale-free structure (Barabasi and Bonabeau 2003), which as a consequence makes them sparse (Del Genio, Gross, and Bassler 2011). Thus, we expect them to be loosely connected and yield a high number of biconnected components. We can view information regarding biconnected components of these networks in Table 2. The findings are quite impressive, in the sense that the maximum order among biconnected components is significantly smaller than the order of the initial graph. For example, the biggest real RCC8 network, namely, `adm2`, has order 1 733 000, but its biggest biconnected component has a number of just 22 808 vertices.

As (Nikolaou and Koubarakis 2014) suggests, some state-of-the-art reasoners, such as GQR (Gantner, Westphal, and Wölfl 2008), use a matrix to represent a QCN. It would be impossible to store a graph of the order of `adm2` in a matrix as we would need $\sim 3TB$ of memory. Even if memory was not the issue, the complexity of $\diamond$-consistency alone would explode, while the backtracking algorithm that is typically used for tackling non-tractable QCNs and makes use of $\diamond$-consistency as a forward checking step would suffer from an increased search space. Heuristics for the backtracking algorithm would also have a hard time distinguishing between biconnected components. Consider for example a situation where the backtracking algorithm backtracks from an instatiation of a constraint in a biconnected component to an

solver	nuts	nuts_D	adm1	adm1_D	gadm1	gadm1_D
GQR	2.0s	0.1s	4.7E3s	5.2E3s	1.4E4s	1.2s
Pha.	4.0s	0.6s	3.4E3s	3.7E3s	1.0E5s	3.5s
Sar.	0.8s	0.6s	161.5s	137.7s	2.0E3s	3.4s
Pha.$\triangledown$	0.9s	0.6s	98.3s	97.4s	1.1E3s	3.0s

Table 3: Performance comparison based on elapsed time

instantiation of a constraint in a different biconnected component. The constraints are completely unrelated as they belong to different biconnected components, but they might define a huge branch in the search-tree that is spawned by the backtracking algorithm. Proposition 1 allows us to treat the QCNs that correspond to biconnected components completely separately, in a parallel or serial fashion, and avoid the aforementioned bothersome issues.

Evaluation We consider the hard networks `nuts`, `adm1`, and `gadm1` from (Nikolaou and Koubarakis 2014) that comprise $\mathcal{NP}_8$ relations (Renz and Nebel 2001) to utilize the whole reasoning engine of a reasoner. If `name` is the name of a QCN, $\texttt{name}_\text{D}$ denotes the input file that comprises all biconnected QCNs derived from that QCN using Algorithm 1. The experiments were carried out on a computer with an Intel Core 2 Quad Q9400 processor with a CPU frequency of 2.66 GHz per core, 8 GB RAM, and the Precise Pangolin x86_64 OS. GQR (under version 1500) was compiled with gcc/g++ 4.6.3 and Sarissa, Phalanx, and Phalanx$\triangledown$ (Sioutis and Condotta 2014) (all under version 0.2) were run with PyPy 2.4.0, which fully implements Python 2.7.8. For all reasoners, the best performing heuristics were enabled. We chose to reason with the biconnected QCNs in a serial fashion, from smaller to bigger QCN, so as to stress how much more $\diamond$-consistency and the backtracking algorithm that utilizes it along with the heuristics in each reasoner benefit from reasoning with the biconnected QCNs than reasoning with the initial loosely connected QCN, when both approaches are offered the same computational power. Thus, only one CPU core was used in our experiments.

The results are shown in Table 3 and make clear that our simple decomposition scheme aids the performance of each reasoner substantially, with the more apparent case being that of `gadm1` which is inconsistent, as opposed to `nuts` and `adm1` that are consistent. In particular, GQR decides `gadm1` in ~ 4 hours and $\texttt{gadm1}_\text{D}$ in 1.2 seconds, while similar results are also obtained for the other reasoners. When an inconsistency is detected in a biconnected QCN in $\texttt{name}_\text{D}$, each reasoner backtracks only within that QCN, and considers a very small search-tree to either verify or dispute that inconsistency with respect to the search-tree of `name`. Obviously, the time obtained for $\texttt{gadm1}_\text{D}$ is the time that it took each reasoner to serially reason with every biconnected QCN, until it reached an inconsistent QCN (thus, assuring that `gadm1` is also inconsistent by Proposition 1).

Conclusion

We improved the state-of-the-art in checking the satisfiability of large real world QCNs, by proposing a simple decomposition scheme that exploits the loosely connected structure of their underlying graphs. Experimental results showed that our appoach significantly decongests search in state-of-the-art reasoners. On another positive note, our approach easily applies to any reasoner, as it can a priori decompose the input QCN before it is fed to the reasoner.

References

Barabasi, A.-L., and Bonabeau, E. 2003. Scale-Free Networks. *Scientific American* 50–59.

Chmeiss, A., and Condotta, J.-F. 2011. Consistency of Triangulated Temporal Qualitative Constraint Networks. In *ICTAI*.

Dechter, R. 2003. *Constraint processing*. Elsevier Morgan Kaufmann.

Del Genio, C. I.; Gross, T.; and Bassler, K. E. 2011. All Scale-Free Networks Are Sparse. *Phys. Rev. Lett.* 107:178701.

Diestel, R. 2012. *Graph Theory*, volume 173 of *Graduate texts in mathematics*. Springer, 4th edition.

Gantner, Z.; Westphal, M.; and Wölfl, S. 2008. GQR-A Fast Reasoner for Binary Qualitative Constraint Calculi. In *AAAI Workshop on Spatial and Temporal Reasoning*.

Hazarika, S. 2012. *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*. Igi Global.

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

Koubarakis, M.; Kyzirakos, K.; Karpathiotakis, M.; Nikolaou, C.; Sioutis, M.; Vassos, S.; Michail, D.; Herekakis, T.; Kontoes, C.; and Papoutsis, I. 2011. Challenges for Qualitative Spatial Reasoning in Linked Geospatial Data. In *Benchmarks and Applications of Spatial Reasoning Workshop @ IJCAI*.

Ladkin, P. B., and Maddux, R. D. 1994. On binary constraint problems. *JACM* 41:435–469.

Lutz, C., and Milicic, M. 2007. A Tableau Algorithm for DLs with Concrete Domains and GCIs. *JAR* 38:227–259.

Nebel, B., and Bürckert, H. 1995. Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. *JACM* 42:43–66.

Nebel, B. 1997. Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class. *Constraints* 1:175–190.

Nikolaou, C., and Koubarakis, M. 2014. Fast Consistency Checking of Very Large Real-World RCC-8 Constraint Networks Using Graph Partitioning. In *AAAI*.

Randell, D. A.; Cui, Z.; and Cohn, A. 1992. A Spatial Logic Based on Regions and Connection. In *KR*.

Renz, J., and Ligozat, G. 2005. Weak Composition for Qualitative Spatial and Temporal Reasoning. In *CP*.

Renz, J., and Nebel, B. 1999. On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus. *AI* 108:69–123.

Renz, J., and Nebel, B. 2001. Efficient Methods for Qualitative Spatial Reasoning. *JAIR* 15:289–318.

Sioutis, M., and Condotta, J.-F. 2014. Tackling large Qualitative Spatial Networks of scale-free-like structure. In *SETN*.

Sioutis, M., and Koubarakis, M. 2012. Consistency of Chordal RCC-8 Networks. In *ICTAI*.

Sioutis, M.; Salhi, Y.; and Condotta, J. 2015. On the Use and Effect of Graph Decomposition in Qualitative Spatial and Temporal Reasoning. In *SAC*.

Sioutis, M. 2014. Triangulation Vs Graph Partitioning for Tackling Large Real World Qualitative Spatial Networks. In *ICTAI*.