A Hybrid Evolutionary Algorithm for Maximizing Satisfiability in Temporal or Spatial Qualitative Constraints

Ali Mensi
CRIL-CNRS UMR 8188
Université Lille-Nord de France
mensi@cril.fr

Jean-François Condotta
CRIL-CNRS UMR 8188
Université Lille-Nord de France
condotta@cril.fr

Issam Nouaouri
LGI2A
Université Lille-Nord de France
issam.nouaouri@univ-artois.fr

Michael Sioutis
AASS
Örebro universitet, Sweden
michael.sioutis@oru.se

Lamjed Ben Saïd
SMART-Lab
Institut Supérieure de Gestion,Tunisia
lamjed.bensaid@isg.rnu.tn

ABSTRACT

In this paper we tackle the MAX-QCN optimization problem, which consists in characterizing a consistent scenario that maximizes the satisfiability of a spatial or temporal qualitative constraint network (QCN). We propose an original hybrid evolutionary algorithm for solving the MAX-QCN problem, which we call EAMQ for short. This EAMQ method consists in randomly generating an initial population of consistent $\mathcal{G}$-scenarios, and then realizing in an iterative manner an evolution of this population by generating new $\mathcal{G}$-scenarios from crossover operations applied on the better individuals of the population at hand. Additionally, every time a new scenario is generated, an exploration of its neighborhood is realized in order to obtain a better scenario. Preliminary experiments conducted on QCNs of the Interval Algebra show the interest of our approach for solving the MAX-QCN problem.

KEYWORDS

Spatio-temporal reasoning, qualitative constraints, optimization

ACM Reference Format:

Ali Mensi, Jean-François Condotta, Issam Nouaouri, Michael Sioutis, and Lamjed Ben Saïd. 2018. A Hybrid Evolutionary Algorithm for Maximizing Satisfiability in Temporal or Spatial Qualitative Constraints. In *SETN '18: 10th Hellenic Conference on Artificial Intelligence, July 9–15, 2018, Rio Patras, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3200947.3201021

1 INTRODUCTION

Qualitative Spatial and Temporal Reasoning (QSTR) is a well-established area in the field of Knowledge Representation and Reasoning that has been continuously growing over the past decades due to the diversity of the proposed qualitative calculi for representing entities and events in time or space [19]. The usefulness of the qualitative approach is demonstrated in a vast range of applications that involve dynamic GIS, cognitive robotics, medicine, and spatio-temporal design [3, 17, 26].

The problem of representing and reasoning about qualitative information can be modeled as a qualitative constraint network (QCN) using a qualitative constraint language. Specifically, a QCN is a network of constraints corresponding to qualitative spatial or temporal relations between spatial or temporal variables respectively, and a qualitative constraint language is used to define those constraints over a finite set of binary relations, called *base relations* (or *atoms*) [19]. In the literature, with respect to time, the Interval Algebra (IA) [2] models the knowledge about the temporal relations between intervals in the timeline by using qualitative constraints such as *precedes*, *overlaps*, and *during*. On the other hand, with respect to space, a subset of the Region Connection Calculus algebra (RCC) [25], namely RCC8, encodes knowledge about the spatial relations between topological regions by using qualitative constraints such as *disconnected*, *overlaps*, and *equals*. These are the most well-known and used calculi in QSTR for representing and reasoning with qualitative temporal and spatial information respectively.

Motivation

Given a QCN, we are mainly interested in its satisfiability checking problem, which is the problem of obtaining a valuation of its set of variables such that all of its constraints are satisfied, that valuation being called a solution. However, obtaining a solution is clearly not possible when the QCN at hand is unsatisfiable. In such cases, we need to *relax* the initial network and try to find a partial valuation that satisfies the maximum number of constraints in the considered QCN. This optimisation problem is called the MAX-QCN problem and it was recently introduced and studied in [9]. 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 $\mathcal{N}$. The motivation behind studying the MAX-QCN problem lies in the fact that representing spatial or temporal information may inevitably lead to inconsistencies. With respect to temporal information for instance, timetabling is an example of a scheduling problem where inconsistencies can naturally occur due to the lack of resources for certain tasks [24]. In particular, timetabling deals with identifying the suitable temporal intervals for a number of tasks that require certain limited in amount resources. In the context of a university, an inconsistency can appear

SETN '18, July 9–15, 2018, Rio Patras, Greece

ACM ISBN 978-1-4503-6433-1/18/07…$15.00
https://doi.org/10.1145/3200947.3201021

when two professors choose to teach the same class of students at overlapping temporal intervals. The incosistency must then be repaired by taking into account the available temporal intervals and the preferences of the professors, and minimizing changes in the timetable so as to distort its structure as little as possible. Thus, solving the MAX-QCN problem is clearly at least as difficult as solving the satisfiability checking problem.

Related Work

In [9], Condotta *et al.* propose a complete and generic branch and bound algorithm based on techniques used in the literature for solving the satisfiability checking problem and the minimal labeling problem of a given QCN. In particular, these techniques involve the use of a tractable subclass of relations, a triangulation of the constraint graph of the input QCN, and the partial closure under weak composition as a filtering method. In [8] the authors view this problem as a partial maximum satisfiability problem [18] and introduce two related families of encodings based on a forbidden covering with regard to the composition table of the considered qualitative calculus. Intuitively, a forbidden covering expresses the non-feasible configurations for three spatial or temporal entities. Further, in [7] the authors present a particular tabu local search method that involves first obtaining a partial scenario S of the given QCN and then exploring neighboring refinements obtained by disconnecting a variable of S and repositioning it appropriately. This approach combines the use of heuristics along with the management of a tabu list of *no-good* scenarios in order to find the best neighboring scenario at each execution step.

Contributions

In this paper we discuss the applicability of Evolutionary Algorithms [5, 10–13, 21, 22, 27] (EA) enhanced by heuristics and adaptive fitness computation for solving the MAX-QCN problem. A typical EA is composed of three essential elements [10, 15, 16]: (*i*) a population consisting of several individuals representing potential solutions of the given problem; (*ii*) a mechanism for assessing the adaptation of each individual of the population to his or her external environment; (*iii*) a mechanism of evolution composed of operators allowing to eliminate certain individuals and to produce new individuals from the selected individuals. Our algorithm consists in randomly generating an initial population of consistent scenarios, and then realizing in an iterative manner an evolution of this population by generating new scenarios from crossover operations applied on the better individuals of the population at hand. Additionally, every time a new scenario is generated, an exploration of its neighborhood is realized in order to obtain a better scenario. Moreover, the EAMQ method integrates a diversification step to avoid convergence towards a local minimum.

The paper is organized as follows. Section 2 is devoted to some preliminaries about QSTR and the MAX-QCN problem. In Section 3, we introduce a hybrid evolutionary algorithm for the MAX-QCN problem, called EAMQ. In Section 4, we describe how to obtain a better scenario from a generated scenario by exploring its neighborhood. In Section 5, we present some original crossover operators for QCNs that allow us to generate a consistent scenario from two consistent scenarios. In Section 6, we report some experimental results about this method. Finally, we conclude the paper and give some perspectives for future work in Section 7.

Relation	Symbol	Inverse	Meaning
precedes	p	pi	
meets	m	mi	
overlaps	o	oi	
starts	s	si	
during	d	di	
finishes	f	fi	
equals	eq	eq	

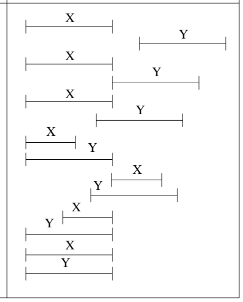

Figure 1: The base relations of IA.

2 PRELIMINARIES

A binary spatial or temporal qualitative calculus considers a domain D of spatial or temporal entities respectively and a finite set B of *jointly exhaustive and pairwise disjoint* relations defined on that domain called base relations [19]. 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 ($^{-1}$). 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. The set 2^{B} is equipped with the usual set operations (union and intersection), the converse operation, and the weak composition operation. The converse of a relation is the union of the converses of its base relations. The weak composition $\diamond$ of $b, b' \in \mathsf{B}$ 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 $r, r' \in 2^{\mathsf{B}}$, $r \diamond r'$ 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} that contains the singleton relations of 2^{B} and the universal relation and that is closed under the operations $^{-1}$, $\diamond$, and $\cap$. Consider the well known temporal qualitative calculus introduced by Allen [1], called the Interval Algebra (IA). Allen represents temporal entities as intervals in the timeline and considers the set of 13 base relations $\mathsf{B}_{\mathsf{IA}} = \{\mathsf{eq}, \mathsf{p}, \mathsf{pi}, \mathsf{m}, \mathsf{mi}, \mathsf{o}, \mathsf{oi}, \mathsf{s}, \mathsf{si}, \mathsf{d}, \mathsf{di}, \mathsf{f}, \mathsf{fi}\}$ (Figure 1).

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, C is such that $C(v, v) \subseteq \{\mathsf{Id}\}$ and $C(v, v') = (C(v', v))^{-1}$. 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 with 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 to

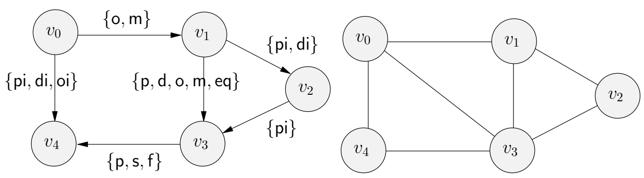

Figure 2: A QCN $\mathcal{N} = (V, C)$ of IA and a graph G that is a triangulation of $\mathrm{G}(\mathcal{N})$.

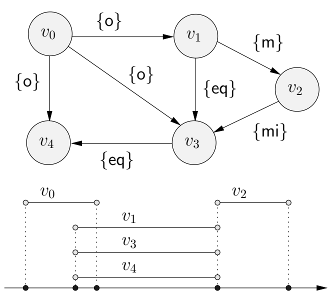

Figure 3: A consistent G-scenario $\mathcal{S}$ (with G in Fig. 2) and a solution of $\mathcal{S}$.

be 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 such that each constraint is defined by a singleton relation. A scenario $\mathcal{S}$ of $\mathcal{N}$ 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 $v'' = v$, and $C'(v', v'') = C(v', v'')$ otherwise. Given a subset of variables $V' \subseteq V$, the projection of $\mathcal{N}$ on V', denoted by $\mathcal{N}_{\downarrow V'}$, is the QCN $\mathcal{N}$ restricted on V'. In the sequel, $\mathcal{N}_{[v,v']/r}$ with $r \in 2^{\mathsf{B}}$, denotes the modified QCN $\mathcal{N}$ for which the relation defining the constraint between v and v' has been substituted by r. Given two QCNs $\mathcal{N}^1 = (V^1, C^1)$ and $\mathcal{N}^2 = (V^2, C^2)$ such that $V^1 \cap V^2 = \emptyset$, $\mathcal{N}^1 \cup \mathcal{N}^2$ will denote in the sequel the QCN (V, C) defined by: $V = V^1 \cup V^2$, for all $v, v' \in V^1$, $C(v, v') = C^1(v, v')$, for all $v, v' \in V^2$, $C(v, v') = C^2(v, v')$, and for all $v, v' \in V$ such that $v \in V^1$ and $v' \in V^2$, or $v \in V^2$ and $v' \in V^1$, $C(v, v') = \mathsf{B}$.

Given two graphs $G = (V, E)$ and $G' = (V', E')$, G is a subgraph of G', denoted by $G \subseteq G'$, iff $V \subseteq V'$ and $E \subseteq E'$. The graph G is a *chordal (or triangulated) graph* iff each of its cycles of length > 3 has a chord [14]. A *partial* scenario *w.r.t.* G, also called a 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 $(v, v') \in E$. The constraint graph of a QCN $\mathcal{N} = (V, C)$ is the graph (V, E), denoted by $\mathrm{G}(\mathcal{N})$, for which $(v, v') \in E$ 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.* G or $\overset{\diamond}{G}$-consistent [6] iff for all (v, v'), (v, v''), $(v'', v') \in E$, we have that $C(v, v') \subseteq C(v, v'') \diamond C(v'', v')$. The closure of $\mathcal{N}$ under $\overset{\diamond}{G}$-consistency, 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 [6, 28], where δ is the maximum degree of G. Note that $\overset{\diamond}{G}(\mathcal{N})$ is equivalent to $\mathcal{N}$ (*i.e.* these QCNs admit the same set of solutions). We will say that $\mathcal{N}$ is minimal *w.r.t.* G iff for each (v, v') and each $b \in \mathcal{N}[v, v']$ there exists a consistent scenario $\mathcal{S}$ of $\mathcal{N}$ such that $\mathcal{S}[v, v'] = \{b\}$. In the sequel, we will consider the following property:

Definition 2.1. Given a set of base relations B, we will say that partial $\diamond$-consistency implies minimality for $\widehat{\mathsf{B}}$ iff for any triangulated graph $G = (V, E)$ and any $\overset{\diamond}{G}$-consistent QCN $\mathcal{N}$ such that $\mathcal{N}[v, v'] = \mathsf{B}$ for any $(v, v') \notin E$ and $\mathcal{N}[v, v'] \in \widehat{\mathsf{B}}$ for any $(v, v') \in E$ we have that $\mathcal{N}$ is minimal *w.r.t.* G.

Partial $\diamond$-consistency implies minimality for $\widehat{\mathsf{B}}$ for many qualitative calculi [20], such as 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 define MAX-QCN more formally, we introduce the operator α, which takes as parameters two QCNs and returns the number of non-overlapping constraints of these QCNs. Formally, given two QCNs $\mathcal{N} = (V, C)$ and $\mathcal{N}' = (V, C')$, $\alpha(\mathcal{N}, \mathcal{N}')$ is defined by $\alpha(\mathcal{N}, \mathcal{N}') = \frac{1}{2}.|\{(v, v') \in V \times V : v \neq v' \text{ and } C(v, v') \cap C'(v, v') = \emptyset\}|$. Given $\mathcal{N} = (V, C)$, a solution of the MAX-QCN for $\mathcal{N}$ is a consistent scenario $\mathcal{S}$ on V, called an optimal scenario of $\mathcal{N}$, such that there is no consistent scenario $\mathcal{S}'$ on V with $\alpha(\mathcal{S}, \mathcal{N}) > \alpha(\mathcal{S}', \mathcal{N})$. Given a QCN $\mathcal{N} = (V, C)$, an optimal G-scenario of $\mathcal{N}$ is a consistent G-scenario $\mathcal{S}$ 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 $\mathrm{G}(\mathcal{N}) \subseteq G$. Note that every consistent scenario of an optimal G-scenario of a QCN $\mathcal{N}$ is an optimal scenario of $\mathcal{N}$. The EAMQ method that we will define in the sequel is adapted for QCNs of a qualitative calculus $\mathcal{Q}$ for which partial $\diamond$-consistency implies minimality for $\widehat{\mathsf{B}}$. Due to this, the method can consider partial scenarios rather than complete scenarios, much like the methods proposed in [8, 9]. The usefulness of considering partial scenarios is due to the faster treatment that results from discarding some constraints.

3 AN EVOLUTIONARY ALGORITHM FOR MAX-QCN

In this section, we present a hybrid evolutionary algorithm for solving the MAX-QCN problem, denoted by EAMQ (Evolutionary Algorithm for MAX-QCN). EAMQ takes as main parameters a QCN $\mathcal{N} = (V, C)$ and a triangulation $G = (V, E)$ of $\mathrm{G}(\mathcal{N})$. The QCN $\mathcal{N}$ is assumed to be defined on a set of base relations B for which partial $\diamond$-consistency implies minimality for $\widehat{\mathsf{B}}$.

Succinctly, to characterize a consistent G-scenario that maximizes the number of satisfied constraints of $\mathcal{N}$, EAMQ randomly generates an initial population of consistent G-scenarios, and then realizes in an iterative manner an evolution of this population by generating new G-scenarios from crossover operations applied on the best individuals of the current population. Additionally, at each generation of a new scenario, an exploration of its neighborhood is realized in order to find a better scenario. Moreover, EAMQ integrates a diversification step to avoid convergence towards a local minimum. Now, we detail the different steps of EAMQ.

Initialization Step. In this step (see lines 2-4 of the function EAMQ), the multiset $\mathsf{S_P}$ is initialized by an initial population of $card_P$ consistent G-scenarios generated partly at random, with $card_P$, a strictly positive integer, given as parameter to EAMQ. The

Function EAMQ($\mathcal{N}$,G,$card_P$,$card_{Best}$,$divT$,$timeoutL$)

```
   in      : A QCN N = (V, C), a triangulation G = (V, E) of G(N), and four
             positive integers card_P, card_Best, divT, timeoutL.
   output  : A consistent G-scenario on V.
1  begin
     // Initialization Step
2    S_P ← ∅; nbLoops ← 0;
3    for i ← 1 to card_P do
4      S ← randomScenario(G); S ← exploreNeighborhood(N, S, G);
       S_P ← S_P ∪ {S}
     // Main loop
5    while the timeout limit timeoutL is not reached do
6      nbLoops ← nbLoops + 1;
       // Selection Step
7      S_Best ← selectBestScenarios(S_P, card_Best);
       // New Generation Step
8      if (nbLoops modulo divT) ≠ 0 then
         // Crossovers Step
9        S_G ← ∅;
10       for i ← 1 to card_P − card_Best do
11         Select randomly S_1 and S_2 from S_Best;
12         S ← crossover(N, G, S_1, S_2);
             S ← exploreNeighborhood(N, S, G);
13         S_G ← S_G ∪ {S}
14       S_P ← S_Best ∪ S_G;
15     else
         // Diversification Step
16       S_P ← {selectBestScenarios(S_Best, 1)};
17       for i ← 1 to card_P − 1 do
18         S ← randomScenario(G);
             S ← exploreNeighborhood(N, S, G);
19         S_P ← S_P ∪ {S}

20   return selectBestScenarios(S_Best, 1);
```

generation of each new partial scenario $\mathcal{S}$ is made in two phases. In a first phase, $\mathcal{S}$ is initialized by a randomly generated consistent G-scenario using the function randomScenario. In a second phase (line 4), $\mathcal{S}$ is possibly replaced by a better neighbor of $\mathcal{S}$, *i.e.* a scenario belonging to the neighborhood of $\mathcal{S}$ and satisfying strictly more constraints of $\mathcal{N}$ than $\mathcal{S}$. This operation is realized by the function exploreNeighborhood, which will be detailed in the next section.

Selection Step. In the first part of the main loop of EAMQ a selection of $card_{Best}$ best scenarios belonging to the multiset S_P is realized, where $card_{Best}$ is a strictly positive integer given as parameter to EAMQ such that $card_{Best} < card_P$. This treatment is realized by the function selectBestScenarios. Moreover, the selected scenarios will be placed in the multiset S_{Best}. After this step, for each scenario $\mathcal{S} \in S_{Best}$ and each scenario $\mathcal{S}' \in S_P \setminus S_{Best}$, we have $\alpha(\mathcal{N}, \mathcal{S}) \leq \alpha(\mathcal{N}, \mathcal{S}')$.

Crossovers Step. In this step, the previously selected scenarios at the selection step are used to generate new scenarios in order to form a part of the new population corresponding to the multiset S_G. Each new scenario $\mathcal{S}$ is first generated with a crossover operator, called, in a generic way, crossover, which is applied on two randomly selected scenarios of the multiset S_{Best}. In the sequel of the paper, some of the crossover operators will be presented in detail. Note that similarly to the generation of scenarios in the initialization step, this generation is completed by the search of a better scenario than $\mathcal{S}$ in its neighborhood. The population of the next iteration of the main loop will be formed by the scenarios of S_{Best} and the new scenarios saved into S_G, unless a diversification treatment is triggered.

Function randomScenario(G)

```
  in     : A chordal graph G = (V, E).
  output : A consistent G-scenario on V randomly generated.
1 begin
2   S ← ⊤_V;            /* ⊤_V is the QCN on V whose each constraint is B */
3   while there exists (v, v') ∈ E such that |S[v, v']| > 1 do
4     Select randomly (v, v') ∈ E such that |S[v, v']| > 1; Select randomly a base
        relation b ∈ S[v, v'];
5     S ← S_[v,v']/{b}; S ← ◇_G(S);
6   return S;
```

Diversification Step. This step is triggered every $divT$ number of loops, where $divT$ is a strictly positive integer given as parameter to EAMQ. The aim of this step is to avoid a convergence towards a local minimum during the search, more particularly to avoid a population of scenarios that do not allow obtaining better scenarios in the sequel of the search. The treatment of this step consists in keeping in the next population one of the best scenarios from the previously selected scenarios, *i.e.* from the scenarios S_{Best}. The aim of this is to obtain a future population that will not be better (in term of optimal value) than the current population. In addition to this scenario, the $card_P - 1$ other scenarios of the next population are randomly generated in a similar manner to the one used in the initialization step.

To end this section, we examine the function randomScenario. This function takes as parameter a chordal graph $G = (V, E)$ and builds in an iterative manner a G-scenario formed by base relations of the set B. At the beginning of each iteration we can notice that the QCN $\mathcal{S}$ is partially $\diamond$-consistent *w.r.t.* the chordal graph G and not trivially inconsistent. Hence, from the assumption made on the set of base relations B, $\mathcal{S}$ is minimal *w.r.t.* G. Consequently, for each base relation $b \in \mathcal{S}[v, v']$ and for all $(v, v') \in E$ there exists a consistent scenario of $\mathcal{S}$ such that the constraint between v and v' is defined by $\{b\}$. From this, we can assert that the QCN $\mathcal{S}$ at the end of each iteration is necessarily not trivially inconsistent and $\overset{\diamond}{_G}$-consistent. Moreover, we can notice that the QCN $\mathcal{S}$ returned is a QCN whose constraints are defined by singleton relations. Thus, we can assert that the QCN $\mathcal{S}$ returned by the function randomScenario is a G-scenario that is not trivially inconsistent and $\overset{\diamond}{_G}$-consistent, where G is a chordal graph. Again, from the assumption made on B we can conclude that the QCN $\mathcal{S}$ returned by the function randomScenario is a consistent G-scenario. We can also notice that the function randomScenario realizes at most $|E|$ iterations. We have the following result:

PROPOSITION 3.1. *Let $G = (V, E)$ be a chordal graph. By assuming that the set of base relations* B *is such that partial $\diamond$-consistency implies minimality for* $\widehat{\mathsf{B}}$, *we have that the function* randomScenario *returns a consistent G-scenario on V in polynomial time.*

4 NEIGHBORHOOD EXPLORATION

Regarding the proposed method, a new scenario in the considered population can be randomly generated from scratch or obtained from a crossover between two selected scenarios. In these two cases, an additional treatment on the initially generated scenario is realized in order to obtain a better scenario. This treatment, which one could consider as a systematic mutation step, consists in exploring

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

in : A QCN $\mathcal{N} = (V, C)$, a triangulation $G = (V, E)$ of $\mathrm{G}(\mathcal{N})$, a consistent G-scenario $\mathcal{S} = (V, C')$.
output : A best neighbor of $\mathcal{S}$ *w.r.t.* $\mathcal{N}$ and G better than $\mathcal{S}$, or $\mathcal{S}$ itself if such a scenario does not exist.

1 **begin**
2 bestNg $\leftarrow \mathcal{S}$; best$\alpha \leftarrow \alpha(\mathcal{N}, \mathcal{S})$;
3 **foreach** $v \in V$ **do** exploreNeighborhoodAux($\mathcal{N}$,G,$\mathcal{S}^{\uparrow v}$) ;
4 **return** bestNg;

Procedure exploreNeighborhoodAux($\mathcal{N}$,G,$\mathcal{N}'$)

in : Two QCNs $\mathcal{N} = (V, C)$, $\mathcal{N}' = (V, C')$ and a triangulation $G = (V, E)$ of $\mathrm{G}(\mathcal{N})$ and $\mathrm{G}(\mathcal{N}')$.

1 **begin**
2 $\mathcal{N}' \leftarrow \overset{\diamond}{_G}(\mathcal{N}')$; $\alpha \leftarrow \alpha(\mathcal{N}, \mathcal{N}')$;
3 **if** $\alpha \geq$ bestα **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 exploreNeighborhoodAux($\mathcal{N}$,G,$\mathcal{N}'_{[v,v']/\{b\}}$);
8 exploreNeighborhoodAux($\mathcal{N}$,G,$\mathcal{N}'_{[v,v']/(\mathcal{N}'[v,v']\setminus\{b\})}$);
9 **else**
10 bestNg $\leftarrow \mathcal{N}'$; best$\alpha \leftarrow \alpha$;

the neighborhood of the initially generated scenario to possibly characterize a better scenario satisfying strictly more constraints of the considered QCN than the former scenario. If such a better scenario exists, it will supersede the initially generated scenario. The notion of neighborhood of a consistent scenario that we use is the one proposed in [7] and corresponds to the set of the consistent scenarios that can be obtained by disconnecting a variable and repositioning it in the considered consistent scenario. More formally, given a consistent G-scenario $\mathcal{S}$ and a graph G, the set of the neigbours of $\mathcal{S}$ *w.r.t.* G, denoted by $\mathsf{Nb}(\mathcal{S}, G)$, is defined by: $\mathsf{Nb}(\mathcal{S}, G) = \bigcup_{v \in V} \{\mathcal{S}' : \mathcal{S}' \neq \mathcal{S}$ and $\mathcal{S}'$ is a consistent G-scenario of $\mathcal{S}^{\uparrow v}\}$. Given a G-scenario $\mathcal{S} = (V, C)$, a graph $G = (V, E)$, and a QCN $\mathcal{N}$, the set of best neighbors of $\mathcal{S}$ *w.r.t.* $\mathcal{N}$ is the subset of the partial scenarios of $\mathsf{Nb}(\mathcal{S}, G)$ that maximize the number of satisfied constraints of the considered QCN $\mathcal{N}$. By denoting this subset by $\mathsf{BestNb}(\mathcal{S}, G, \mathcal{N})$ we have $\mathsf{BestNb}(\mathcal{S}, G, \mathcal{N})$ which corresponds to the set $\{\mathcal{S} \in \mathsf{Nb}(\mathcal{S}, G)$ such that there is no $\mathcal{S}' \in \mathsf{Nb}(\mathcal{S}, G)$ with $\alpha(\mathcal{N}, \mathcal{S}) > \alpha(\mathcal{N}, \mathcal{S}')\}$. By adapting the algorithm bestNeighbors proposed in [7], which allows us to compute the set of partial scenarios $\mathsf{BestNb}(\mathcal{S}, G, \mathcal{N})$, we propose the function exploreNeighborhood, which allows us to characterize a scenario of the aforementioned subset that is better than $\mathcal{S}$ (if such a scenario exists).

The function exploreNeighborhood receives three parameters, namely, a QCN $\mathcal{N} = (V, C)$ for which we want to solve MAX-QCN, a triangulation $G = (V, E)$ of $\mathrm{G}(\mathcal{N})$, and a consistent G-scenario $\mathcal{S} = (V, C')$ for which we want to compute a better scenario through an exploration in its neighborhood. In a first step, exploreNeighborhood initializes the global variables bestNg by $\mathcal{S}$ and bestα by the number of constraints of $\mathcal{N}$ that are unsatisfied by $\mathcal{S}$. In the case where there is no better neighbor than $\mathcal{S}$, these variables will not change and $\mathcal{S}$ will correspond to the returned partial scenario. In a second step (line 3), the relaxation of the partial scenario $\mathcal{S}$ *w.r.t.* each variable $v \in V$ is considered in order to search a better neighbor with respect to this relaxation. This search is done through a call to the recursive function exploreNeighborhoodAux. This function has three parameters: $\mathcal{N}$, G, and $\mathcal{N}'$. The parameters $\mathcal{N}$ and G are similar to the parameters of exploreNeighborhood. The third parameter $\mathcal{N}'$ is a QCN with the same set of variables as $\mathcal{N}$ and such that G is also a triangulation of $\mathrm{G}(\mathcal{N}')$. The objective of the function exploreNeighborhoodAux is to find a consistent G-scenario of $\mathcal{N}'$ that is better than bestNg. At line 2, function exploreNeighborhoodAux prunes some non-feasible base relations of $\mathcal{N}'$ by enforcing $\overset{\diamond}{_G}$-consistency on $\mathcal{N}'$. We can show that the QCN $\mathcal{N}'$ given as parameter is always a consistent QCN whose constraints are defined by relations of $\hat{\mathsf{B}}$. Moreover, we know that G is a triangulation of $\mathrm{G}(\mathcal{N}')$ and the set of base relations B is such that partial $\diamond$-consistency implies minimality for $\widehat{\mathsf{B}}$ for any QCN defined on the subclass $\hat{\mathsf{B}}$. From this, we can assert that enforcing $\overset{\diamond}{_G}$-consistency on $\mathcal{N}'$ makes this QCN minimal *w.r.t* the graph G. Hence, after enforcing $\overset{\diamond}{_G}$-consistency on $\mathcal{N}'$ we have that $\mathcal{N}'$ is non trivially inconsistent and each base relation of its constraints is feasible (*i.e.* there exists at least one consistent scenario of $\mathcal{N}'$ satisfying it). After enforcing $\overset{\diamond}{_G}$-consistency on $\mathcal{N}'$, the integer α of non-overlapping constraints between $\mathcal{N}$ and $\mathcal{N}'$ is computed. In the case where $\alpha \geq$ bestα we know that all the consistent G-scenarios cannot be strictly better than bestNg, consequently the treatment terminates (line 3). In the contrary case, a pair of variables $(v, v') \in E$ such that the relation $\mathcal{N}'[v, v']$ is a non-singleton relation is selected. If such a pair of variables does not exist we can assert that $\mathcal{N}'$ is a consistent G-scenario that is better than bestNg. Hence, $\mathcal{N}'$ becomes the best found G-scenario (line 10). In the case where such a pair of variables exists, the search continues by exploring the two QCNs $\mathcal{N}'_{[v,v']/\{b\}}$ and $\mathcal{N}'_{[v,v']/(\mathcal{N}'[v,v']\setminus\{b\})}$, which are two strict subQCNs of $\mathcal{N}'$ that cover the consistent G-scenarios of $\mathcal{N}'$.

Proposition 4.1. *Let* $\mathcal{N} = (V, C)$ *and* $\mathcal{S} = (V, C')$ *be two* QCN*s of a qualitative calculus for which partial* $\diamond$*-consistency implies minimality for* $\widehat{\mathsf{B}}$ *and* $G = (V, E)$ *a triangulation of* $\mathrm{G}(\mathcal{N})$ *such that* $\mathcal{S}$ *is a consistent* G*-scenario. Then, the function* exploreNeighborhood *with parameters* $\mathcal{N}$, $\mathcal{S}$, G *returns a scenario of the set* $\mathsf{BestNb}(\mathcal{S}, G, \mathcal{N})$ *in the case where* $\mathsf{BestNb}(\mathcal{S}, G, \mathcal{N}) \neq \emptyset$*, and the scenario* $\mathcal{S}$ *in the contrary case.*

5 CROSSOVER OPERATORS FOR QCNS

In this section, we present some recombination operators that, given two consistent scenarios, build a new consistent scenario that integrates parts of the given scenarios, which we will refer to as the parents of the new scenario. The main difficulty concerning a crossover operator for QCNs is generating a good scenario fast; a good scenario is a scenario that is consistent and integrates the best parts of its parents w.r.t. to the unsatisfied constraints of the QCN for which we want to solve the MAX-QCN problem. For this purpose, we propose five crossover operators that are divided into two groups. The first group corresponds to the operators crossConsA and crossConsB whereas the second corresponds to the operators crossVarsA, crossVarsB, and crossVarsC. The main difference between these operators is that, given two scenarios, a crossCons operator considers the constraints of these two scenarios individually in order to build a new scenario whereas a crossVars operator considers as a first step the constraints of $\mathcal{S}_1$ and $\mathcal{S}_2$ with respect to a selection of subsets of variables. Figure 4 and Figure 5 illustrate these two approaches respectively.

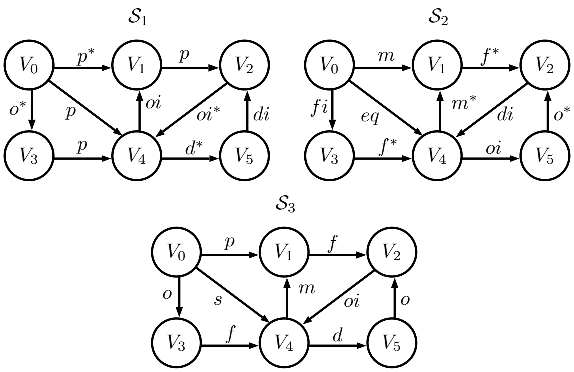

Figure 4: Illustration of a crossCons **operator that builds a scenario $\mathcal{S}_3$ from two scenarios $\mathcal{S}_1$ and $\mathcal{S}_2$ by individually selecting constraints from $\mathcal{S}_1$ and $\mathcal{S}_2$ (viz., the constraints marked by a star).**

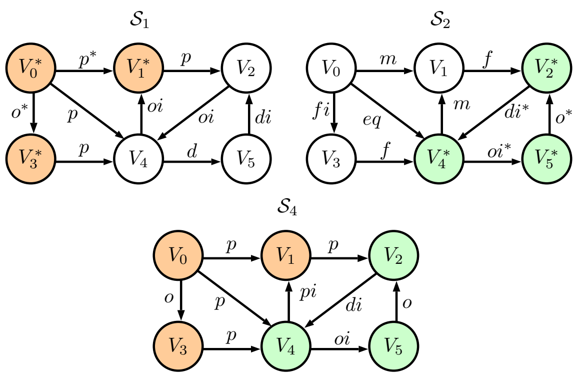

Figure 5: Illustration of a crossVars **operator that builds a scenario $\mathcal{S}_3$ from two scenarios $\mathcal{S}_1$ and $\mathcal{S}_2$ by selecting (as a first step) constraints of $\mathcal{S}_1$ and $\mathcal{S}_2$ with respect to some subsets of variables.**

Function crossConsA($\mathcal{N}$,G,$\mathcal{S}_1$,$\mathcal{S}_2$)

in : A QCN $\mathcal{N} = (V, C)$, a graph $G = (V, E)$ triangulation of $\mathrm{G}(\mathcal{N})$, two consistent G-scenarios $\mathcal{S}_1$ and $\mathcal{S}_2$ defined on V.
output : A consistent G-scenario on V.

begin
  $\mathcal{S} \leftarrow \top_V$;
  while *there exists* $(v, v') \in E$ *such that* $|\mathcal{S}[v, v']| > 1$ **do**
    Select randomly $(v, v') \in E$ such that $|\mathcal{S}[v, v']| > 1$;
    Select randomly $(\mathcal{S}'_1, \mathcal{S}'_2) \in \{(\mathcal{S}_1, \mathcal{S}_2), (\mathcal{S}_2, \mathcal{S}_1)\}$;
    if $\mathcal{S}[v, v'] \cap \mathcal{N}[v, v'] \neq \emptyset$ **then** $rel \leftarrow \mathcal{S}[v, v'] \cap \mathcal{N}[v, v']$;
    else $rel \leftarrow \mathcal{S}[v, v']$;
    if $\mathcal{S}'_1[v, v'] \cap rel \neq \emptyset$ **then** $rel \leftarrow \mathcal{S}'_1[v, v'] \cap rel$;
    else if $\mathcal{S}'_2[v, v'] \cap rel \neq \emptyset$ **then** $rel \leftarrow \mathcal{S}'_2[v, v'] \cap rel$;
    Select randomly a base relation $b \in rel$; $\mathcal{S} \leftarrow \mathcal{S}_{[v,v']/\{b\}}$; $\mathcal{S} \leftarrow \overset{\diamond}{_G}(\mathcal{S})$;
  return $\mathcal{S}$;

The crossover operator crossConsA. The operator crossConsA builds step by step a consistent scenario $\mathcal{S}$ from two parents $\mathcal{S}_1$ and $\mathcal{S}_2$ defined on a set V in an incremental way (see the corresponding function). Starting from an initialization of $\mathcal{S}$ by the QCN $\top_V$ (the QCN on the set of variables V for which all constraints on two different variables are defined by the universal relation B), step by step, this operator refines $\mathcal{S}$ until each one of its constraints is defined by a singleton relation. At each step, a pair of

---

Function crossVarsA($\mathcal{N}$,G,$\mathcal{S}_1$,$\mathcal{S}_2$)

in : A QCN $\mathcal{N} = (V, C)$, a graph $G = (V, E)$ triangulation of $\mathrm{G}(\mathcal{N})$, two consistent G-scenarios $\mathcal{S}_1$ and $\mathcal{S}_2$ defined on V.
output : A consistent G-scenario on V.

begin
  $V_1 \leftarrow V$; $V_2 \leftarrow \emptyset$;
  while $|V_1| > |V_2|$ **do**
    Select randomly $v \in V_1$;
    $V_1 \leftarrow V_1 \setminus \{v\}$; $V_2 \leftarrow V_2 \cup \{v\}$;
  $\mathcal{S} \leftarrow (\mathcal{S}_1)_{\downarrow V_1} \cup (\mathcal{S}_2)_{\downarrow V_2}$;
  while *there exists* $(v, v') \in E$ *such that* $|\mathcal{S}[v, v']| > 1$ **do**
    Select randomly $(v, v') \in E$ such that $|\mathcal{S}[v, v']| > 1$;
    if $\mathcal{S}[v, v'] \cap \mathcal{N}[v, v'] \neq \emptyset$ **then** $rel \leftarrow \mathcal{S}[v, v'] \cap \mathcal{N}[v, v']$;
    else $rel \leftarrow \mathcal{S}[v, v']$;
    Select randomly a base relation $b \in rel$; $\mathcal{S} \leftarrow \mathcal{S}_{[v,v']/\{b\}}$; $\mathcal{S} \leftarrow \overset{\diamond}{_G}(\mathcal{S})$;
  return $\mathcal{S}$;

Function crossVarsC($\mathcal{N}$,G,$\mathcal{S}_1$,$\mathcal{S}_2$)

in : A QCN $\mathcal{N} = (V, C)$, a graph $G = (V, E)$ triangulation of $\mathrm{G}(\mathcal{N})$, two consistent G-scenarios $\mathcal{S}_1$ and $\mathcal{S}_2$ defined on V.
output : A consistent G-scenario on V.

begin
  $V_1 \leftarrow \emptyset$; $V_2 \leftarrow \emptyset$;
  foreach $v \in V$ **do**
    if *there exists* $v' \in V$ *such that* $(v, v') \in E$ *and* $\mathcal{S}_1[v, v'] \cap \mathcal{N}[v, v'] = \emptyset$ **then** $V_2 \leftarrow V_2 \cup \{v\}$;
    else $V_1 \leftarrow V_1 \cup \{v\}$;
  $\mathcal{S} \leftarrow (\mathcal{S}_1)_{\downarrow V_1} \cup (\mathcal{S}_2)_{\downarrow V_2}$;
  while *there exists* $(v, v') \in E$ *such that* $|\mathcal{S}[v, v']| > 1$ **do**
    Select randomly $(v, v') \in E$ such that $|\mathcal{S}[v, v']| > 1$;
    if $\mathcal{S}[v, v'] \cap \mathcal{N}[v, v'] \neq \emptyset$ **then** $rel \leftarrow \mathcal{S}[v, v'] \cap \mathcal{N}[v, v']$;
    else $rel \leftarrow \mathcal{S}[v, v']$;
    Select randomly a base relation $b \in rel$; $\mathcal{S} \leftarrow \mathcal{S}_{[v,v']/\{b\}}$; $\mathcal{S} \leftarrow \overset{\diamond}{_G}(\mathcal{S})$;
  return $\mathcal{S}$;

---

variables (v, v') of $\mathcal{S}$ for which the corresponding constraint is defined by a non singleton relation is randomly selected, and the two consistent scenarios $\mathcal{S}_1$ and $\mathcal{S}_2$ are randomly ordered in a pair $(\mathcal{S}'_1, \mathcal{S}'_2)$. Then, the relation *rel* containing the future considered possible base relations for the constraint of $\mathcal{S}$ between the variables v and v' is built. The relation *rel* will be defined by the first non empty relation among the following ordered list of relations : $\mathcal{S}[v, v'] \cap \mathcal{N}[v, v'] \cap \mathcal{S}'_1[v, v']$, $\mathcal{S}[v, v'] \cap \mathcal{N}[v, v'] \cap \mathcal{S}'_2[v, v']$, $\mathcal{S}[v, v'] \cap \mathcal{S}'_1[v, v']$, $\mathcal{S}[v, v'] \cap \mathcal{S}'_2[v, v']$ and $\mathcal{S}[v, v']$. By this definition, we try to prioritize some constraints provided by the two parents $\mathcal{S}_1$ and $\mathcal{S}_2$. After the relation *rel* has been defined, a base relation $b \in rel$ is randomly chosen to define the new constraint of $\mathcal{S}$ between v and v'. In order to maintain $\mathcal{S}$ globally consistent, the closure w.r.t. a triangulation G of $\mathcal{N}$ is realized at the end of each main loop.

The crossover operator crossConsB. The operator crossConsB is very similar to the operator crossConsA; it differs from the latter operator in the order of the constraints to be handled in the main loop. Indeed, the operator crossConsB, contrary to the operator crossConsA, selects in priority in the main loop a pair of variables (v, v') among the set of pairs of variables (v, v') such that $\mathcal{S}[v, v'] \cap \mathcal{N}[v, v']$ is a non empty relation before considering the set of pairs of variables (v, v') such that $\mathcal{S}[v, v'] \cap \mathcal{N}[v, v']$ is an empty relation.

The crossover operator crossVarsA. With respect to the operator crossVarsA, another policy is used to integrate parts of the two parents $\mathcal{S}_1$ and $\mathcal{S}_2$ into a consistent scenario $\mathcal{S}$. Indeed, contrary to the two previous operators, the operator crossVarsA integrates into

$\mathcal{S}$ a projection of $\mathcal{S}_1$ and a projection of $\mathcal{S}_2$ w.r.t. a set of variables V_1 and a set of variables V_2 respectively such that $V = V_1 \cup V_2$. By examining the function corresponding to the operator crossVarsA, we can note that V_1 and V_2 are randomly defined and in a balanced manner (*i.e.* the cardinalities of these two sets differ in number by not more than one). After integrating the two projections $(\mathcal{S}_1)_{\downarrow V_1}$ and $(\mathcal{S}_2)_{\downarrow V_2}$ into $\mathcal{S}$ by initializing $\mathcal{S}$ by the union of these two projections, the treatment of the operator crossVarsA consists in refining $\mathcal{S}$ into a consistent scenario with a treatment similar to the one used by the operator crossConsA.

The crossover operator crossVarsB. The operator crossVarsB is very similar to the operator crossVarsA; it differs from the latter operator in the combination of the projections for the initialization of $\mathcal{S}$. Indeed, for the operator crossVarsB, $\mathcal{S}$ is initialized by the QCN $\mathcal{N}_1 = (\mathcal{S}_1)_{\downarrow V_1} \cup (\mathcal{S}_2)_{\downarrow V_2}$ or by the QCN $\mathcal{N}_2 = (\mathcal{S}_1)_{\downarrow V_2} \cup (\mathcal{S}_2)_{\downarrow V_1}$ according to the numbers of non overlapping constraints α_1 and α_2 between $\mathcal{N}$ and respectively the QCNs $\mathcal{N}_1$ and $\mathcal{N}_2$. In the case where $\alpha_1 \leq \alpha_2$, $\mathcal{S}$ is initialized by $\mathcal{N}_1$. In the contrary case, $\mathcal{S}$ is initialized by $\mathcal{N}_2$.

The crossover operator crossVarsC. This operator is similar to the two previous operators, it differs from the operator crossVarsA in the projections considered for the initialization of $\mathcal{S}$ and more particularly in the definitions of the used sets of variables V_1 and V_2. Indeed, for the operator crossVarsC, V_1 is defined by the maximal subset of variables of V which are not implied by a constraint of $\mathcal{S}_1$ not satisfying a constraint of the QCN $\mathcal{N}$. More formally, $V_1 = \bigcup\{v : \text{ there not exist } (v, v') \in E \text{ such that } \mathcal{S}[v, v'] \cap \mathcal{N}[v, v'] = \emptyset\}$ and $V_2 = V \setminus V_1$. Intuitively, this operator tries to complete the satisfactory constraints of $\mathcal{S}_1$ by constraints of $\mathcal{S}_2$ in order to generate a new consistent scenario.

Concerning the proposed crossover operators, we have the following property:

Proposition 5.1. *Let* $\mathcal{N} = (V, C)$, $\mathcal{S}_1 = (V, C^1)$ *and* $\mathcal{S}_2 = (V, C^2)$ *be three* QCN*s of a qualitative calculus for which partial* $\diamond$*-consistency implies minimality for* $\widehat{\mathsf{B}}$ *and* $G = (V, E)$ *a triangulation of* $\mathrm{G}(\mathcal{N})$ *such that* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *are two consistent* G*-scenarios. Then, for each crossover operator* Crossover $\in$ {crossConsA, crossConsB, crossVarsA, crossVarsB, crossVarsC}*, the operator* Crossover *with parameters* $\mathcal{N}$, G, $\mathcal{S}_1$, $\mathcal{S}_2$ *returns a consistent* G*-scenario.*

6 EXPERIMENTS

In the conducted experiments, we consider QCNs of IA used in the experiments reported in [7, 8]. These QCNs are randomly generated using the model $A(n, d, s)$ proposed by Nebel in [23] and extensively used in the literature concerning qualitative constraint networks. The parameter n of this model is the number of variables of the generated QCNs, d is the density of constraints defined by a relation other than the trivial relation (*i.e.* B), and s is the average number of base relations in each constraint. For the number of variables $n = 20$, the average number of base relations $s = 6.5$ and the density of constraints $d \in \{8, 9, 10, 11, 12, 13, 14\}$, 10 instances were generated. Hence, we conducted experiments on 70 different QCNs.

Concerning triangulations of the constraint graphs of QCNs, we also use the same ones as those used in [7, 8]. These triangulations were generated using a greedy triangulation algorithm (see the GreedyFillIn heuristic in [4]). Except if mentioned otherwise, given a QCN $\mathcal{N}$ as parameter to EAMQ, the parameter corresponding to the graph G is computed using this triangulation algorithm. Moreover, our experiments have been conducted on a Quad-core Intel XEON X5550 with 32Gb of memory and the method EAMQ has been implemented in Java. [1]

The first part of our analysis concerns a comparison between the different crossovers operators. In Table 1 we report the results of executions of EAMQ with configurations that differ in the cardinality of the population ($card_P$), the number of best scenarios selected at each generation ($card_{Best}$), and the number of loops between each diversification ($divT$). The timeout limit has been fixed to 180 seconds. By examining the number of QCNs for which a scenario with the exact optimal number of unsatisfied constraints has been found (the value #*F*), we note that, in general, the operators crossConsA and crossConsB outperform the other operators. The operators crossVarsA and crossVarsB are the worst performers. This observation is confirmed by the examination of the value $\overline{\mathrm{dif}_{\mathrm{ov}}}$, which corresponds to the average of the difference beetween the number of constraints unsatisfied by the scenario returned by EAMQ and the number of constraints unsatisfied by an optimal scenario (*i.e.* $\overline{\mathrm{dif}_{\mathrm{ov}}} = \frac{1}{70}\Sigma_{i \in 1,\ldots,70}(\alpha(\mathcal{N}_i, \mathcal{S}_i^{\mathrm{found}}) - \alpha(\mathcal{N}_i, \mathcal{S}_i^{\mathrm{opt}}))$. Concerning this indicator, the best values for the crossovers crossConsA, crossConsB, crossVarsA, crossVarsB, and crossVarsC are respectively 0.92, 0.92, 1.62, 1.37, and 1.48. Now, by considering the value #*B*, which corresponds to the number of QCNs for which the crossover operation is better than the other crossover operators (by comparing the difference between the number of constraints unsatisfied by the scenario returned and the value of an optimal scenario and by using the running time as a tie-breaker), we notice that in general crossConsB outperforms the other operators. The best performance of this operator with respect to the other operators is clearly confirmed by comparing the different indicators concerning the portfolios (the best values of all configurations are taken into account) of the different crossover operators. To end this first analysis, we can notice that performances of the different operators are not the same for a given configuration. It seems that the operators crossVarsA, crossVarsB and crossVarsC are performing better with a rather great value of $divT$ (400,600), whereas for the operators crossConsA, crossConsB smaller values seem to lead to better performances. Concerning the size of the considered population, for almost all the crossover operators a size of less than 200 leads to better performance.

The second part of our analysis focuses on the best configurations for each of the crossover operators. For each operator, we select a configuration for which #*F* is maximal and a configuration for which $\overline{\mathrm{dif}_{\mathrm{ov}}}$ is minimal. In Table 1, these retained configurations correspond to lines with values in bold. Focusing on these selected configurations, experiments have been conducted with a timeout limit of 20 minutes. The corresponding results are reported in Table 2. As with the previous analysis, we clearly constate that the crossover operators crossConsA and crossConsB greatly outperform the other operators. By considering the indicator $\overline{\mathrm{dif}_{\mathrm{ov}}}$, we notice that the performance gaps between the different operators

[1] The program and the benchmark can be found with the authors on request.

			crossConsA			crossConsB			crossVarsA			crossVarsB			crossVarsC		
c_P	c_{Best}	$divT$	#F	$\overline{\text{dif}_{\text{ov}}}$	#B	#F	$\overline{\text{dif}_{\text{ov}}}$	#B	#F	$\overline{\text{dif}_{\text{ov}}}$	#B	#F	$\overline{\text{dif}_{\text{ov}}}$	#B	#F	$\overline{\text{dif}_{\text{ov}}}$	#B
60	20	50	27	0.94	29	**29**	0.98	34	6	3.2	1	8	2.52	1	15	2.32	5
60	20	100	27	**0.92**	26	27	1.0	33	11	2.52	3	14	2.02	2	16	2.41	6
60	20	200	23	1.07	23	24	1.08	33	12	2.01	2	15	1.68	3	17	2.35	9
60	20	400	22	1.18	23	26	1.04	30	17	1.67	3	16	1.52	5	16	2.42	9
60	20	600	19	1.22	23	24	1.12	30	15	**1.62**	4	**18**	1.52	6	15	2.28	7
100	20	50	26	1.04	33	24	0.98	29	9	2.4	0	13	1.91	3	16	2.6	5
100	20	100	22	1.11	32	24	1.04	26	12	2.04	5	14	1.65	3	17	2.58	4
100	20	200	22	1.18	25	26	0.98	30	18	1.8	5	14	1.44	5	17	2.67	5
100	20	400	20	1.25	28	24	1.05	27	18	1.71	4	17	**1.37**	7	18	2.7	4
100	20	600	21	1.24	24	25	1.07	26	**21**	1.64	8	16	1.37	7	17	2.5	5
160	30	50	12	2.48	24	14	2.41	19	1	8.92	0	2	7.15	0	14	2.15	27
160	30	100	17	1.57	25	18	1.42	29	2	7.04	0	2	5.78	0	16	2.02	16
160	30	200	**29**	0.94	29	25	1.01	30	3	5.61	0	2	4.01	0	20	1.74	11
160	30	400	27	0.95	29	27	0.94	32	3	4.15	0	5	2.87	0	21	1.74	9
160	30	600	26	1.0	27	26	0.94	33	3	3.7	0	8	2.57	0	**24**	1.62	10
160	40	50	16	1.81	22	17	1.48	30	2	7.6	0	2	6.47	0	11	1.94	18
160	40	100	25	1.22	27	24	1.04	35	2	6.18	0	2	5.05	0	11	1.87	8
160	40	200	29	0.95	28	26	0.94	31	3	4.67	0	4	3.54	0	19	1.62	11
160	40	400	28	0.95	28	26	**0.92**	32	4	3.51	0	6	2.64	0	19	1.65	10
160	40	600	25	1.0	24	25	0.98	32	7	2.84	2	10	2.34	0	18	1.62	12
260	40	50	11	3.0	17	15	3.22	25	1	9.51	0	2	7.94	0	10	2.14	28
260	40	100	19	1.82	21	18	1.75	35	2	7.7	0	2	6.15	0	14	1.92	14
260	40	200	23	1.31	22	24	1.12	37	2	6.01	0	2	4.8	0	14	1.77	11
260	40	400	23	1.07	26	26	1.08	31	2	4.65	0	4	3.54	0	20	1.52	13
260	40	600	27	1.0	25	23	1.08	33	3	4.04	0	5	3.01	0	20	**1.48**	12
260	60	50	18	1.94	23	15	1.92	25	1	8.1	0	2	6.88	0	14	2.05	22
260	60	100	26	1.3	28	22	1.12	29	2	6.38	0	2	5.4	0	16	1.8	13
260	60	200	26	1.1	27	26	0.94	31	2	5.07	0	3	3.81	0	17	1.67	12
260	60	400	28	1.01	30	23	0.98	28	3	3.77	0	4	2.84	0	19	1.52	12
260	60	600	25	1.07	29	24	1.04	27	6	3.38	0	7	2.52	0	19	1.48	14
Portfolio			44	0.44	22	48	0.37	30	29	1.0	6	27	0.91	2	41	0.55	10

Table 1: Results of the execution of EAMQ **with different configurations (one configuration by line in the table) for each crossover operator. The timeout limit is** 180 **seconds and the GreedyFillIn triangulations are used for all the configurations. The parameters** $card_P$**,** $card_{Best}$**, and** $divT$ **correspond to the columns titled** c_P**,** c_{Best}**, and** $divT$ **respectively.** $\#F$ **corresponds to the number of** QCNs **for which** EAMQ **found an optimal partial scenario.** $\overline{\text{dif}_{\text{ov}}}$ **is the average of the difference between the number of constraints unsatisfied by the scenario returned and the number of constraints unsatisfied by an optimal scenario.** $\#B$ **corresponds to the number of** QCNs **for which the considered crossover is better than the other crossovers.**

				Timeout limit : 20 mn				Timeout limit : 3 h			
c_P	c_{Best}	$divT$	Crossover	#F	$\overline{\text{dif}_{\text{ov}}}$	#B	$\overline{\text{Time}}$	#F	$\overline{\text{dif}_{\text{ov}}}$	#B	$\overline{\text{Time}}$
60	20	100	crossConsA	36	0.68	20	173.4	44	0.47	18	1104.4
160	30	200	crossConsA	35	0.75	3	134.6	41	0.62	3	605.16
60	20	50	crossConsB	36	0.74	21	105.8	47	0.41	22	1017.01
160	40	400	crossConsB	37	0.7	7	121.2	41	0.5	7	643.4
60	20	600	crossVarsA	24	1.11	3	180.6	30	0.81	3	1437.6
100	20	600	crossVarsA	25	1.17	6	175.8	31	0.88	7	1026.1
60	20	600	crossVarsB	22	1.18	1	148.5	29	0.85	0	1354.6
100	20	400	crossVarsB	23	1.11	2	136.6	34	0.72	3	1396.5
160	30	600	crossVarsC	27	1.25	6	135.9	30	0.97	6	796.1
260	40	600	crossVarsC	22	1.24	1	118.7	26	0.95	1	931.4
Portfolio				57	0.18	–	92.2	61	0.12	–	284.6

Table 2: Results of the execution of EAMQ **on the** 70 **considered** QCNs **of** IA **with different selected configurations for each crossover operator with a timeout limit equal to** 20 **minutes and a timeout limit equal to** 3 **hours.**

are significant enough. By considering the portfolio, we constate that for a timeout limit of 20 minutes an optimal scenario has been characterized for 57 QCNs. This could be explained by hard instances belonging to the used benchmark [8].

Now, our analysis concerns two of the main ingredients of the proposed method EAMQ, namely the use of triangulations of the constraint graphs of the QCNs, and the neighborhood exploration that is realized at each generation of a new QCN. In order to do this, we compare EAMQ with different variations. For these experiments, we focus on the first configurations already treated in the results reported in Table 1 by using the crossover operator crossConsA and a timeout limit equal to 180s. By examining the reported results in Table 3 and by comparing the values of the indicators $\#F$ and $\overline{\text{dif}_{\text{ov}}}$, we constate that EAMQ using both the triangulations obtained by the algorithm *GreedyFillIn* and the neigborhood exploration technique (column titled without GIF&NE), greatly outperforms EAMQ using just the neigborhood exploration technique (column titled without GIF) and EAMQ using just *GreedyFillIn* triangulations (column titled without NE). This observation has also been made for all configurations with which we have experimented, these

			with GIF&NE		without GIF		without NE	
c_P	c_{Best}	$divT$	#F	$\overline{\text{dif}_{\text{ov}}}$	#F	$\overline{\text{dif}_{\text{ov}}}$	#F	$\overline{\text{dif}_{\text{ov}}}$
60	20	50	27	0.94	13	1.45	14	1.85
60	20	100	27	0.92	11	1.57	13	1.85
60	20	200	23	1.07	13	1.54	10	1.88
60	20	400	22	1.18	11	1.65	12	2.07
60	20	600	19	1.22	13	1.58	10	2.01
Portfolio			32	0.74	17	1.31	16	1.51

Table 3: Results of the execution of EAMQ on the 70 considered QCNs of IA for configurations corresponding to $cardP = 60$, $card_{Best} = 20$, the crossover crossConsA, and a timeout limit of 180 seconds. GFI denotes the use of the $GreedyFillIn$ triangulation heuristic. NE denotes the use of the neigbourhood exploration technique.

				#B	
c_P	c_{Best}	$divT$	Crossover	180 s	20 mn
60	20	100	crossConsA	21	19
160	30	200	crossConsA	2	2
60	20	50	crossConsB	19	15
160	40	400	crossConsB	5	5
60	20	600	crossVarsA	0	1
100	20	600	crossVarsA	4	3
60	20	600	crossVarsB	1	0
100	20	400	crossVarsB	1	1
160	30	600	crossVarsC	4	5
260	40	600	crossVarsC	1	1
			QLS	12	18

Table 4: Comparison of EAMQ with QLS.

results are not reported here due to space limitations. This experimental evaluations seems to validate the use of triangulations of the constraint graphs of the QCNs and the neighborhood exploration technique in the context of the proposed method EAMQ.

The last part of our analysis concerns a comparison between our approach and the one proposed in [7] that proposes a local search algorithm called QLS in order to solve the MAX-QCN problem. For this analysis, we focus on the configurations of EAMQ previously selected and compare them with QLS with different timeout limits. By examining the results concerning the indicator #B reported in Table 4 we can constate that two of the ten selected configurations of EAMQ outperform QLS for a timeout limit of 180 seconds and one of them is better than QLS for a timeout limit of 20 minutes.

7 CONCLUSION

In this paper we focused on the MAX-QCN problem [9]. To solve this problem, we proposed an original hybrid evolutionary algorithm, which we call EAMQ. This algorithm integrates several novel ingredients in the context of QCNs such as crossover operators, neighborhood exploration, diversification, and triangulations of QCNs. The preliminary experiments that we have conducted with temporal qualitative constraint networks of the Interval Algebra, show the interest of our approach for solving the MAX-QCN problem. Future work consists of conducting experiments with calculi other than IA, such as RCC8, and with larger QCNs (whose number of variables is greater than 20). Another perspective is to integrate the techniques implemented in EAMQ in a complete algorithm for solving the MAX-QCN problem.

REFERENCES

[1] J-F. Allen. 1981. An interval-based representation of temporal knowledge. In *IJCAI*.

[2] James F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Commun. ACM* 26, 11 (1983), 832–843.

[3] Mehul Bhatt, Hans Guesgen, Stefan Wölfl, and Shyamanta Hazarika. 2011. Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions. *Spatial Cognition & Computation* 11 (2011), 1–14.

[4] Hans L. Bodlaender and Arie M. C. A. Koster. 2010. Treewidth computations I. Upper bounds. *Inf. Comput.* 208 (2010), 259–275.

[5] Juan Pablo Cares, María Cristina Riff Rojas, and Ignacio Araya. 2013. LS2R: A local search algorithm to solve scheduling radiotherapy problems. In *HIS*.

[6] Assef Chmeiss and Jean-François Condotta. 2011. Consistency of Triangulated Temporal Qualitative Constraint Networks. In *ICTAI*.

[7] Jean-François Condotta, Ali Mensi, Issam Nouaouri, Michael Sioutis, and Lamjed Ben Saïd. 2016. Local Search for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In *AIMSA*.

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

[9] Jean-François Condotta, Ali Mensi, Issam Nouaouri, Michael Sioutis, and Lamjed Ben Saïd. 2015. A Practical Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In *ICTAI*.

[10] Bart G. W. Craenen and A. E. Eiben. 2005. Hybrid evolutionary algorithms for constraint satisfaction problems: memetic overkill?. In *CEC*.

[11] Bart G. W. Craenen, A. E. Eiben, and Jano I. van Hemert. 2003. Comparing evolutionary algorithms on binary constraint satisfaction problems. *IEEE Trans. Evolutionary Computation* 7, 5 (2003), 424–444.

[12] Bart G. W. Craenen and Ben Paechter. 2006. A Tabu Search Evolutionary Algorithm for Solving Constraint Satisfaction Problems. In *PPSN*.

[13] Bart G. W. Craenen and Ben Paechter. 2008. A Conflict Tabu Search Evolutionary Algorithm for Solving Constraint Satisfaction Problems. In *EvoCOP*.

[14] Reinhard Diestel. 2012. *Graph Theory, 4th Edition*. Graduate texts in mathematics, Vol. 173. Springer.

[15] A. E. Eiben, J. K. van der Hauw, and Jano I. van Hemert. 1998. Graph Coloring with Adaptive Evolutionary Algorithms. *J. Heuristics* 4, 1 (1998), 25–46.

[16] Xingxing Hao and Jing Liu. 2017. A multiagent evolutionary algorithm with direct and indirect combined representation for constraint satisfaction problems. *Soft Comput.* 21, 3 (2017), 781–793.

[17] S.M. Hazarika. 2012. *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*. Igi Global.

[18] Miyuki Koshimura, Tong Zhang, Hiroshi Fujita, and Ryuzo Hasegawa. 2012. QMaxSAT: A Partial Max-SAT Solver. *JSAT* 8, 1/2 (2012), 95–100.

[19] Gérard Ligozat and Jochen Renz. 2004. What is a qualitative calculus? A general framework. In *PRICAI*.

[20] Zhiguo Long and Sanjiang Li. 2015. On Distributive Subalgebras of Qualitative Spatial and Temporal Calculi. In *COSIT*.

[21] Elizabeth Montero and María Cristina Riff. 2014. Evaluating components of artificial immune algorithms: A Performance-aware Method based on Evolutionary Calibrator. In *SMC*.

[22] Elizabeth Montero, María Cristina Riff, and Bertrand Neveu. 2010. An evaluation of off-line calibration techniques for evolutionary algorithms. In *GECCO*.

[23] B. Nebel. 1996. Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficienty of Using the ORD-Horn Class. In *ECAI*.

[24] Sanja Petrovic and Edmund K. Burke. 2004. University Timetabling. In *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*.

[25] David A. Randell, Zhan Cui, and Anthony G. Cohn. 1992. A Spatial Logic based on Regions and Connection. In *KR*.

[26] J. Renz and B. Nebel. 2007. Qualitative Spatial Reasoning Using Constraint Calculi. In *Handbook of Spatial Logics*. Springer Netherlands, 161–215.

[27] María Cristina Riff, Teddy Alfaro, Xavier Bonnaire, and Carlos Grandón. 2008. EA-MP: An evolutionary algorithm for a mine planning problem. In *CEC*.

[28] Michael Sioutis and Manolis Koubarakis. 2012. Consistency of Chordal RCC-8 Networks. In *ICTAI*.