A Practical Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks

Jean-François Condotta*, Ali Mensi*†, Issam Nouaouri‡, Michael Sioutis* and Lamjed Ben Saïd†

*Université Lille-Nord de France - Université d'Artois - CRIL-CNRS UMR 8188
{condotta,mensi,sioutis}@cril.fr
†Université de la Mannouba - SOIE
‡Université Lille-Nord de France - Université d'Artois - LGI2A
issam.nouaouri@univ-artois.fr
§Institut Supérieur de Gestion (ISG) - SOIE
lamjed.bensaid@isg.rnu.tn

***Abstract*—We introduce and study the problem of obtaining a spatial or temporal configuration that maximizes the number of constraints satisfied in a qualitative constraint network (QCN). We call this problem the MAX-QCN problem and prove that it is NP-hard for most of the qualitative calculi. We also propose a complete generic branch and bound algorithm for solving the MAX-QCN problem. This algorithm builds on techniques used in the literature for solving the consistency checking problem and the minimal labeling problem of a given QCN. In particular, we make use of a tractable subclass of relations, a chordal graph provided by a triangulation of the input QCN, and the partial weak composition as a filtering method. The experimentation that we have conducted with QCNs from the Interval Algebra and the Region Connection Calculus shows the interest of our proposed algorithm.**

I. Introduction

Spatial and temporal reasoning is a major field of study in Artificial Intelligence; particularly in Knowledge Representation. 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, but are not limited to, ambient intelligence, dynamic GIS, cognitive robotics, and spatiotemporal design [1]. In this context, an emphasis has been made on qualitative spatial and temporal reasoning, which abstracts from numerical quantities of space and time using qualitative values instead (e.g., earlier, bigger, left of). The conciseness of the representational language used in the qualitative approach provides a promising framework that further boosts research and applications in spatial and temporal reasoning.

The Interval Algebra (IA) [2] and a subset of the Region Connection Calculus (RCC) [3], namely RCC-8, are the dominant Artificial Intelligence approaches for representing and reasoning about qualitative temporal and topological relations respectively. These qualitative calculi use constraints to encode knowledge about the spatial or temporal relationships between entities. Thus, qualitative information can be modelled as a domain-specific variant of a Constraint Satisfaction Problem (CSP) [4]. Infinite domains is the main difference of spatial or temporal CSPs to normal CSPs. For instance, there are infinitely many time points or temporal intervals on the time line and infinitely many regions in a two or three dimensional space. One way of dealing with infinite domains is using constraints over a finite set of binary relations by employing an algebra [5]. This particular type of infinite-domain CSP that makes use of an algebra to handle qualitative constraints can be formulated as a Qualitative Constraint Network (QCN), which comprises a set of spatial or temporal entities V and a mapping C that associates a binary spatial or temporal relation respectively with each pair of entities.

Given a QCN $\mathcal{N}$, we are particularly interested in the MAX-QCN problem, which is the problem of obtaining a spatial or temporal configuration that maximizes the number of constraints satisfied in $\mathcal{N}$. The MAX-QCN is a generalization of the well studied consistency checking problem, which is the problem of determining if a QCN admits a solution. Contrary to the consistency checking problem, the MAX-QCN problem is an optimization problem that can be greatly useful for reasoning about a set of spatial or temporal qualitative constraints representing *inconsistent* information. Handling such inconsistent information can be necessary for a number of applications in Artificial Intelligence such as temporal planning, reasoning about preferences or reasoning with manually annotated corpus containing spatial or temporal information. Consider, for example, a planning problem specified by a QCN of IA representing inconsistent information. In this case, solve MAX-QCN for this set of constraints allows to obtain a temporal planning that best meets the specification given.

The main objective of this paper is to propose an efficient algorithm for solving the MAX-QCN problem. To this end, we elaborate a branch and bound algorithm built on the following techniques that have proven to be useful for the consistency checking problem of a QCN: the use of a tractable class of relations to minimize the width of the search tree, the use of a chordal graph corresponding to a triangulation of a given QCN to reduce the number of constraints processed during search, and the use of the partial closure under weak composition as an inference method to filter out some unfeasible base relations.

The paper is organized as follows. After some preliminaries about temporal and spatial QCNs, we introduce the MAX-QCN problem and state its complexity in the general case. Section IV is devoted to particular sets of optimal consistent scenarios for which some technical results are established. In Section V we describe the algorithm proposed to solve the MAX-QCN problem. We report some experimental results about this algorithm in Section V. Finally, we conclude and give some perspectives for future work.

II. Preliminaries

A (binary) temporal or spatial qualitative calculus considers a domain D to represent temporal or spatial entities respectively and a finite set B of *jointly exhaustive and pairwise disjoint* (JEPD) relations defined on this domain D [5]. The elements of B are called base relations and represent the set of possible configurations between two temporal or spatial entities. Set B contains the identity relation Id, and is closed under the converse operation ($^{-1}$). Indefinite knowledge between two temporal or spatial entities can be described by a relation that corresponds to a union of base relations and is represented by the set containing them. Hence, 2^{B} represents the total set of relations. Given $x, y \in \mathsf{D}$ and $r \in 2^{\mathsf{B}}$, $x\ r\ y$ will denote that x and y satisfy a base relation $b \in r$. The set 2^{B} is equipped with the usual set-theoretic operations (union and intersection), the converse operation, and the weak composition operation (also called algebraic closure). The converse of a relation is the union of the converses of its base relations. The weak composition $\diamond$ of two base relations b and b' belonging to a set of base relations B is the relation of 2^{B} defined by $b \diamond b' = \{b'' : \exists x, y, z \in \mathsf{D}$ such that $x\ b\ y,\ y\ b'\ z$ and $x\ b''\ z\}$. For two relations $r, r' \in 2^{\mathsf{B}}$, $r \diamond r'$ is the relation of 2^{B} defined by $r \diamond r' = \bigcup_{b \in r, b' \in r'} b \diamond b'$. A subclass $\mathcal{A}$ is a subset of relations, *i.e.*, $\mathcal{A} \subseteq 2^{\mathsf{B}}$, closed under intersection, converse, and weak composition. In the sequel, we will assume that a considered subclass contains all the singletons relations defined on B and the universal relation B. Given a relation $r \in 2^{\mathsf{B}}$, $\mathcal{A}(r)$ denotes the smallest relation of $\mathcal{A}$ including r.

As illustration, consider the well-know qualitative calculi of the Interval Algebra (IA) and the Region Connection Calculus (RCC8). IA, introduced by Allen [2], serves for temporal reasoning. IA considers intervals to represent temporal entities and the set of base relations $\mathsf{B}_{\mathsf{IA}} = \{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$. Each base relation of B_{IA} represents a particular ordering of the four endpoints of two intervals on the timeline (see Figure 1a). RCC8 [3] serves for reasoning about spatial configurations, and considers spatial regions which can be interpreted as non-empty regular closed subsets of some topological space and the set of base relations $\mathsf{B}_{\mathsf{RCC8}} = \{DC, EC, TPP, NTPP, PO, EQ, TPPi, NTPPi\}$. These base relations allow us to specify how regions and their interiors are related to each other (see Figure 1b).

A. A Qualitative Constraint Network (QCN)

Temporal or spatial information about the relative positions of a set of entities can be represented by a Qualitative Constraint Network (QCN). A QCN is a pair formed by a set of variables and a set of constraints. Each variable represents

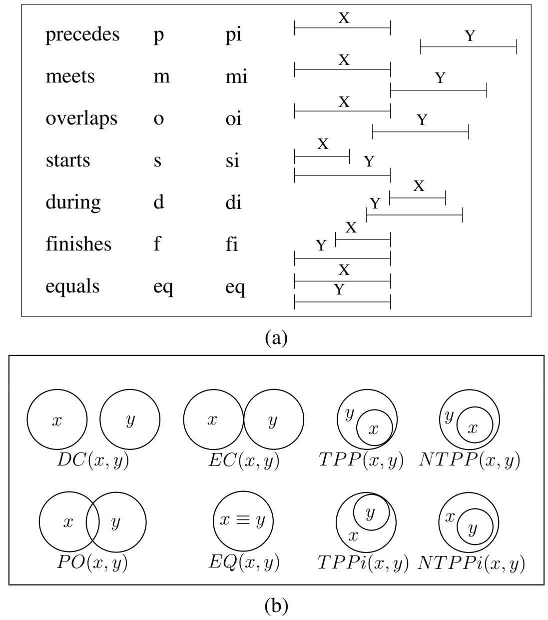


Fig. 1: (a) The base relations of IA (b) and RCC8 (b).

a temporal or spatial entity. Each constraint is defined by a relation of 2^{B} and specifies the set of acceptable qualitative configurations between two entities.

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 $C(v, v') \in 2^{\mathsf{B}}$ with each pair (v, v') of $V \times V$. C is such that $C(v, v) \subseteq \{\mathsf{Id}\}$ and $C(v, v') = (C(v', v))^{-1}$.

Given a QCN $\mathcal{N} = (V, C)$ and $v, v' \in V$, $\mathcal{N}[v, v']$ will denote the relation $C(v, v')$ in what follows. $\mathcal{N}_{[v,v']/r}$ with $r \in 2^{\mathsf{B}}$ is the QCN $\mathcal{N}'$ defined by $\mathcal{N}'[v, v'] = r$, $\mathcal{N}'[v', v] = r^{-1}$, and $\mathcal{N}'[v'', v'''] = \mathcal{N}[v'', v''']\ \forall (v'', v''') \in (V \times V) \setminus \{(v, v'), (v', v)\}$. Given a set of variables V, $\bot_V$ will denote the particular QCN over V where each constraint is defined by the empty relation $\emptyset$, and $\top_V$ will denote the QCN (V, C) where $C(v, v) = \{\mathsf{Id}\}$ and $C(v, v') = \mathsf{B}$ for all $v, v' \in V$. Given a QCN $\mathcal{N} = (V, C)$ we have the following definitions [6]: $\mathcal{N}$ is said to be *trivially inconsistent* iff $\exists v, v' \in V$ with $C(v, v') = \emptyset$. An *instantiation* of V is a mapping σ defined from V to the domain D. A *solution* σ of $\mathcal{N}$ is an instantiation of V such that for every pair (v, v') of variables in V, $(\sigma(v), \sigma(v'))$ satisfies $C(v, v')$, *i.e.*, there exists a base relation $b \in C(v, v')$ such that b is defined by $(\sigma(v), \sigma(v'))$. $\mathcal{N}$ is *consistent* iff it admits a solution. Two QCNs are *equivalent* iff they admit the same set of solutions. A *sub-QCN* $\mathcal{N}'$ of $\mathcal{N}$, denoted by $\mathcal{N}' \subseteq \mathcal{N}$, is a QCN (V, C') such that $C(v, v') \subseteq C(v, v')\ \forall v, v' \in V$. A scenario $\mathcal{S}$ is a QCN whose contraints are defined by singleton relations, *i.e.*, $|C(v, v')| = 1$ for all pairs of variables of the QCN. A (consistent) scenario $\mathcal{S}$ of $\mathcal{N}$ is a (consistent) scenario which is a sub-QCN of $\mathcal{N}$. In the sequel, we assume that for the qualitative calculi considered, the consistency of a scenario can be decided in polynomial time. The set of consistent scenarios of $\mathcal{N}$ will be denoted by $[[\mathcal{N}]]$. $\mathcal{N}$ is

$\diamond$-*consistent* or *closed under weak composition* iff $\forall v, v', v'' \in V$, $C(v,v') \subseteq C(v,v'') \diamond C(v'',v')$. The *closure* under weak composition of $\mathcal{N}$, denoted by $\diamond(\mathcal{N})$, is the greatest $\diamond$-consistent sub-QCN of $\mathcal{N}$. This closure can be computed in $O(n^3)$ time, where n is the number of variables of $\mathcal{N}$.

As illustration, let us consider the two QCNs $\mathcal{N}_0$ and $\mathcal{N}_1$ of the IA depicted in Figure 2. In each of the graphs, a variable is represented by a node, and a constraint by an arc labeled with the associated relation; there is no arc going from v to v' when there is an arc going from v' to v or when $v = v'$. We can check that $\mathcal{N}_0$ is an inconsistent QCN and $\mathcal{N}_1$ a consistent one. A solution σ of $\mathcal{N}_1$ is depicted in Figure 2c, with the corresponding consistent scenario shown in Table I.

Given a subclass $\mathcal{A}$ and a QCN $\mathcal{N}$, the closure of $\mathcal{N}$ *w.r.t.* $\mathcal{A}$ denoted by $\mathcal{A}(\mathcal{N})$ is the QCN (V, C') defined by : $C'(v,v') = \mathcal{A}(C(v,v'))$ for all $v, v' \in V$. Given a QCN $\mathcal{N}' = (V, C')$, $\mathcal{N} \cup \mathcal{N}'$ denotes the QCN $\mathcal{N}'' = (V, C'')$ where $C''(v,v') = C(v,v') \cup C'(v,v')$ for all $v, v' \in V$.

Given two (undirected) graphs $G = (V, E)$ and $G' = (V', E')$, G is a subgraph of G', denoted by $G \subseteq G'$, iff $V \subseteq V'$ and $E \subseteq E'$. A graph $G = (V, E)$ is a *chordal (or triangulated) graph* iff each of its cycles of length > 3 has a chord, *i.e.*, an edge joining two vertices that are not adjacent in the cycle. The constraint graph of a QCN $\mathcal{N} = (V, C)$ is the graph (V, E), denoted by $\mathsf{G}(\mathcal{N})$, for which we have that $(v, v') \in E$ iff $C(v,v') \neq \mathsf{B}$. Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, $\mathcal{N}$ is partially $\diamond$-consistent *w.r.t.* graph G or $\overset{\diamond}{G}$-consistent [7] iff for $\forall (v,v'), (v,v''), (v'',v') \in E$, $C(v,v') \subseteq C(v,v'') \diamond C(v'',v')$. The closure under $\overset{\diamond}{G}$-consistency of $\mathcal{N}$, denoted by $\overset{\diamond}{G}(\mathcal{N})$, is the greatest sub-QCN of $\mathcal{N}$ which is $\overset{\diamond}{G}$-consistent. Note that $\overset{\diamond}{G}(\mathcal{N})$ is equivalent to $\mathcal{N}$. In what follows, $\overset{\diamond}{G}$-consistency will be said to be complete for a subclass $\mathcal{A}$, if for any QCN $\mathcal{N}$ defined on $\mathcal{A}$ and a chordal graph G such that $\mathsf{G}(\mathcal{N}) \subseteq G$, we have that $\mathcal{N}$ is consistent iff $\overset{\diamond}{G}(\mathcal{N})$ is not trivially inconsistent.

	v_0	v_1	v_2	v_3	v_4	v_5
v_0	$\{eq\}$	$\{bi\}$	$\{mi\}$	$\{s\}$	$\{d\}$	$\{bi\}$
v_1	$\{b\}$	$\{eq\}$	$\{d\}$	$\{b\}$	$\{d\}$	$\{f\}$
v_2	$\{m\}$	$\{di\}$	$\{eq\}$	$\{m\}$	$\{d\}$	$\{oi\}$
v_3	$\{si\}$	$\{bi\}$	$\{mi\}$	$\{eq\}$	$\{d\}$	$\{bi\}$
v_4	$\{di\}$	$\{di\}$	$\{di\}$	$\{di\}$	$\{eq\}$	$\{oi\}$
v_5	$\{b\}$	$\{fi\}$	$\{o\}$	$\{b\}$	$\{o\}$	$\{eq\}$

Tab. I: A consistent scenario $\mathcal{S}$ of IA.

III. THE MAXIMUM CONSISTENCY PROBLEM OF QCNS

In this section we introduce the maximum consistency problem of QCNs, denoted by MAX-QCN, which corresponds to the well known problems of MAX-SAT [8] and MAX-CSP [9] in SAT and constraint programming respectively. Given a QCN $\mathcal{N}$ over a set of variables V, the MAX-QCN problem is the problem of finding a consistent scenario over V which maximizes the number of constraints satisfied in $\mathcal{N}$.

Before formally introducing the MAX-QCN problem, we introduce two operators, namely, $\#\mathsf{disjC}$ and $\#\mathsf{satC}$, which will allow measuring the number of unsatisfied constraints and the number of satisfied constraints respectively between a scenario and a QCN. Given two QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$ defined on the same set of variables V, $\#\mathsf{disjC}(\mathcal{N}, \mathcal{N}')$ will denote the integer corresponding to the number of constraints of $\mathcal{N}$ and $\mathcal{N}'$ that do not share a base relation. More formally, $\#\mathsf{disjC}(\mathcal{N}, \mathcal{N}') = \frac{1}{2}.|\{(v,v') \in V \times V : v \neq v' \text{ and } C(v,v') \cap C'(v,v') = \emptyset\}| + |\{v \in V : C(v,v) \cap C'(v,v) = \emptyset\}|$. Note that the factor $\frac{1}{2}$ is used for taking into account the fact that $C(v,v')$ and $C(v',v)$ represent the same constraint since $C(v,v') = C(v',v)^{-1}$ for any QCN $\mathcal{N} = (V, C)$ and $(v,v') \in V \times V$. We can extend the operator $\#\mathsf{disjC}$ to instantiations of QCNs by considering their associated scenarios. Given an instantiation σ and a QCN $\mathcal{N}$ over a set of variables V, $\#\mathsf{disjC}(\sigma, \mathcal{N}) = \#\mathsf{disjC}(\mathcal{S}_\sigma, \mathcal{N})$, where $\mathcal{S}_\sigma$ is the unique scenario over V corresponding to σ. Note that $\#\mathsf{disjC}(\sigma, \mathcal{N})$ corresponds to the number of constraints of $\mathcal{N}$ that are not satisfied by σ. Given a consistent scenario $\mathcal{S} = (V, C')$ over V and a QCN $\mathcal{N} = (V, C)$, we introduce the operator $\#\mathsf{satC}$ to allow measuring the number of constraints of $\mathcal{N}$ satisfied by $\mathcal{S}$ in the following manner: $\#\mathsf{satC}(\mathcal{S}, \mathcal{N}) = \frac{1}{2}.|\{(v,v') \in V \times V : v \neq v' \text{ and } C'(v,v') \subseteq C(v,v')\}| + |\{v \in V : C'(v,v) \subseteq C(v,v)\}|$. For an instantiation σ over V, $\#\mathsf{satC}(\sigma, \mathcal{N}) = \#\mathsf{satC}(\mathcal{S}_\sigma, \mathcal{N})$, where $\mathcal{S}_\sigma$ is the consistent scenario corresponding to σ.

As example, let us consider the two QCNs $\mathcal{N}_0$ and $\mathcal{N}_1$ of IA, and the instantiation σ of $\mathcal{N}_1$ depicted in Figure 2 along with its associated consistent scenario shown in Table I. We can check that $\#\mathsf{disjC}(\mathcal{N}_1, \mathcal{N}_0) = \#\mathsf{disjC}(\mathcal{S}, \mathcal{N}_0) = \#\mathsf{disjC}(\sigma, \mathcal{N}_0) = 2$ and $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}_1) = \#\mathsf{disjC}(\sigma, \mathcal{N}_1) = 0$. Also, $\#\mathsf{satC}(\mathcal{S}, \mathcal{N}_0) = \#\mathsf{satC}(\sigma, \mathcal{N}_0) = 19$ and $\#\mathsf{satC}(\mathcal{S}, \mathcal{N}_1) = \#\mathsf{satC}(\sigma, \mathcal{N}_1) = 21$.

Concerning the operator $\#\mathsf{disjC}$ we have :

Proposition 1: Let $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$ be two QCNs. We have:

(a) $\#\mathsf{disjC}(\mathcal{N}', \mathcal{N}) = \#\mathsf{disjC}(\mathcal{N}, \mathcal{N}')$;

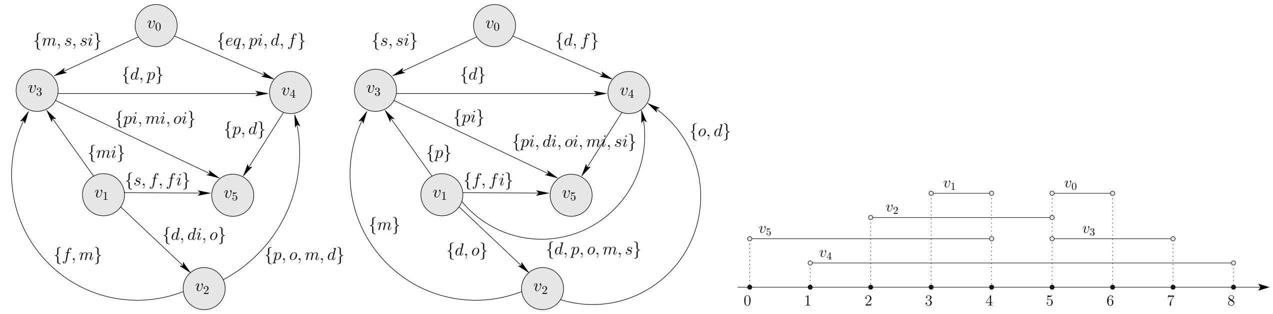
(b) For any QCN $\mathcal{N}'' = (V, C'')$ such that $\mathcal{N}'' \subseteq \mathcal{N}'$ we have that $\#\mathsf{disjC}(\mathcal{N}'', \mathcal{N}) \geq \#\mathsf{disjC}(\mathcal{N}', \mathcal{N})$.

Proposition 2: Let $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$ be two QCNs such that for all $(v,v') \in V \times V$, $C'(v,v') \subseteq C(v,v')$ or $C'(v,v') \cap C(v,v') = \emptyset$. We have: for all $\mathcal{S} \in [[\mathcal{N}']]$, $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) = \#\mathsf{disjC}(\mathcal{N}', \mathcal{N})$.

Now, it is time to formally define the MAX-QCN problem:

Definition 2: (MAX-QCN) Given a QCN $\mathcal{N} = (V, C)$, the MAX-QCN problem consists of finding a consistent scenario $\mathcal{S}$ over V such that $\#\mathsf{satC}(\mathcal{S}, \mathcal{N}) = \max\{\#\mathsf{satC}(\mathcal{S}', \mathcal{N}) : \mathcal{S}' \text{ is a consistent scenario over } V\}$.

Note that we could propose an alternative definition of the MAX-QCN problem by considering as solution of the MAX-QCN problem a consistent scenario $\mathcal{S}$ over V that minimizes the unsatisfied contraints of $\mathcal{N}$, *i.e.*, a scenario $\mathcal{S}$ such that $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) = \min\{\#\mathsf{disjC}(\mathcal{S}', \mathcal{N}) : \mathcal{S}' \text{ is a consistent scenario over } V\}$. Moreover, we could consider instantiations over V rather than consistent scenarios over V. For the purpose of this paper, the aforementioned definition would be equivalent to Definition 2, as for the calculi considered here one can build a solution out of a



(a) $\mathcal{N}_0$ (b) $\mathcal{N}_1$ (c) σ

Fig. 2: Two QCNs $\mathcal{N}_0$ and $\mathcal{N}_1$ of IA and a solution σ of $\mathcal{N}_1$.

consistent scenario in polynomial time and vice versa. The MAX-QCN problem is an optimization problem; the related decision problem, denoted by KMAX-QCN in the sequel, can be defined in the following way:

Definition 3: (KMAX-QCN) Given a QCN $\mathcal{N} = (V, C)$ and an integer $k \geq 0$, the problem KMAX-QCN consists of deciding whether there exists or not a consistent scenario $\mathcal{S}$ over V such that $\#\mathsf{satC}(\mathcal{S}, \mathcal{N}) \geq k$.

We have the following complexity result:

Theorem 1: Let $\mathcal{Q}$ be a qualitative calculus for which the consistency checking problem is NP-Complete. We have: the KMAX-QCN problem for $\mathcal{Q}$ is NP-complete.

Proof: The KMAX-QCN problem for $\mathcal{Q}$ is in NP as we can, in PTIME, guess a scenario over V, check if it is a consistent scenario over V, and check if $\#\mathsf{satC}(\mathcal{S}, \mathcal{N}) \geq k$. Further, we can solve the consistency checking problem of a QCN $\mathcal{N} = (V, C)$ by determining if the KMAX-QCN problem has a solution for $\mathcal{N}$ and the integer $k = \frac{1}{2}.|V|.(|V| - 1) + |V|$. Hence, $\mathcal{Q}$ is NP-hard as the consistency checking problem for $\mathcal{Q}$ is an NP-hard problem. ■

Corollary 1: The KMAX-QCN problem is NP-complete for the Interval Algebra and the RCC8 calculus.

IV. Optimal consistent scenarios

Given two QCNs $\mathcal{N}$ and $\mathcal{N}'$ defined on the same set of variables, and an integer α, in the sequel we will consider the consistent scenarios of $\mathcal{N}'$ which satisfy the maximal number of constraints of $\mathcal{N}$ and which are disjoint from $\mathcal{N}$ in strictly less than α constraints. We call these scenarios the optimal consistent scenarios of $\mathcal{N}'$ for $\mathcal{N}$ *w.r.t.* the integer α. This set of scenarios will be denoted by $[\mathcal{N}']^{<\alpha}_{\mathcal{N}}$ and is formally defined in the following way:

Definition 4: Let $\mathcal{N}$ and $\mathcal{N}'$ be two QCNs defined over V and α an integer such that $\alpha > 0$. $[\mathcal{N}']^{<\alpha}_{\mathcal{N}}$ is the subset of consistent scenarios of $\mathcal{N}'$ defined by $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} = \{\mathcal{S} \in [[\mathcal{N}']] : \#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) < \alpha$ and there is no $\mathcal{S}' \in [[\mathcal{N}']]$ with $\#\mathsf{disjC}(\mathcal{S}', \mathcal{N}) < \#\mathsf{disjC}(\mathcal{S}, \mathcal{N})\}$.

Note that, given a QCN $\mathcal{N} = (V, C)$, the solutions of the MAX-QCN problem for $\mathcal{N}$ correspond to the set of consistent scenarios over V belonging to the set $[\top_V]^{<\alpha}_{\mathcal{N}}$ with $\alpha = \frac{1}{2}.|V|.(|V| - 1) + |V| + 1$. Next, we will state some properties about the set of optimal consistent scenarios. These technical results will be used in the next section to prove the soundness and the completeness of the proposed algorithm for solving the MAX-QCN problem. The first result shows that the optimal consistent scenarios of $\mathcal{N}'$ for $\mathcal{N}$ *w.r.t.* the integer α can be obtained by considering a QCN having the same set of solutions as $\mathcal{N}'$:

Proposition 3: Let $\mathcal{N}, \mathcal{N}', \mathcal{N}''$ be three QCNs over a set of variables V and α an integer such that $\alpha \geq 0$. We have: if $\mathcal{N}'$ and $\mathcal{N}''$ are equivalent QCNs then $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} = [\mathcal{N}'']^{<\alpha}_{\mathcal{N}}$.

The following result characterizes a case for which we have no optimal consistent scenario:

Proposition 4: Let $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$ be two QCNs and α an integer such that $\alpha \geq 0$. If $\#\mathsf{disjC}(\mathcal{N}', \mathcal{N}) \geq \alpha$ then $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} = \emptyset$.

Proof: Suppose that $\#\mathsf{disjC}(\mathcal{N}', \mathcal{N}) \geq \alpha$. Let $\mathcal{S}$ be a consistent scenario of $\mathcal{N}'$. $\mathcal{S}$ is a subQCN of $\mathcal{N}'$, hence, from Proposition 1 (b) we know that $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) \leq \#\mathsf{disjC}(\mathcal{N}', \mathcal{N})$. It follows that $\mathcal{S}$ cannot belong to the set $[\mathcal{N}']^{<\alpha}_{\mathcal{N}}$. Consequently, we have that $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} = \emptyset$ since it cannot contain any consistent scenario of $\mathcal{N}'$. ■

Now, we characterize a case where the optimal consistent scenarios of $\mathcal{N}'$ for $\mathcal{N}$ *w.r.t.* the integer α correspond exactly to the set of the consistent scenarios of $\mathcal{N}'$.

Proposition 5: Let $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$ be two QCNs and α an integer such that $\alpha \geq 0$ and: (a) $\#\mathsf{disjC}(\mathcal{N}', \mathcal{N}) < \alpha$, (b) for all $(v, v') \in V \times V$, $C'(v, v') \subseteq C(v, v')$ or $C'(v, v') \cap C(v, v') = \emptyset$. We have: $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} = [[\mathcal{N}']]$.

Proof: By definition, we know that $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} \subseteq [[\mathcal{N}']]$. Now, we prove that $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} \supseteq [[\mathcal{N}']]$. By hypothesis on $\mathcal{N}'$ and Proposition 2, we know that for all $\mathcal{S} \in [[\mathcal{N}']]$, $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) = \#\mathsf{disjC}(\mathcal{N}', \mathcal{N})$. Moreover, as $\#\mathsf{disjC}(\mathcal{N}', \mathcal{N}) < \alpha$, we can assert that $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) < \alpha$. From this, for all $\mathcal{S} \in [[\mathcal{N}']]$, $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) < \alpha$, there is no $\mathcal{S}' \in [[\mathcal{N}']]$ such that $\#\mathsf{disjC}(\mathcal{S}', \mathcal{N}) < \#\mathsf{disjC}(\mathcal{S}, \mathcal{N})$. We can conclude that $\mathcal{S} \in [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$. ■

Proposition 6: Let $\mathcal{N}, \mathcal{N}', \mathcal{N}''$ be three QCNs over a set of variables V such that $[[\mathcal{N}'']] \subseteq [[\mathcal{N}']]$ and α an integer such that $\alpha > 0$. We have: for all scenarios $\mathcal{S} \in [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$, if $\mathcal{S} \in$

$[[\mathcal{N}'']]$ then $\mathcal{S} \in [\mathcal{N}'']^{<\alpha'}_{\mathcal{N}}$ for any integer $\alpha' > \#\mathsf{disjC}(\mathcal{S}, \mathcal{N})$.

Proposition 7: Let $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C)$ be two QCNs, α an integer such that $\alpha \geq 0$, and $(\mathcal{N}'_1, \ldots, \mathcal{N}'_k)$ a sequence of QCNs such that $[[\mathcal{N}']] = [[\mathcal{N}'_1]] \cup \ldots \cup [[\mathcal{N}'_k]]$. We have that $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} \subseteq [\mathcal{N}'_1]^{<\alpha}_{\mathcal{N}} \cup \ldots \cup [\mathcal{N}'_k]^{<\alpha}_{\mathcal{N}}$.

Proof: Let $\mathcal{S} \in [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$. Since $[[\mathcal{N}']] = [[\mathcal{N}'_1]] \cup \ldots \cup [[\mathcal{N}'_k]]$ there exists an integer $i \in \{1, \ldots, k\}$ such that $\mathcal{S} \in [[\mathcal{N}'_i]]$. Moreover, we can note that $[[\mathcal{N}'_i]] \subseteq [[\mathcal{N}']]$ and $\alpha > \#\mathsf{disjC}(\mathcal{S}, \mathcal{N})$. From Prop. 6, we can assert that $\mathcal{S} \in [\mathcal{N}'_i]^{<\alpha}_{\mathcal{N}}$. ■

The following result will be fundamental in the proof of the completeness of the proposed algorithm for MAX-QCN:

Proposition 8: Let $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C)$ be two QCNs, α an integer such that $\alpha > 0$, and $(\mathcal{N}'_1, \ldots, \mathcal{N}'_k)$ a sequence of QCNs such that $[[\mathcal{N}']] = [[\mathcal{N}'_1]] \cup \ldots \cup [[\mathcal{N}'_k]]$. By defining the sequence of integers $(\alpha_1, \ldots, \alpha_k)$ with $k \geq 1$ in the following manner: $\alpha_1 = \alpha$; for $i \in \{2, \ldots, k\}$, $\alpha_i = \alpha_{i-1}$ if $[\mathcal{N}'_{i-1}]^{<\alpha_{i-1}}_{\mathcal{N}} = \emptyset$ else $\alpha_i = \#\mathsf{disjC}(\mathcal{S}_{i-1}, \mathcal{N})$, where $\mathcal{S}_{i-1}$ is some consistent scenario of $[\mathcal{N}'_{i-1}]^{<\alpha_{i-1}}_{\mathcal{N}}$; we have:
(a) if for all $i \in \{1, \ldots, k\}$, $[\mathcal{N}']^{<\alpha_i}_{\mathcal{N}} = \emptyset$ then $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} = \emptyset$.
(b) if for some $i \in \{1, \ldots, k\}$, $[\mathcal{N}']^{<\alpha_i}_{\mathcal{N}} \neq \emptyset$ then $[\mathcal{N}'_l]^{<\alpha_l}_{\mathcal{N}} \subseteq [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$ with $l = \max\{i : i \in \{1, \ldots, k\}$ and $[\mathcal{N}']^{<\alpha_i}_{\mathcal{N}} \neq \emptyset\}$.

Proof: (Sketch) We give just a partial proof for **(b)**. Suppose that there exists an integer $i \in \{1, \ldots, k\}$ such that $[\mathcal{N}']^{<\alpha_i}_{\mathcal{N}} \neq \emptyset$. Let $\mathcal{S} \in [\mathcal{N}'_l]^{<\alpha_l}_{\mathcal{N}}$. We can show that $\alpha_i \leq \alpha$ for all $i \in \{1, \ldots, k\}$. Hence, we have $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) < \alpha_l \leq \alpha$. Moreover, we can assert that $\mathcal{S} \in [[\mathcal{N}']]$ since $\mathcal{N}'_l \subseteq \mathcal{N}'$. Suppose by contradiction, that $\mathcal{S} \notin [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$. As $\#\mathsf{disjC}(\mathcal{S}, \mathcal{N}) < \alpha$ and $\mathcal{S} \in [[\mathcal{N}']]$, there exists a scenario $\mathcal{S}'$ different from $\mathcal{S}$ such that $\mathcal{S}' \in [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$ and $\#\mathsf{disjC}(\mathcal{S}', \mathcal{N}) < \#\mathsf{disjC}(\mathcal{S}, \mathcal{N})$. Note that, as $\mathcal{S}' \in [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$, we have that $\mathcal{S}' \in [[\mathcal{N}']]$. Moreover, since $[[\mathcal{N}']] = [[\mathcal{N}'_1]] \cup \ldots \cup [[\mathcal{N}'_k]]$, there exists $m \in \{1, \ldots, k\}$ such that $\mathcal{S}' \in [[\mathcal{N}'_m]]$. By considering the three possible cases $m \in \{1, \ldots, l-1\}$, $m = l$, and $m \in \{l+1, \ldots, k\}$, we can show that such a scenario $\mathcal{S}'$ cannot exist. Consequently $\mathcal{S} \in [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$. ■

V. A COMPLETE ALGORITHM FOR MAX-QCN

To solve the MAX-QCN problem, we present here the algorithm MaxQCN. Function MaxQCN has two parameters, the first one being a QCN $\mathcal{N} = (V, C)$ for which we aim to characterize a consistent scenario over V corresponding to a solution of the MAX-QCN problem for $\mathcal{N}$, and the second one being a subclass $\mathcal{A}$ for which $\overset{\diamond}{_G}$-consistency is complete (for the consistency checking problem). Function MaxQCN uses two main auxiliary functions called MaxQCNAux and ExtractConsScen.

Function MaxQCNAux takes as parameters two QCNs $\mathcal{N}$ and $\mathcal{N}'$ defined on a same set of variables V, an integer α, a graph $G = (V, E)$ intended to be a triangulated graph of the graph of constraints of $\mathcal{N}$, and a subclass $\mathcal{A}$ for which $\diamond$-consistency is complete. The aim of this function is to characterize a consistent subQCN of $\mathcal{N}'$, that we denote by $\mathcal{N}_{res}$, whose consistent scenarios are optimal consistent scenarios of $\mathcal{N}'$ for $\mathcal{N}$ with respect to α. In other terms, $[[\mathcal{N}']] \subseteq [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$ and $[[\mathcal{N}']] \neq \emptyset$. Moreover, the constraints of this sub-QCN $\mathcal{N}''$ are defined by relations of the subclass $\mathcal{A}$. In the case where $[\mathcal{N}']^{<\alpha}_{\mathcal{N}}$ is an empty set, function MaxQCNAux will return the trivially inconsistent QCN $\perp_V$. The general structure of MaxQCNAux is very close to that of the functions proposed in [7], [10] for solving the consistency checking problem of a QCN and to that of the function proposed in [6] for solving the minimal labeling problem (the problem consisting of determining the feasible base relations of a QCN). Function MaxQCNAux consists of a branch-and-bound method exploring the QCN $\mathcal{N}'$ to characterize one of its sub-QCNs having the required properties. The search stops when the whole search tree is explored. Graph G, given as parameter, is used to narrow the set of constraints to be handled and processed during search. In a first step, MaxQCNAux uses the closure under $\overset{\diamond}{_G}$-consistency as a filtering method (line 1) to prune some non-feasible relations of the QCN $\mathcal{N}'$ and possibly to detect an inconsistency of the initial QCN $\mathcal{N}'$. In the case where an inconsistency is characterized (line 3) we know that no optimal consistent scenario exists and the QCN $\perp_V$ is returned (line 4). In the contrary case, a second test is realized at line 5 to check that the number of constraints of $\mathcal{N}$ already unsatisfied by taking into account $\mathcal{N}'$ is less than the maximal permitted number of unsatisfied constraints, *i.e.*, α. In the case where the number of non-overlapping constraints of $\mathcal{N}$ and $\mathcal{N}'$ is greater than α, we can assert that there is no optimal scenario of $\mathcal{N}'$ for $\mathcal{N}$ with respect to α. Again, in this case the QCN $\perp_V$ is returned (line 6). In the next step, a pair of variables (v, v') corresponding to an edge of G not already handled is selected. The corresponding constraint $C'(v, v')$ must have the following characteristics: it must not be defined by a relation of the subclass $\mathcal{A}$, or some base relations of $C'(v, v')$ must not belong to $C(v, v')$ and some base relations of $C'(v, v')$ must belong to $C(v, v')$. In the case where such a pair does not exist, we will see in the sequel that the closure of $\mathcal{N}'$ with respect to the subclass $\mathcal{A}$ is a consistent QCN whose consistent scenarios are optimal consistent scenarios of the initial QCN $\mathcal{N}'$ (given as parameter) for $\mathcal{N}$ with respect to α. This closure is returned at line 9. In the contrary case, at lines 11 and 12, the relation defining $C'(v, v')$ is split into non-empty subrelations of $\mathcal{A}$. This splitting is made in two steps: at line 11 the relation $C'(v, v') \setminus C(v, v')$ is split, then at line 12 the relation $C'(v, v') \cap C(v, v')$ is split. Hence, each obtained subrelation belongs to $\mathcal{A}$, and does not contain both a base relation of $C'(v, v') \setminus C(v, v')$ and a base relation of $C'(v, v') \cap C(v, v')$. Then, the constraint $C'(v, v')$ is iteratively instantiated with each of these subrelations. Due to the previous splitting, note that each future constraint $C'(v, v')$ will be defined by a relation of $\mathcal{A}$ which is either part of $C(v, v')$ or distinct from $C(v, v')$. The search continues through recursive calls of MaxQCNAux. After each call, in the case where a QCN containing *better* optimal consistent scenarios for $\mathcal{N}$ is found, the result is updated (line 16) along with the number of unsatisfied constraints of $\mathcal{N}$. We have the following result:

Proposition 9: Let $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$ be

Function MaxQCN($\mathcal{N}$,$\mathcal{A}$)

in : A QCN $\mathcal{N} = (V, C)$, a subclass $\mathcal{A}$.
output : A consistent scenario solution of the MAX-QCN problem for $\mathcal{N}$.

1 **begin**
2 $G = (V, E) \leftarrow \mathsf{Triangulation}(\mathsf{G}(\mathcal{N}))$;
3 $\alpha \leftarrow \frac{1}{2}.|V|.(|V| - 1) + |V|$;
4 $\mathcal{N}' \leftarrow \mathsf{MaxQCNAux}(\mathcal{N}, \top_V, \alpha, G, \mathcal{A})$;
5 **return** ExtractConsScen($\mathcal{N}'$);

Function MaxQCNAux($\mathcal{N}$,$\mathcal{N}'$,α,G,$\mathcal{A}$)

in : A QCN $\mathcal{N} = (V, C)$, a QCN $\mathcal{N}' = (V, C')$, an integer α, a graph $G = (V, E)$, a subclass $\mathcal{A}$.
output : A QCN.

1 **begin**
2 $\mathcal{N}' \leftarrow \diamond_G(\mathcal{N}')$;
3 **if** *$\mathcal{N}'$ is trivially inconsistent* **then**
4 **return** $\bot_V$;
5 **if** $\#\mathsf{disjC}(\mathcal{N}, \mathcal{N}') \geq \alpha$ **then**
6 **return** $\bot_V$;
7 Select $(v, v') \in E$ such that (v, v') has not already been selected and $(C'(v, v') \notin \mathcal{A}$ or $((C'(v, v') \setminus C(v, v') \neq \emptyset)$ and $(C'(v, v') \cap C(v, v') \neq \emptyset)))$;
8 **if** *such a pair does not exist* **then**
9 **return** $\mathcal{A}(\mathcal{N}')$;
10 $\mathcal{N}_{res} \leftarrow \bot_V$;
11 $\mathcal{R} \leftarrow \mathsf{split}_{\mathcal{A}}(C'(v, v') \setminus C(v, v'))$;
12 $\mathcal{R}' \leftarrow \mathsf{split}_{\mathcal{A}}(C(v, v') \cap C'(v, v'))$;
13 **foreach** $r \in \mathcal{R} \cup \mathcal{R}'$ **do**
14 $\mathcal{N}'' \leftarrow \mathsf{MaxQCNAux}(\mathcal{N}, \mathcal{N}'_{[v,v']/r}, \alpha, G, \mathcal{A})$;
15 **if** $\mathcal{N}'' \neq \bot_V$ **then**
16 $\mathcal{N}_{res} \leftarrow \mathcal{N}''$;
17 $\alpha \leftarrow \#\mathsf{disjC}(\mathcal{N}, \mathcal{N}'')$;
18 **return** $\mathcal{N}_{res}$;

two QCNs, $\mathcal{A} \subseteq 2^{\mathsf{B}}$ a subclass for which $\diamond_G$-consistency is complete, $G = (V, E)$ a triangulated graph such that $\mathsf{G}(\mathcal{N}) \subseteq G$, and α an integer such that $\alpha > 0$. Let $\mathcal{N}_{res}$ be the QCN returned by MaxQCNAux with parameters $\mathcal{N}$, $\mathcal{N}'$, α, G, $\mathcal{A}$. We have: if $\mathcal{N}_{res} = \bot_V$ then $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} = \emptyset$, else $\mathcal{N}_{res}$ is a consistent sub-QCN of $\mathcal{N}'$ defined on $\mathcal{A}$ such that $[[\mathcal{N}_{res}]] \subseteq [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$.

Proof: (Sketch) Let n be defined by $n = |\{(v, v') \in E$: (v, v') not already selected, and $(C'(v, v') \notin \mathcal{A}$, or $((C'(v, v') \setminus C(v, v') \neq \emptyset)$ and $(C'(v, v') \cap C(v, v') \neq \emptyset)))\}|$. We can make a proof by induction on n. Here we will prove the property uniquely for the base case where $n = 0$.

- Case $n = 0$**.** Let $\mathcal{M} = \diamond_G(\mathcal{N}')$. Consider in an exhaustive manner the two possible following cases:

• $\mathcal{M}$ is trivially inconsistent or $\#\mathsf{disjC}(\mathcal{N}, \mathcal{M}) \geq \alpha$. Function MaxQCNAux returns $\bot_V$ (line 4 or line 6). We have that $[[\mathcal{M}]] = \emptyset$ or $\#\mathsf{disjC}(\mathcal{N}, \mathcal{M}) \geq \alpha$. Hence, $[\mathcal{M}]^{<\alpha}_{\mathcal{N}} = \emptyset$. As $\mathcal{M}$ and $\mathcal{N}'$ are equivalent, we have that $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} = \emptyset$.

• $\mathcal{M}$ is not trivially inconsistent and $\#\mathsf{disjC}(\mathcal{N}, \mathcal{N}') < \alpha$. Denote by $\mathcal{M}'$ the QCN $\mathcal{A}(\mathcal{M})$ (the closure of $\mathcal{M}$ w.r.t. the subclass $\mathcal{A}$). $\mathcal{M}'$ is returned by MaxQCNAux at line 9. As $\mathcal{M}$ is not trivially inconsistent and $\diamond_G$-consistent, we can show that $\mathcal{M}' = \mathcal{A}(\mathcal{M})$ is also not trivially inconsistent and $\diamond_G$-consistent. Since $\mathcal{M}'$ is defined on $\mathcal{A}$, G is a triangulation of $\mathsf{G}(\mathcal{M}')$, and $\diamond_G$-consistency is complete for $\mathcal{A}$, we can assert that $\mathcal{M}'$ is a consistent QCN. Now, consider the QCN $\mathcal{M}''$ defined as follows: for all $(v, v') \in V \times V$, if $(v, v') \notin E$ or (v, v') is a constraint not already handled then $\mathcal{M}''[v, v'] = \mathcal{M}[v, v']$, otherwise $\mathcal{M}''[v, v'] = r$ where r is the subrelation of $\mathcal{A}$ used to instantiate the constraint between v and v' at line 14 of function MaxQCNAux. By construction of $\mathcal{M}''$, we can show that the constraints of $\mathcal{M}''$ are defined by relations belonging to $\mathcal{A}$, and $\mathcal{M} \subseteq \mathcal{M}' \subseteq \mathcal{M}'' \subseteq \mathcal{N}'$. Hence, $\mathcal{M}' \subseteq \mathcal{N}'$. Moreover, again by construction of $\mathcal{M}''$, we can show that $\forall v, v' \in V$ we have that $\mathcal{M}''[v, v'] \subseteq \mathcal{N}[v, v']$ or $\mathcal{M}''[v, v'] \cap \mathcal{N}[v, v'] = \emptyset$. Consequently, as $\mathcal{M}' \subseteq \mathcal{M}''$, for all $v, v' \in V$ we have that $\mathcal{M}'[v, v'] \subseteq \mathcal{N}[v, v']$ or $\mathcal{M}'[v, v'] \cap \mathcal{N}[v, v'] = \emptyset$. From this and the fact that $\mathcal{M} \subseteq \mathcal{M}'$, we can deduce that $\#\mathsf{disjC}(\mathcal{N}, \mathcal{M}') = \#\mathsf{disjC}(\mathcal{N}, \mathcal{M}) < \alpha$. From Proposition 5, we have that $[[\mathcal{M}']] = [\mathcal{M}']^{<\alpha}_{\mathcal{N}}$. As $\mathcal{M} \subseteq \mathcal{M}' \subseteq \mathcal{N}'$ and $\mathcal{M}$ is an equivalent QCN to $\mathcal{N}'$, we can assert that $\mathcal{M}'$ and $\mathcal{N}'$ are equivalent. Consequently, from Proposition 3, we have that $[\mathcal{M}']^{<\alpha}_{\mathcal{N}} = [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$. Hence, $[[\mathcal{M}']] = [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$. From all this, we can conclude that $\mathcal{M}'$ is a consistent subQCN of $\mathcal{N}'$ defined on $\mathcal{A}$ such that $[[\mathcal{M}']] \subseteq [\mathcal{N}']^{<\alpha}_{\mathcal{N}}$. Moreover, note that $[\mathcal{N}']^{<\alpha}_{\mathcal{N}} \neq \emptyset$ since $[[\mathcal{M}']] \neq \emptyset$.

- Case $n > 0$**.** We can prove the property by using the different propositions established in Section IV, more particularly by using Proposition 7 and Proposition 8. ■

Function ExtractConsScen($\mathcal{N}$)

in : A consistent QCN $\mathcal{N} = (V, C)$.
output : A consistent scenario of $\mathcal{N}$.

1 **begin**
2 Select $(v, v') \in (V \times V)$ such that $|C(v, v')| > 1$;
3 **if** *such a pair does not exist* **then**
4 **return** $\mathcal{N}$;
5 Select a base relation $b \in C(v, v')$ such that $|C(v, v')| > 1$;
6 **for** $b \in C(v, v')$ **do**
7 $\mathcal{N}' \leftarrow \diamond(\mathcal{N}_{[v,v']/\{b\}})$;
8 **if** $\mathcal{N}' \neq \bot_V$ **then**
9 break;
10 **return** ExtractConsScen($\mathcal{N}'$);

One can note that, whatever the parameters given to function MaxQCNAux, this function eventually terminates since for each recursive call there is a new constraint which is handled. Roughly, we can have at most $\frac{1}{2}.|V|.(|V| - 1).|\mathsf{B}|$ recursive calls, where V is the set of variables of the input QCNs.

Now, consider function ExtractConsScen. This function takes as parameter a QCN $\mathcal{N} = (V, C)$ intended to be a QCN defined on a subclass $\mathcal{A}$ for which $\diamond$-consistency is complete. This function returns a consistent scenario of the QCN $\mathcal{N}$ given as parameter. For this purpose, at each step it considers a constraint defined by a non-singleton relation (line 5) and selects a base relation for which we are ensured to have a consistent scenario satisfying it. This choice is realized by use of the $\diamond$-consistency method (line 7). The base relation selected is used to define the constraint selected. The process continues through a recursive call of ExtractConsScen. The function stops after at most $\frac{1}{2}.|V|.(|V|-1).|\mathsf{B}|$ recursive calls. Each call realizes at most $|B|$ computations of closures under $\diamond$-consistency.

Proposition 10: Let $\mathcal{N}$ be a consistent QCN defined on a subclass $\mathcal{A}$ for which $\diamond$-consistency is complete. We have that ExtractConsScen with $\mathcal{N}$ as parameter returns a consistent scenario of $\mathcal{A}$.

Function ExtractConsScen is a generic function running for subclasses for which $\diamond$-consistency is complete. Note that for some subclasses we can have other more specific functions that can perform better for obtaining a similar result.

Function MaxQCN is the main function of the proposed algorithm. It takes two parameters: a QCN $\mathcal{N}$ for which we want to solve the MAX-QCN problem and a subclass $\mathcal{A}$ for which $\overset{\diamond}{G}$-consistency is complete. A triangulation of the constraint graph of $\mathcal{N}$ is computed at line 2. Along with the triangulated graph G of $\mathcal{N}$, $\top_V$, $\alpha = \frac{1}{2}.|V|.(|V|-1) + |V| + 1$, and the subclass $\mathcal{A}$ are used as parameters of a call to MaxQCNAux at line 4. We know that $[\top_V]_{\mathcal{N}}^{<\frac{1}{2}.|V|.(|V|-1)+|V|+1}$ corresponds exactly to the set of solutions of the MAX-QCN problem for $\mathcal{N}$, in particular, it corresponds to a non-empty set of consistent scenarios over V, hence, we are ensured from Proposition 9 that the call of MaxQCNAux returns a QCN $\mathcal{N}'$ which is a consistent QCN defined on $\mathcal{A}$ such that $[[\mathcal{N}']] \subseteq [\top_V]_{\mathcal{N}}^{<\frac{1}{2}.|V|.(|V|-1)+|V|+1}$. Using function ExtractConsScen, a consistent scenario of this QCN is extracted and returned at line 5. We have:

Theorem 2: Let $\mathcal{N}$ be a QCN and $\mathcal{A} \subseteq 2^{\mathsf{B}}$ a subclass for which $\diamond$-consistency is complete. Function MaxQCN, with $\mathcal{N}$ and $\mathcal{A}$ as parameters, returns a solution of the MAX-QCN problem for $\mathcal{N}$.

A naive variant of the algorithm presented in this section, that we will use in the experimental evaluation to follow to stress the efficiency of our approach, is obtained by considering the set of singleton relations as a splitset and the complete graph as a chordal graph in all cases.

VI. EXPERIMENTAL EVALUATION

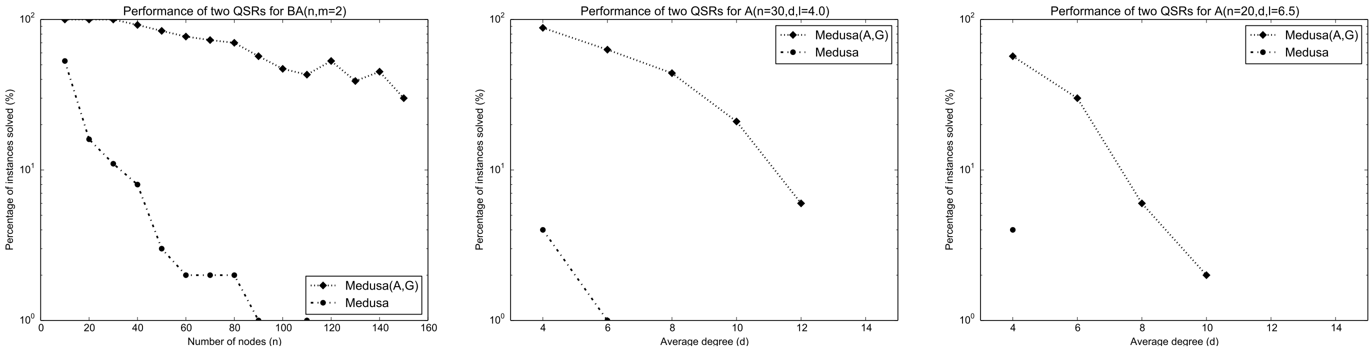
In this section, we make a preliminary evaluation of the algorithms that we discussed in Section V, namely, the naive algorithm that operates on a complete constraint graph trying to instatiate a base relation to every constraint, and the state-of-the-art algorithm that makes use of a chordal constraint graph and a tractable subclass of relations. Both of these algorithms are implemented under the hood of a novel reasoner called Medusa. In particular, Medusa differentiates its functionality based on particular invocation flag options, and is thus able to simulate both considered algorithms. In what follows, Medusa will denote the naive algorithm, and Medusa$(G, \mathcal{A})$ the state-of-the-art algorithm where G is a chordal graph and $\mathcal{A}$ some tractable subclass of relations.

Technical Specifications: The experiments were carried out on a computer with an Intel Core i7-2820QM processor with a CPU frequency of 2.30 GHz per core, 8 GB of RAM, and the Trusty Tahr x86_64 OS (Ubuntu Linux). Medusa was implemented in Python and run with PyPy 2.2.1 (http://pypy.org/), which fully implements Python 2.7. Only one of the CPU cores was used.

Dataset and Measures: We considered random datasets consisting of RCC-8 networks generated by the BA(n, m) model [11], the use of which is well motivated in [12], [13], and RCC-8 and IA networks generated by the standard A(n, d, l) model [14], used extensively in literature. In short, BA(n, m) creates random scale-free-like networks of size n and a preferential attachment value m, and A(n, d, l) creates random networks of size n, degree d, and an average number l of base relations per edge. For model BA(n, m) the average number of base relations per edge defaults to $|\mathsf{B}|/2$, where B is the set of base relations of a qualitative constraint language as a reminder.

Regarding model BA(n, m), we considered 100 unsatisfiable RCC-8 networks for each size n between 10 and 150 nodes with a 10-node step and a preferential attachment value of $m = 2$. For this specific value of m and for the network sizes considered, the networks of model BA(n, m) lie within the *phase transition* region, where it is equally possible for networks to be satisfiable or unsatisfiable, thus, they are harder to solve [13]. Unsatisfiable network instances were randomly filtered out of a large number of 10 000 instances that were created in the phase transition region. Regarding model A(n, d, l), we considered 100 unsatisfiable networks for each average node degree d between 4 and 14 with a 2-degree step and for each of the calculi of RCC-8 and IA. For RCC-8 we set a node size of $n = 30$ and an average number of base relations per edge of $l = 4.0$, and for IA we set values of $n = 20$ and $l = 6.5$ respectively. For this specific range of node degrees d, the networks of model A(n, d, l) lie within the *phase transition* region, similarly to the case of the scale-free networks described earlier. Again, we only used unsatisfiable network instances that were randomly filtered out of a large number of 10 000 instances that were created in the phase transition region.

Our experimentation involves two measures which we describe as follows. The first measure considers the performance of Medusa and Medusa$(G, \mathcal{A})$ based on the percentage (%) of network instances that they are able to solve within a 60-sec timeout per instance. We found this timeout to be adequate for separating network instances that were able to be solved within a reasonable amount of time, and network instances that were impossible to be dealt with even after several minutes of CPU-intensive reasoning. The second measure concerns the average splitting ratio $\rho = s/s'$ for the reasoner considered, where s' denotes the number of base relations of an initial constraint in the network $\mathcal{N}'$ that is given as input to function MaxQCNAux and s denotes the number of splits that are generated from s' with respect to some splitset, such as a tractable subclass of relations $\mathcal{A}$. The lesser the value of ρ, the better performance we have. Clearly, if we use the set of singleton relations as a splitset, as is the case with Medusa, we have that $\rho = 1.0$.



(a) Performance comparison for random scale-free RCC-8 networks.

(b) Performance comparison for random standard RCC-8 networks.

(c) Performance comparison for random standard IA networks.

Fig. 3: Performance comparisons for RCC-8 and IA networks.

Results: In what follows, and with respect to Medusa$(G, \mathcal{A})$, we note that we used the class of Horn relations as a maximal tractable subclass of relations $\mathcal{A}$ both for RCC-8 and IA respectively (denoted for example by $\hat{\mathcal{H}}_8$ for RCC-8 [14]), and a triangulation procedure to obtain a chordal constraint graph G based on the *maximum cardinality search algorithm* [15], as described in [13].

Regarding model BA(n, m), the experimental results are shown in Figure 3a. Medusa$(G, \mathcal{A})$ is able to solve 100% of the smaller RCC-8 network instances and around 40% of the bigger ones in the given 60-sec timeout per instance. On the other hand, Medusa is only able to solve around 50% of the smaller RCC-8 network instances and none of the bigger ones in the given timeout. For Medusa$(G, \mathcal{A})$, the average splitting ratio ρ fluctuated around ~ 0.36 for all network sizes.

Regarding model A(n, d, l), the experimental results for RCC-8 and IA are shown in Figure 3b and Figure 3c respectively. Medusa$(G, \mathcal{A})$ is able to solve around 90% of the sparser RCC-8 network instances and around 10% of the denser ones in the given 60-sec timeout per instance. On the other hand, Medusa is only able to solve around 4% of the sparser RCC-8 network instances and none of the denser ones in the given timeout. In fact, Medusa becomes completely impractical for networks of an average degree $d > 4$. For Medusa$(G, \mathcal{A})$, the average splitting ratio ρ ranged from a value of ~ 0.36 for the sparser RCC-8 networks to a value of ~ 0.46 for the denser ones. The results for IA are qualitatively similar, although it should be noted that the performance of both Medusa and Medusa$(G, \mathcal{A})$ deteriorates at a faster pace than the one obtained for RCC-8. This is merely explained by the fact that IA is a bigger and hence harder calculus to deal with than RCC-8. For Medusa$(G, \mathcal{A})$, the average splitting ratio ρ ranged from a value of ~ 0.41 for the sparser IA networks to a value of ~ 0.59 for the denser ones.

VII. Conclusion

In this paper we introduced and studied the MAX-QCN problem, which is the problem of obtaining a consistent scenario maximizing the number of constraints satisfied in a given QCN. To solve this problem we proposed a generic algorithm taking advantage of a class of relations for which $\diamond_G$-consistency is complete for the consistency checking problem. The experimentation that we have conducted on qualitative constraint networks of IA and RCC8 shows the interest of our approach. Future work consists of using other methods, in particular methods of local search, and comparing the behavior of our algorithm with these different methods. Other research perspectives consist of studying encodings of the MAX-QCN problem into the partial MAX-SAT problem and, using our approach to propose algorithm to solve merging problems such as the one handled in [16].

References

[1] M. Bhatt, H. Guesgen, S. Wölfl, and S. Hazarika, "Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions," *Spatial Cognition & Computation*, vol. 11, pp. 1–14, 2011.

[2] J.-F. Allen, "An interval-based representation of temporal knowledge," in *IJCAI*, 1981.

[3] D. Randell, Z. Cui, and A. Cohn, "A Spatial Logic Based on Regions and Connection," in *KR*, 1992.

[4] U. Montanari, "Networks of constraints: Fundamental properties and application to picture processing," *Information Sciences*, vol. 7, pp. 95–132, 1974.

[5] P. B. Ladkin and R. D. Maddux, "On Binary Constraint Problems," *JACM*, vol. 41, pp. 435–469, 1994.

[6] N. Amaneddine, J. Condotta, and M. Sioutis, "Efficient Approach to Solve the Minimal Labeling Problem of Temporal and Spatial Qualitative Constraints," in *IJCAI*, 2013.

[7] A. Chmeiss and J.-F. Condotta, "Consistency of Triangulated Temporal Qualitative Constraint Networks," in *ICTAI*, 2011.

[8] J. Argelich, C. M. Li, F. Manyà, and J. Planes, "The first and second max-sat evaluations," *JSAT*, vol. 4, pp. 251–278, 2008.

[9] E. C. Freuder and R. J. Wallace, "Partial constraint satisfaction," *AIJ*, vol. 58, pp. 21–70, 1992.

[10] M. Sioutis and M. Koubarakis, "Consistency of Chordal RCC-8 Networks," in *ICTAI*, 2012.

[11] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science (New York)*, vol. 286, pp. 509–512, 1999.

[12] M. Sioutis and J.-F. Condotta, "Tackling large Qualitative Spatial Networks of scale-free-like structure," in *SETN*, 2014.

[13] M. Sioutis, J.-F. Condotta, and M. Koubarakis, "An Efficient Approach for Tackling Large Real World Qualitative Spatial Networks," *IJAIT*, 2015, in press.

[14] J. Renz and B. Nebel, "Efficient Methods for Qualitative Spatial Reasoning," *JAIR*, vol. 15, pp. 289–318, 2001.

[15] R. E. Tarjan and M. Yannakakis, "Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs," *SIAM J. Comput.*, vol. 13, pp. 566–579, 1984.

[16] J. Condotta, S. Kaci, P. Marquis, and N. Schwind, "A syntactical approach to qualitative constraint networks merging," in *Proceedings of LPAR-17*, 2010, pp. 233–247.