Local Search for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks

Jean-François Condotta[1], Ali Mensi[1,3], Issam Nouaouri[2], Michael Sioutis[1], and Lamjed Ben Saïd[3]

[1]Université Lille-Nord de France - Université d'Artois - CRIL-CNRS UMR 8188
{condotta,mensi,sioutis}@cril.fr
[2]Université Lille-Nord de France - Université d'Artois - LGI2A
issam.nouaouri@univ-artois.fr
[3]Institut Suprieure de Gestion (ISG) - SOIE
lamjed.bensaid@isg.rnu.tn

Abstract. We focus on the recently introduced problem of maximizing the number of satisfied constraints in a qualitative constraint network (QCN), called the MAX-QCN problem. We present a particular local search method for solving the MAX-QCN problem of a given QCN, which involves first obtaining a partial scenario $\mathcal{S}$ of that QCN and then exploring neighboring scenarios that are obtained by disconnecting a variable of $\mathcal{S}$ and repositioning it appropriately. The experimentation that we have conducted with qualitative constraint networks from the Interval Algebra shows the interest and promise of our approach for maximizing satisfiability in qualitative spatial and temporal constraint networks.

1 Introduction

Qualitative spatial and temporal reasoning (QSTR) is a major field of study in Knowledge Representation that abstracts from numerical quantities of space and time by using qualitative descriptions instead (e.g., precedes, contains, left of). The representational languages used in the qualitative approach have increasingly gained a lot of attention during the last decades, as they have the advantage of being conceptually concise and sufficiently expressive for a variety of applications in many areas, such as ambient intelligence, dynamic GIS, cognitive robotics, and spatiotemporal design [2, 8, 14].

The Interval Algebra (IA) [1] and a subset of the Region Connection Calculus (RCC) [13], namely RCC8, are the dominant calculi in QSTR for representing qualitative temporal and spatial information respectively. In particular, IA encodes knowledge about the temporal relations between intervals in the timeline (see Figure 1a), and RCC8 encodes knowledge about the spatial relations between regions in some topological space. In addition to IA and RCC8, numerous other qualitative calculi have been proposed in the literature for representing spatial and temporal information [10].

The problem of reasoning about qualitative spatial or temporal information can be modelled as a qualitative constraint network (QCN), i.e., a network comprising constraints corresponding to qualitative spatial or temporal relations between spatial or temporal variables respectively. In this paper, we focus on a recently introduced problem in the context of QSTR, called the MAX-QCN problem [5]. Given a QCN $\mathcal{N}$, the MAX-QCN problem is the problem of obtaining a spatial or temporal configuration that maximizes the number of satisfied constraints in $\mathcal{N}$. Solving the MAX-QCN problem is clearly at least as difficult as solving the consistency problem, which is NP-complete in general. To solve this optimization problem, the authors in [5] propose a branch and bound algorithm based on state of the art techniques for checking the consistency of a QCN, viz., the use of a triangulation of the constraint graph of the considered QCN to reduce the number of constraints to be treated, the use of a tractable subclass of relations to reduce the width of the search tree, and the use of partial $\diamond$-consistency to efficiently propagate constraints and consequently prune non-feasible base relations during search. In another approach, the authors in [6] view the MAX-QCN problem as a partial maximum satisfiability problem (PMAX-SAT) and propose two related families of encodings. Each proposed PMAX-SAT encoding is based on, what is called, a forbidden covering with regard to the composition table of the considered qualitative calculus. Intuitively, a forbidden covering is a compact set of triples that express all the non-feasible configurations for three spatial or temporal entities.

We follow another approach for solving the MAX-QCN problem of a QCN and present a particular local search method [11] which involves first obtaining a partial atomic refinement $\mathcal{S}$ of that QCN and then exploring neighboring atomic refinements that are obtained by disconnecting a variable of $\mathcal{S}$ and repositioning it appropriately. The search for the best neighboring atomic refinement is guided by a combination of heuristics for minimizing the number of unsatisfied constraints in a given neighboring atomic refinement of a QCN, a tabu list for excluding certain already considered atomic refinements or atomic refinements that are known to not be candidates for best neighboring atomic refinement, and a particular restart policy to deal with local minima.

The paper is organized as follows. Some preliminaries on QSTR and the MAX-QCN problem are made in Section 2. In Section 3, we introduce the notion of neighborhoods of partial scenarios which will be used in the proposed method and an algorithm to compute them. In Section 4, we define the local search based method proposed to solve the MAX-QCN problem, namely, the method QLS. In Section 5, we report some experimental results about QLS. Finally, we conclude and give some perspectives for future works.

2 Preliminaries

A (binary) spatial or temporal qualitative calculus [10] considers a domain D of spatial or temporal entities respectively and a finite set B of *jointly exhaustive and pairwise disjoint* (JEPD) relations defined on that domain called base relations. Each base relation of B represents a particular configuration between two spatial or temporal entities. The set B contains the identity relation Id, and is closed under the converse operation ($^{-1}$). A (complex) relation corresponds to a union of base relations and is represented by the set containing them. Hence, 2^{B} represents the 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, the complement operation and the weak composition operation. The converse of a relation is the union of the converses of its base relations. The complement of a relation r, denoted by $\overline{r}$, is the relation $\{b \in \mathsf{B} : b \notin r\}$. The weak composition $\diamond$ of two base relations $b, b' \in \mathsf{B}$ 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 $r \diamond r' = \bigcup_{b \in r, b' \in r'} b \diamond b'$. In the sequel, $\widehat{\mathsf{B}}$ will denote the smallest subset of 2^{B} which contains the singleton relations of 2^{B} and the universal relation and, which is closed under the operations $^{-1}$, $\diamond$ and $\cap$.

As an illustration, consider the well known temporal qualitative calculus introduced by Allen [1] and called the Interval Algebra (IA). Allen represents the temporal entities by the intervals of the line and considers a set of 13 base relations $\mathsf{B}_{\mathsf{IA}} = \{\mathsf{eq}, \mathsf{p}, \mathsf{pi}, \mathsf{m}, \mathsf{mi}, \mathsf{o}, \mathsf{oi}, \mathsf{s}, \mathsf{si}, \mathsf{d}, \mathsf{di}, \mathsf{f}, \mathsf{fi}\}$ represented in Figure 1a.

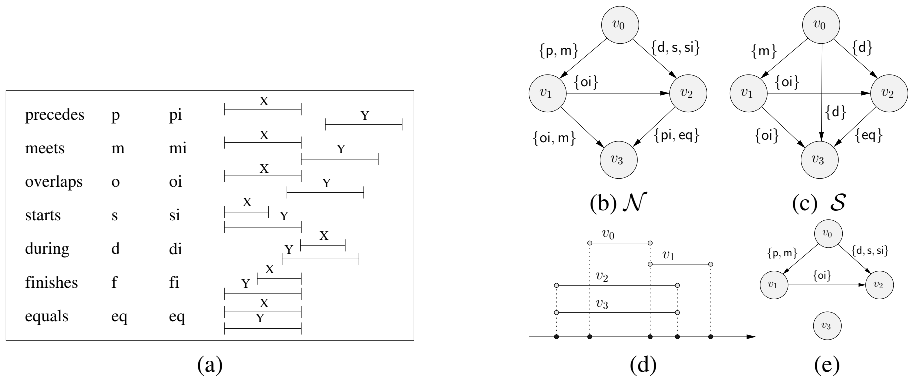

Fig. 1: The base relations of IA (a), a consistent QCN $\mathcal{N}$ of IA (b), a consistent scenario $\mathcal{S}$ of $\mathcal{N}$ (c), a solution of $\mathcal{N}$ and $\mathcal{S}$ (d), the QCN $\mathcal{N}^{\uparrow v_3}$, *i.e.* the relaxation of $\mathcal{N}$ *w.r.t.* v_3 (e). Notice that for the represented QCNs, the constraint between two variables v and v' is not represented when it is equal to B or $v = v'$ or when the constraint between v' and v is already represented.

Spatial or temporal information about a set of entities can be represented by a qualitative constraint network (QCN), which is a pair of a set of variables and a set of constraints. Each constraint is defined by a relation of 2^{B} and specifies the set of acceptable qualitative configurations between two spatial or temporal variables. Formally, a QCN is defined as follows: a QCN is a pair $\mathcal{N} = (V, C)$ where V is a non-empty finite set of variables, and C is a mapping that associates a relation $C(v, v') \in 2^{\mathsf{B}}$ with each pair (v, v') of $V \times V$. Further, mapping 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 $v, v' \in V$, the relation $C(v, v')$ will also be denoted by $\mathcal{N}[v, v']$.

Concerning a QCN $\mathcal{N} = (V, C)$, we have the following definitions. A *solution* σ of $\mathcal{N}$ is a valuation σ of each variable V by an element of D such that for every pair (v, v') of variables in V, $(\sigma(v), \sigma(v'))$ satisfies a base relation belonging to the relation $C(v, v')$. $\mathcal{N}$ is *consistent* iff it admits a solution. $\mathcal{N}$ will be said trivially inconsistent iff one of its constraints is defined by the empty relation. 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 that each contraint is defined by a singleton relation, *i.e.*, a relation defined by exactly one base relation. A scenario $\mathcal{S}$ of $\mathcal{N}$ is a scenario which is a sub-QCN of $\mathcal{N}$. Given a variable $v \in V$, the relaxation of $\mathcal{N}$ *w.r.t.* v, denoted by $\mathcal{N}^{\uparrow v}$ is the QCN $\mathcal{N} = (V, C')$ defined by : for all $v', v'' \in V$, $C'(v', v'') = \mathsf{B}$ if $v' \neq v''$ and, $v' = v$ or $v'' = v$, $C'(v', v'') = C(v', v'')$ else. In the sequel $\mathcal{N}_{[v,v']/r}$ with a relation $r \in 2^{\mathsf{B}}$, will denote the QCN defined on V corresponding to $\mathcal{N}$ for which the relation defining constraint between v and v' has been substitued by the relation r.

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 [7]. 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 [4] iff for $\forall (v, v'), (v, v''), (v'', v') \in E$, we have that $C(v, v') \subseteq C(v, v'') \diamond C(v'', v')$. The closure under $\overset{\diamond}{G}$-consistency of $\mathcal{N}$, denoted by $\overset{\diamond}{G}(\mathcal{N})$, is the greatest $\overset{\diamond}{G}$-consistent sub-QCN of $\mathcal{N}$. This closure can be computed in $O(\delta|E|)$

time [4, 15], where δ is the maximum degree of G. Note that if $\mathsf{G}(\mathcal{N}) \subseteq G$, $\diamond_G(\mathcal{N})$ is equivalent to $\mathcal{N}$, *i.e.* has the same solutions than $\mathcal{N}$.

Given a graph $G = (V, E)$, a *partial* scenario *w.r.t.* G, also called G-scenario, is a QCN (V, C) such that $C(v, v) = \{\mathsf{Id}\}$ for all $v \in V$, $C(v, v') = \mathsf{B}$ for all $(v, v') \notin E$, and $|C(v, v')| = 1$ for all $(v, v') \in E$. As illustration, let us consider the inconsistent QCN $\mathcal{N}'$ represented in Figure 2a. Its constraint graph $\mathsf{G}(\mathcal{N}')$ is represented in Figure 2b. Moreover, a triangulated graph G such that $\mathsf{G}(\mathcal{N}') \subseteq G$ is illustrated in Figure 2c. The four QCNs $\mathcal{S}_0$, $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}_3$ in Figure 2 are four $\diamond_G$-consistent (and consistent) G-scenarios.

Now, we highlight a property useful in the sequel :

Definition 1. *The partial $\diamond$-consistency will be said complete for $\widehat{\mathsf{B}}$ iff for any triangulated graph $G = (V, E)$ and any $\diamond_G$-consistent $\mathcal{N}$ such that $\mathcal{N}[v, v'] = \mathsf{B}$ for any $(v, v') \in E$ and $\mathcal{N}[v, v'] \in \widehat{\mathsf{B}}$ for any $(v, v') \notin E$ we have $\mathcal{N}$ which is a consistent* QCN.

Notice that the partial $\diamond$-consistency is complete $\widehat{\mathsf{B}}$ for many qualitative calculi [9, 15], in particular for IA and RCC8.

Given a QCN $\mathcal{N} = (V, C)$, the MAX-QCN problem is the problem of finding a consistent scenario over V that minimizes the number of unsatisfied constraints in $\mathcal{N}$ (or maximizes the number of satisfied constraints in $\mathcal{N}$). In order to more formally define the MAX-QCN problem we introduce the binary operator α which takes as parameters two QCNs and returns the number of non overlapping constraints of these two QCNs. Formally, given two $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$, $\alpha(\mathcal{N}, \mathcal{N}')$ is the integer defined by $\alpha(\mathcal{N}, \mathcal{N}') = \frac{1}{2}.|\{(v, v') \in V \times V : v \neq v' \text{ and } C(v, v') \cap C'(v, v') = \emptyset\}|$. As illustration, by considering again the QCNs in Figure 2, we have $\alpha(\mathcal{S}_0, \mathcal{N}') = \alpha(\mathcal{S}_1, \mathcal{N}') = 3$, $\alpha(\mathcal{S}_2, \mathcal{N}') = 2$ and $\alpha(\mathcal{S}_3, \mathcal{N}') = 4$. Given a QCN $\mathcal{N} = (V, C)$, a solution of the MAX-QCN problem for $\mathcal{N}$ is a consistent scenario $\mathcal{S}$ on V, said optimal scenario of $\mathcal{N}$, such that there is no consistent scenario $\mathcal{S}'$ on V with $\alpha(\mathcal{S}, \mathcal{N}) > \alpha(\mathcal{S}', \mathcal{N})$. The notion of optimal scenario can been extended to the partial scenarios in a direct manner: given a QCN $\mathcal{N} = (V, C)$, an optimal G-scenario of $\mathcal{N}$ is a consistent G-scenario $\mathcal{S}$ such that $G = (V, E)$ is a graph such that there is no consistent G-scenario $\mathcal{S}'$ with $\alpha(\mathcal{S}, \mathcal{N}) > \alpha(\mathcal{S}', \mathcal{N})$ and such that G is a triangulated graph (V, E) with $\mathsf{G}(\mathcal{N}) \subseteq G$. Notice that all consistent scenario of an optimal G-scenario of a QCN $\mathcal{N}$ is an optimal scenario of $\mathcal{N}$. From [5, 6], we have:

Property 1. *Let $\mathcal{Q}$ be a qualitative calculus for which the partial $\diamond$-consistency is complete for $\widehat{\mathsf{B}}$ and $\mathcal{N} = (V, C)$ a* QCN *in $\mathcal{Q}$. For any triangulated graph $G = (V, E)$ such that $\mathsf{G}(\mathcal{N}) \subseteq G$ and $\diamond_G$-consistent G-scenario $\mathcal{S}$ we have : (1) $\mathcal{S}$ is an optimal G-scenario of $\mathcal{N}$, (2) any consistent scenario of $\mathcal{S}$ is an optimal scenario of $\mathcal{N}$ and (3) a consistent scenario of $\mathcal{S}$ can be computed in polynomial time.*

The method that we will define in the sequel is adapted for QCNs of a qualitative calculus $\mathcal{Q}$ that has the aforementioned property, *i.e.* a qualitative calculus $\mathcal{Q}$ for which the partial $\diamond$-consistency is complete for $\widehat{\mathsf{B}}$. From this, the method can consider partial scenarios rather than complete scenarios. The useful to consider partial scenarios is to render the treatment more fast by discarding some constraints of the considered QCNs.

3 Neighborhood of Partial Consistent Scenarios

The proposed search method that will be detailed in the next section, moves from partial consistent scenario to partial consistent scenario until a shut-off criterion is reached. Given a consistent partial scenario $\mathcal{S}$ *w.r.t.* a graph G, the candidate partial scenario to be considered in the future step will be called neighbors of $\mathcal{S}$ and the set they comprise will be

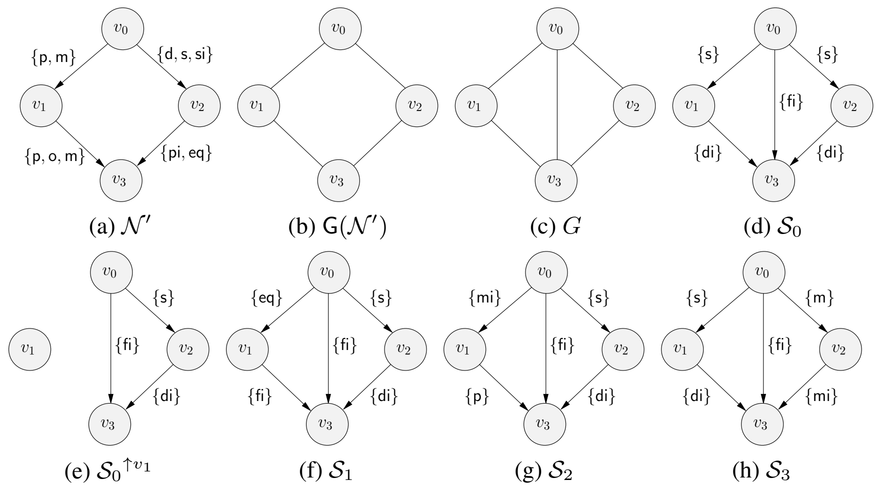

Fig. 2: An inconsistent QCN $\mathcal{N}'$ of IA (a), the constraint graph of $\mathcal{N}'$, *i.e.* $\mathsf{G}(\mathcal{N}')$ (b), a triangulation G of $\mathsf{G}(\mathcal{N}')$ (c), a consistent G-scenario $\mathcal{S}_0$ of $\mathcal{N}'$ (d), the relaxation of $\mathcal{S}$ *w.r.t.* variable v_1 (e). Three neighbors $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ of $\mathcal{S}$ *w.r.t.* G (f, g, and h respectively).

denoted by $\mathsf{Nb}(\mathcal{S}, G)$. Intuitively, a neighbor of the consistent G-scenario $\mathcal{S}$ is a consistent G-scenario different from $\mathcal{S}$ that can be obtained by disconnecting a variable of $\mathcal{S}$ and repositioning it. A neighbor of $\mathcal{S}$ *w.r.t.* G is a consistent G-scenario of a relaxation of $\mathcal{S}$ with respect to one of its variables. The set of neighbors of a partial scenario is defined as:

Definition 2. *Let $\mathcal{S} = (V, C)$ be a consistent G-scenario, where $G = (V, E)$ is a graph. The set of neighbors of $\mathcal{S}$* w.r.t. *G is the set:*

$$\mathsf{Nb}(\mathcal{S}, G) = \bigcup_{v \in V} \{\mathcal{S}' : \mathcal{S}' \neq \mathcal{S} \text{ and } \mathcal{S}' \text{ is a consistent } G\text{-scenario of } \mathcal{S}^{\uparrow v}\}.$$

As illustration, let us consider the QCN $\mathcal{N}' = (V, C)$ in Figure 2a, the graph $G = (V, E)$ in Figure 2c, and the consistent G-scenario $\mathcal{S}_0$ in Figure 2d. Among the neighbors of $\mathcal{S}_0$ *w.r.t.* graph G, we have the three partial scenarios $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ (depicted in Figures 2f, 2g, and 2h respectively). Note that $\mathcal{S}_1$ and $\mathcal{S}_2$ are consistent G-scenarios of the relaxation of $\mathcal{S}_0$ *w.r.t.* v_1 (Figure 2e), whereas $\mathcal{S}_3$ results from the relaxation of $\mathcal{S}_0$ *w.r.t.* v_2.

At each step of the proposed method, among the neighbors of the current partial scenario, the future chosen partial scenario is selected among the best neighbors, *i.e.*, the neighbors satisfying a maximal number of constraints of the considered QCN and not belonging to a particular set of partial scenarios that can be seen as a tabu list. We will present an efficient algorithm to compute this set of best neighbors; prior to this, we define that set:

Definition 3. *Let $\mathcal{N} = (V, C)$ be a* QCN*, $\mathcal{S} = (V, C')$ a consistent G-scenario, where $G = (V, E)$ is a graph, and $\mathcal{T}$ a set of G-scenarios. The set of partial scenarios* $\mathsf{BestNb}(\mathcal{N}, \mathcal{S}, G, \mathcal{T})$ *is defined by:* $\mathsf{BestNb}(\mathcal{N}, \mathcal{S}, G, \mathcal{T}) = \{\mathcal{S}' \in \mathsf{Nb}(\mathcal{S}, G) \setminus \mathcal{T}$: *there exists no* $\mathcal{S}'' \in \mathsf{Nb}(\mathcal{S}, G) \setminus \mathcal{T}$ *such that* $\alpha(\mathcal{N}, \mathcal{S}'') < \alpha(\mathcal{N}, \mathcal{S}')\}$.

Now, we present function bestNeighbors, which allows computing the set of best neighbors of a partial scenario. Function bestNeighbors receives four parameters, namely, a QCN $\mathcal{N} = (V, C)$ for which we want to solve the MAX-QCN problem, a triangulation $G = (V, E)$ of $\mathsf{G}(\mathcal{N})$, a consistent G-scenario $\mathcal{S} = (V, C')$ for which we want to compute the best neighbors, and a set $\mathcal{T}$ of G-scenarios that contains the partial scenarios to be excluded. The aim of bestNeighbors is to compute the set of partial scenarios

Function bestNeighbors($\mathcal{N}$,$\mathcal{S}$,G,$\mathcal{T}$)

in : A QCN $\mathcal{N} = (V, C)$, a triangulation $G = (V, E)$ of $\mathsf{G}(\mathcal{N})$, a consistent G-scenario $\mathcal{S} = (V, C')$, and a set of G-scenarios $\mathcal{T}$.
output : The set of best neighbors of $\mathcal{S}$ *w.r.t.* $\mathcal{N}$, G and $\mathcal{T}$.

1 **begin**
2   bestNb $\leftarrow \emptyset$; best$\alpha \leftarrow +\infty$;
3   **foreach** $v \in V$ **do** bestNeighborsAux($\mathcal{N}$,$\mathcal{S}$,G,$\mathcal{T}$,$\mathcal{S}^{\uparrow v}$) ;
4   **return** bestNb;

Procedure bestNeighborsAux($\mathcal{N}$,$\mathcal{S}$, G,$\mathcal{T}$,$\mathcal{N}'$)

in : Two QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C'')$, a triangulation $G = (V, E)$ of $\mathsf{G}(\mathcal{N})$, a consistent G-scenario $\mathcal{S} = (V, C')$, and a set $\mathcal{T}$ of G-scenarios of $\mathcal{N}$.

1 **begin**
2   $\mathcal{N}' \leftarrow \overset{\diamond}{_G}(\mathcal{N}')$; $\alpha \leftarrow \alpha(\mathcal{N},\mathcal{N}')$;
3   **if** $\mathcal{N}'$ *is trivially inconsistant* **or** $\alpha >$ bestNb **then** **return**; ;
4   Select $(v, v') \in E$ such that $\mathcal{N}'[v, v']$ is not a singleton relation;
5   **if** *such a pair* (v, v') *exists* **then**
6   Select a base relation $b \in \mathcal{N}'[v, v']$;
7   bestNeighborsAux($\mathcal{N}$,$\mathcal{S}$,G,$\mathcal{T}$,$\mathcal{N}'_{[v,v']/\{b\}}$);
8   bestNeighborsAux($\mathcal{N}$,$\mathcal{S}$,G,$\mathcal{T}$,$\mathcal{N}'_{[v,v']/(\mathcal{N}'[v,v']\setminus\{b\})}$);
9   **return**;
10   **if** $\mathcal{N}' \notin \mathcal{T}$ **then**
11   **if** $\alpha <$ bestNb **then**
12   bestNb $\leftarrow \{\mathcal{N}'\}$; best$\alpha \leftarrow \alpha$;
13   bestNb $\leftarrow$ bestNb $\cup \{\mathcal{N}'\}$;

BestNb$(\mathcal{N}, \mathcal{S}, G, \mathcal{T})$. In a first step, bestNeighbors initializes the two global variables bestNb and bestα. bestNb, an initially empty set, will contain the computed best neighbors. bestα corresponds to the number of constraints of $\mathcal{N}$ that are unsatisfied by a partial scenario of set bestNb; this variable is initially assigned a value greater than the number of constraints of $\mathcal{N}$ (symbolized by $+\infty$). In a second step (line 3), in an iterative manner, the relaxation of partial scenario $\mathcal{S}$ *w.r.t.* each variable $v \in V$ is addressed in order to characterize the consistent partial scenarios that are candidates to being best neighbors. This is done though a call to function bestNeighborsAux.

Function bestNeighborsAux receives five parameters. The first four parameters $\mathcal{N}$, $\mathcal{S}$, G, and $\mathcal{T}$ are similar to the parameters of bestNeighbors. The fifth parameter $\mathcal{N}'$ is a QCN defined on the set of variables of $\mathcal{N}$. Function bestNeighbors computes the consistent G-scenarios of $\mathcal{N}'$ that are candidates to being best neighbors of $\mathcal{S}$ by taking into account the candidate best neighbors of bestNb previously computed. In a first step, bestNeighborsAux prunes some non-feasible base relations of $\mathcal{N}'$ by enforcing $\overset{\diamond}{_G}$-consistency on $\mathcal{N}'$. The integer α of non-overlapping constraints between $\mathcal{N}$ and $\mathcal{N}'$ is computed. When $\mathcal{N}'$ is detected as inconsistent or when the number of non-overlapping constraints between $\mathcal{N}$ and $\mathcal{N}'$ is greater than the number of unsatisfied constraints *w.r.t.* the neighbors of bestNb, we can assert that there exists no consistent G-scenario of $\mathcal{N}'$ that is better than those previously computed. Consequently, bestNeighborsAux terminates (line 3). In the contrary case, the treatment continues by selecting an edge (v, v') of G such that the corresponding constraint of $\mathcal{N}'$ is defined by a non-singleton relation. When such an edge exists, a base relation b

is extracted from relation $\mathcal{N}'[v, v']$ and the treatment continues through two recursive calls to bestNeighborsAux. The first call (line 7) allows exploring the QCN resulting from $\mathcal{N}'$ by replacing the constraint between v and v' with relation $\{b\}$. The second call (line 8) concerns the QCN resulting from $\mathcal{N}'$ by removing b from the constraint between v and v'. When all the constraints of $\mathcal{N}'$ corresponding to edges of G are defined by a singleton relation, we can assert that $\mathcal{N}'$ is a G-scenario. As G is a triangulation of $\mathsf{G}(\mathcal{N})$ and $\mathcal{N}'$ is $\overset{\diamond}{G}$-consistent, we have that $\mathcal{N}'$ is a consistent G-scenario. Also, we know that the number of non-overlapping constraints between $\mathcal{N}$ and $\mathcal{N}'$ is smaller or equal to bestα, *i.e.* the number of non-overlapping constraints between $\mathcal{N}$ and any of the partial scenarios of bestNb. Consequently, if $\mathcal{N}'$ does not belong to the set of excluded partial scenarios $\mathcal{T}$, $\mathcal{N}'$ must be considered as a best neighbor of $\mathcal{S}$ and added to the set bestNb (lines 10–13).

Theorem 1. *Let $\mathcal{N} = (V, C)$ and $\mathcal{S} = (V, C')$ be two* QCN*s of a qualitative calculus for which the partial $\diamond$-consistency is complete for $\widehat{\mathsf{B}}$, $G = (V, E)$ a triangulation of $\mathsf{G}(\mathcal{N})$, $\mathcal{T}$ a set of G-scenarios, and $\mathcal{S}$ a consistent G-scenario. Function* bestNeighbors *with parameters $\mathcal{N}$, $\mathcal{S}$, G, and $\mathcal{T}$ correctly computes the set* $\mathsf{BestNb}(\mathcal{N}, \mathcal{S}, G, \mathcal{T})$.

4 A Local Search Based Algorithm for the MAX-QCN problem

In this section, we present a local search based algorithm, called QLS (Qualitative Local Search), for solving the MAX-QCN problem. The QLS method takes as parameters a QCN $\mathcal{N} = (V, C)$ defined on a set of base relations B for which partial $\diamond$-consistency is complete for partial scenarios and a triangulation $G = (V, E)$ of $\mathsf{G}(\mathcal{N})$. In short, to find a consistent G-scenario that minimizes the number of unsatisfied constraints of $\mathcal{N}$, the QLS method moves from consistent G-scenario $\mathcal{S}$ to consistent G-scenario $\mathcal{S}'$ according to the notion of best neighbors presented in the previous section. The procedure QLS also includes a heuristic based on a weighting of the constraints to select at each iteration one of the best neighbors of the current G-scenario. A tabu list to exclude the G-scenarios already visited and a restart policy have been included. Now, we present in more detail the different steps realized in an iterative manner by the QLS method.

Initialization Step. An initial consistent G-scenario $\mathcal{S}$ on V to serve as the current partial scenario is randomly generated (see function randomScenario). The best partial scenario found, denoted by $\mathcal{S}^*$, is initialized with $\mathcal{S}$. Moreover, an integer weight denoted by $w(v, v')$ is associated with each edge $(v, v') \in E$. For each $(v, v') \in E$, $w(v, v')$ is initialized with 1 if $\mathcal{S}[v, v'] \subseteq \mathcal{N}[v, v']$, and 0 otherwise. In the case where the current partial G-scenario satisfies the constraint of $\mathcal{N}$ between v and v', the weight $w(v, v')$ will be incremented. Hence, $w(v, v')$ will represent the number of iterations for which the current partial G-scenario satisfies the constraint of $\mathcal{N}$ between v and v'. On the other hand, a tabu list $\mathcal{T}$ is introduced and initialized with $\{\mathcal{S}\}$. This tabu list will be used to avoid selecting new partial scenarios already selected.

Neighbor Generation and Selection Step. By using the function bestNeighbors presented earlier, the set of best neighbors $\mathsf{BestNb}(\mathcal{N}, \mathcal{S}, G, \mathcal{T})$ of $\mathcal{S}$ *w.r.t.* $\mathcal{N}$, G, and $\mathcal{T}$ is generated. Let us denote by BN this set. In the case where BN is an empty set, the procedure returns to the initialization step. In the contrary case, one partial scenario $\mathcal{S}'$ from the set BN is selected using a heuristic that uses assigned weights to each edge of E. More particularly, a weight $w(\mathcal{S}'')$ defined by $w(\mathcal{S}'') = 0 + \sum\{w(v, v') : (v, v') \in E \text{ and } \mathcal{S}''[v, v'] \subseteq \mathcal{N}[v, v']\}$ is associated with each partial scenario $\mathcal{S}''$ of BN. Intuitively, given a $\mathcal{S}'' \in$ BN, the greater the weight $w(\mathcal{S}'')$, the more the partial scenario $\mathcal{S}''$ corresponds to constraints used by the partial scenarios selected in the previous iterations. Consequently, to satisfy constraints of $\mathcal{N}$ which have been the least satisfied by the previous selected partial scenarios, the partial scenario $\mathcal{S}'$ is randomly selected among the elements of the set $\{\mathcal{S}'' \in \mathsf{BN} : w(\mathcal{S}'') = min\{w(\mathcal{S}''') : \mathcal{S}''' \in \mathsf{BN}\}\}$.

Acceptance and Restarts Step. The G-scenario $\mathcal{S}'$ replaces the current partial scenario $\mathcal{S}$ and the best partial scenario found $\mathcal{S}^*$ is possibly updated. Moreover, $\mathcal{S}'$ is added to the tabu list $\mathcal{T}$. For each $(v, v') \in E$ such that $\mathcal{S}^*[v, v'] \subseteq \mathcal{N}[v, v']$, $w(v, v')$ is incremented. On the other hand, when for the nbDivLoops last iterations (with nbDivLoops a positive integer parameter of QLS), the number of unsatisfied constraints of $\mathcal{N}$ by the selected partial scenario forms an increasing series of diversification is realised by substituing the current partial scenario $\mathcal{S}$ by a partial scenario randomly selected in the tabu list $\mathcal{T}$. Concerning restarts, after a number of nbRestartDiv diversifications (with nbRestartDiv a positive integer parameter of QLS) a restart stage is realized by returning to the initialization step.

Termination Step. In this step, two additional integer parameters of QLS called maxLoops and expectedValue (with 0 as default value) are used. Whether the number of iterations is greater or equals to maxLoops or whether the $\alpha(\mathcal{N}, \mathcal{S}^*)$ is less or equals to expectedValue, the treatment terminates after returning the best partial scenario $\mathcal{S}^*$.

In the next section we will report some experimental results regarding the aforementioned method QLS and three variations of it denoted by $\mathsf{QLS}^{\not r}$, $\mathsf{QLS}^{\not w}$ and $\mathsf{QLS}^{\mathcal{T}}$. The method $\mathsf{QLS}^{\not r}$ is the method QLS without using restarts and the method $\mathsf{QLS}^{\not w}$ corresponds to the method QLS without using the weights. Hence, for $\mathsf{QLS}^{\not w}$, the selected neighbor is one of the elements of the whole set of best neighbors of the current partial scenario. $\mathsf{QLS}^{\mathcal{T}}$ corresponds to QLS without using a triangulation G of the considered QCN $\mathcal{N}$. In an equivalent manner, $\mathsf{QLS}^{\mathcal{T}}$ is the method QLS for which the parameter G is the complete graph over V, where V is the set of variable of the input QCN $\mathcal{N}$.

Function randomScenario(G)

in : A graph $G = (V, E)$.
output : A consistent G-scenario on V randomly generated.

begin
  $\mathcal{N} \leftarrow \mathsf{B}_V$; `/*` B_V `is the` QCN `on` V `whose each constraint is` B `*/`
  foreach $(v, v') \in E$ **do**
    Select randomely a base relation $b \in \mathcal{N}[v, v']$ such that $\diamond_G(\mathcal{N}_{[v,v']/\{b\}})$ is not trivially inconsistent;
    $\mathcal{N} \leftarrow \diamond_G(\mathcal{N}_{[v,v']/\{b\}})$
  return $\mathcal{N}$;

5 Experiments

In this section, we report a preliminary evaluation of the method QLS presented in the previous section. We considered the QCNs from IA used in the experiments reported in [6]. These 280 QCNs were randomly generated using the model $A(n, d, s)$ (proposed in [12]), with n being the number of variables of the generated QCNs, d the density of constraints defined by a relation other than the trivial relation (*i.e.* B), and s the average number of base relations in each constraint. The parameters used to generate the considered QCNs are $n = 20$, d varying from 8 to 14.5 with a step of 0.25, and $s = 6.5$. The relatively small number of variables $n = 20$ was decided in order to present results as complete as possible. For each considered value of d, 10 instances were generated. Concerning triangulations of the constraint graphs of QCNs, we also use the same ones as those used in [6]. These triangulations were generated using a greedy triangulation algorithm (cf. the GreedyFillIn heuristic [3]). Moreover, the method QLS and its different variations have been implemented in Java. The main objective of our experiment is to validate the main different ingredients of the method QLS, *i.e.*, the use of (1) triangulations of the constraint graphs of the QCNs, (2) the proposed heuristic based on weights on the constraints, and (3) restarts. In order to do this, we compare QLS with its different variations QLS$^{\not\mathcal{R}}$, QLS$^{\not\mathcal{W}}$, and QLS$^{\not\mathcal{T}}$. Note that for this comparison we used the value 5 for the parameter nbDivLoops and the same value for the parameter nbRestartDiv (the best values among the tested values). Moreover, we used the exact optimal number of unsatisfied constraints for the parameter expectedValue.

The first part of our analysis concerns the optimal number of unsatisfied constraints. Figure 3a presents the average of the optimal number of unsatisfied constraints found by the different methods for a maximal number of iterations equal to 4000 (*i.e.* $\mathsf{maxLoops} = 4000$) which corresponds approximatively to a 15-minutes timeout. The optimal values found are close enough to the exact optimal values. We note that QLS and QLS$^{\not\mathcal{T}}$ outperform QLS$^{\not\mathcal{R}}$ and QLS$^{\not\mathcal{W}}$. Further, QLS is slightly better than QLS$^{\not\mathcal{T}}$. This observation is reinforced when considering the number of exact optimal values found by each method. Indeed, for QLS, QLS$^{\not\mathcal{R}}$, QLS$^{\not\mathcal{W}}$, and QLS$^{\not\mathcal{T}}$, the number of QCNs out of the 280 QCNs used for which the exact value is found is respectively 189, 58, 93, and 160. Moreover, note that by considering all the methods, 214 exact values are found.

The second part of our analysis concerns a more precise comparison between the two methods QLS and QLS$^{\not\mathcal{T}}$ which were found to be close to one another in terms of performance in the previous analysis. Figure 3b reports the average time needed by these two methods for reaching their optimal values. The QCNs considered in this report are the 208 QCNs out of the 280 QCNs for which QLS and QLS$^{\not\mathcal{T}}$ obtain the same optimal values. Clearly, in general QLS takes largely less time than QLS$^{\not\mathcal{T}}$ to obtain the same optimal values. On the other hand, by examining the total number of (partial) scenarios computed during the neighbor generation step (not reported due to lack of space), we note that the respective numbers are close enough to one another regarding QLS and QLS$^{\not\mathcal{T}}$. From this, we can explain the better performance of QLS considering time, due to the fact that the computation of the neighbors during the generation step is faster than that of QLS$^{\not\mathcal{T}}$ as the use of triangulations in the former case allows considering less constraints in general.

The last part of our analysis concerns a comparison between our approach and the one proposed in [6] which encodes the MAX-QCN problem into the partial maximum satisfiability problem (PMAX-SAT) using a notion called *forbidden coverings*. Due to lack of space, we do not give details about this approach. Nevertheless, we mention that we used the encoding referenced by $\mathcal{C}^{5;8}_{\mathsf{FCTEX}}$ in [6] for our experiments and that we will use the term of *complete method* to refer to the corresponding solving method. We ran the QLS method and the complete method with a 9000-second timeout. The number of QCNs for which we obtain the exact optimal value is 261 for QLS and 279 for the complete method. Regarding the solving time, in general the complete method outperforms the QLS method for the QCNs with a density of non trivial constraints less than 12.5, which are easy to solve for the complete method. Focusing on the 80 instances with a density greater or equal to 13.0, for 27 instances the time needed by QLS to find the exact optimal value is less than that needed by the complete method.

6 Conclusions

To solve MAX-QCN, we propose an original local search based method called QLS which is generic in the sense that it can be used in the context of numerous qualitative calculi. Preliminary experimentation shows the interest of our approach, in particular the useful of the different ingredients components of QLS : the using of traingulations of the constraint

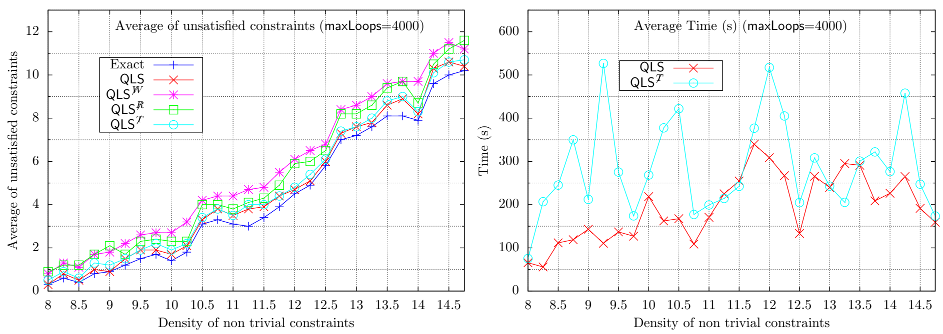

Fig. 3: (left) Optimal number of unsattisfied constraints; (right) average time for the instances of $A(20, d, 6.5)$ with $d \in \{8, \ldots, 14.75\}$.

graphs, the using of restarts and the using of an heuristics based on weights on the constraints. Future work consists of conducting experiments with other calculi than IA and with large QCNs. Concerning QLS, another perspective consists in the using of other neigbours that this one used currently. Particularly, we envisage to consider the neigbors computed from the moving of two or several variables, rather that from the moving of one variable. We also envisage to define hybrid method mixing the proposed local search method and complete method as the branch and bound algorithm proposed in [5].

References

1. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Comm. ACM 26, 832–843 (1983)
2. Bhatt, M., Guesgen, H., Wölfl, S., Hazarika, S.: Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions. Spatial Cognition & Computation 11, 1–14 (2011)
3. Bodlaender, H.L., Koster, A.M.C.A.: Treewidth computations I. Upper bounds. Inf. Comput. 208, 259–275 (2010)
4. Chmeiss, A., Condotta, J.F.: Consistency of Triangulated Temporal Qualitative Constraint Networks. In: ICTAI (2011)
5. Condotta, J.F., Mensi, A., Nouaouri, I., Sioutis, M., Saïd., L.B.: A Practical Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In: ICTAI (2015)
6. Condotta, J., Nouaouri, I., Sioutis, M.: A SAT Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In: KR (2016)
7. Diestel, R.: Graph Theory, 4th Edition, Graduate texts in mathematics, vol. 173. Springer (2012)
8. Hazarika, S.: Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions. Igi Global (2012)
9. Huang, J.: Compactness and Its Implications for Qualitative Spatial and Temporal Reasoning. In: KR (2012)
10. Ligozat, G., Renz, J.: What is a qualitative calculus? A general framework. In: PRICAI (2004)
11. Martí, R., Laguna, M., Glover, F.: Principles of tabu search. In: Gonzalez, T.F. (ed.) Handbook of Approximation Algorithms and Metaheuristics. Chapman and Hall (2007)
12. Nebel, B.: Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class. Constraints 1, 175–190 (1997)
13. Randell, D., Cui, Z., Cohn, A.: A Spatial Logic Based on Regions and Connection. In: KR (1992)
14. Renz, J., Nebel, B.: Qualitative spatial reasoning using constraint calculi. In: Handbook of Spatial Logics, pp. 161–215 (2007)
15. Sioutis, M., Koubarakis, M.: Consistency of Chordal RCC-8 Networks. In: ICTAI (2012)