Merging Qualitative Information: Rationality and Complexity

Frank Dylla[1], Jan Oliver Wallgrün[2], and Jasper van de Ven[1]

[1] Cognitive Systems, University of Bremen, Germany
{dylla,vandeven}@cs.uni-bremen.de
[2] Human Factors in GIScience Lab, Pennsylvania State University
wallgrun@psu.edu

Abstract. This paper addresses the problem of merging several constraint networks over the same qualitative calculus representing information from different sources into a single consistent constraint network. We define the general problem, discuss rationality criteria, and describe a distance-based approach using the notion of conceptual neighborhood between the relations. Our approach is relation-based rather than model-based and, hence, can also be used to relax a single inconsistent network. We investigate the properties of the proposed merging operators, describe an algorithm for their computation, and discuss its complexity. We also argue that the merging problem is still not sufficiently understood and deserves further research.

1 Introduction

While checking consistency and constraint propagation in qualitative constraint networks (QCNs) have been extensively studied wrt. complexity and design of efficient algorithms in the qualitative spatial and temporal reasoning (QSTR) community (see [1, 2] for overviews), the same cannot be said for a second class of important reasoning problems often referred to as neighborhood-based reasoning tasks because the underlying concept is the notion of conceptual neighborhood [3] between spatial (or temporal) relations. Alternative denominations are similarity- or distance-based qualitative reasoning.

We address one particular neighborhood-based reasoning problem arising in the context of qualitative spatial representations: the problem of merging constraint networks over the same calculus in a suitable way (discussed later in the paper) and ensuring that the result is always a consistent network. This problem occurs, for instance, when merging information from several databases containing qualitative information or when combining the beliefs of multiple agents (e.g., humans or robots) expressed in a qualitative way. One important particularity of the QCNs that has to be taken into account when defining suitable merging operators is that in contrast to similar merging problems in propositional logic it is often not possible to express all disjunctions of possible models without admitting additional models. This aspect will play an important role in the investigations presented in this paper. A related problem not covered in this paper is the revision of qualitative information, i.e., dealing with changing beliefs regarding a world when new information becomes available, as studied for example by Hue and Westphal [4].

The problem of merging information from several constraint networks into a single consistent network can be seen as the task of finding a consistent network that is in some sense "in the middle" or "approximately equally close" to the individual input networks. Shortest path distance in the conceptual neighborhood graph is typically the distance measure of choice to calculate the distance between two base relations but there are many ways how neighborhood distances can be aggregated to capture the distance between entire constraint networks [5–8].

The merging problem as considered in this paper subsumes the problem of resolving conflicts or contradictions in a single network by turning to the closest consistent network or relaxing the constraints until the network becomes consistent. In earlier work [7], we developed a first approach to relax a single inconsistent constraint network. The merging operators we describe in this paper are a direct extension of this idea to the more general problem of merging several qualitative constraint networks. We relate QCN merging to work on logic-based merging [9, 10] (as originally suggested in [11, 12]) and formulate clear rationality criteria for merging operators. In contrast to the merging operators defined in [11] which are *model-based* in the sense that the result only depends on the models of the input networks, we aim for a *relation-based approach* in which every relation contained in the input QCN is able to affect the merging result. We argue that this approach is advantageous in many application scenarios, in particular when considering spatial database integration where the spatial relations stored in the database are based on independent observations (for a concrete example, cf. Sec. 2.3). In addition, this approach allows us to also deal with inconsistent input networks, which is not directly possible in a model-based framework.

With this paper, we are aiming at drawing the attention of researchers from the field of QSTR to the problem of merging QCNs and neighborhood-based reasoning problems in general. It is our hope that this will lead to a better understanding of the complexity of these problems and availability of improved algorithms in the future. The remainder of the paper is structured as follows: We first formalize the merging problem and related concepts (Section 2). We then discuss the issue of rationality criteria that suitable merging operators should satisfy (Section 3). In Section 4 we summarize our merging operators, explain what is known about their properties wrt. the defined rationality criteria, and present an algorithm to compute the merging results for these operators. Additionally, we discuss the complexity of the algorithm. We close with a brief discussion on our approach (Section 5).

2 Merging Problem and Operators

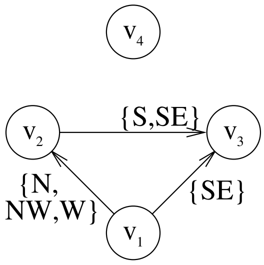
2.1 Qualitative calculi, QCNs, and Consistency

In this paper, we restrict ourselves to spatial calculi over binary relations. However, the methods described here can be adapted to relations of higher arity or other domains, e.g., temporal relations, as well. In the remainder of the paper, we will refer to the following concepts and notations: A *qualitative spatial calculus* $\mathcal{C}$ defines a set $\mathcal{B}_\mathcal{C}$ of jointly exhaustive and pairwise disjoint spatial relations over a domain of spatial objects $\mathcal{D}_\mathcal{C}$ (e.g., points, lines, regions). For example, $\mathcal{C}$ could define a set of cardinal directions north-of (N), northwest-of (NW), west-of (W), southwest-of (SW), etc. plus the *identity relation* equal (EQ) for points in the plane. To be able to express incomplete or imprecise spatial knowledge, the qualitative spatial calculus actually is concerned with the so-called set of general relations $\mathcal{R}_\mathcal{C}$ containing all possible unions of base relations. We adopt the often used way of notating general relations as sets of base relations instead of unions, meaning that $\mathcal{R}_\mathcal{C} = 2^{\mathcal{B}_\mathcal{C}}$ and that, for instance, $A\,\{\mathrm{NE}, \mathrm{N}, \mathrm{NW}\}\,B$ means A is either to the northeast, north, or northwest of B. Complete ignorance is expressed by the universal relation $U = \{b \in \mathcal{B}_\mathcal{C}\}$. Another special relation is the empty relation $\varnothing$ which cannot be realized by any pair of objects.

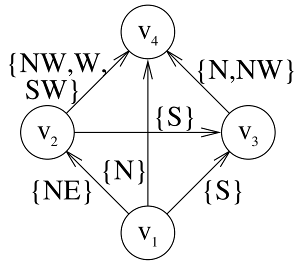
In addition to defining relations, a qualitative calculus also defines a set $\mathcal{O}_\mathcal{C} = \{\cap, \cup, ^-, ^\smile, \circ\}$ of operations over $\mathcal{R}_\mathcal{C}$. $\cap$, $\cup$, and $^-$ are the operations of intersection, union, and complement which keep their set-theoretic meaning. The unary operation $^\smile$ is the converse operation which tells us the relation holding between B and A from the relation holding between A and B, e.g., $\{\mathrm{N}\}^\smile = \{\mathrm{S}\}$. The binary composition operation $\circ$ yields the relation that has to hold between A and C when we know the relation holding between A and B as well as between B and C, e.g., $\{\mathrm{N}\} \circ \{\mathrm{SW}\} = \{\mathrm{NW}, \mathrm{W}, \mathrm{SW}\}$.

A *qualitative constraint network* (QCN) specifies the spatial arrangement of objects O_i in terms of relations from a qualitative calculus $\mathcal{C}$ (see Fig. 1). It is a directed graph $G = (V, E)$ in which the vertices (or variables) v_i represent the objects and the directed edges e_i are associated with the spatial relation holding between the objects by a function $C : V^2 \to \mathcal{R}_\mathcal{C}$ mapping each pair of variables from V to a relation from $\mathcal{R}_\mathcal{C}$. We will use the abbreviation C_{ij} for $C(v_i, v_j)$. The relations can be seen as constraints that restrict which values of $\mathcal{D}_\mathcal{C}$ can be assigned to the objects O_i. The QCNs in Fig. 1 contain a vertex for every variable v_i and one directed edge for every pair of variables v_i, v_j with $i < j$ which is labeled by the corresponding relation. By convention, edges labeled with the universal relation U are omitted.

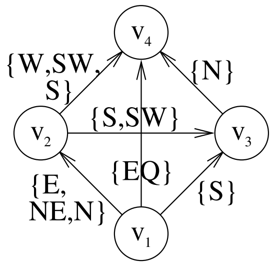
A QCN can be *consistent* (*satisfiable*) or not. An assignment of values from $\mathcal{D}_\mathcal{C}$ to the variables v_i is a *solution* if it satisfies all constraints C_{ij}. A QCN N is *consistent* if it has at least one solution. A QCN s is called *atomic* or a *scenario* if all C_{ij} consists of a single base relation. We say that a scenario $s = (V, C')$ *is a scenario of QCN* $N = (V, C)$ if all $C'_{ij} \subseteq C_{ij}$. We will use a predicate *consistent(N)* to state that QCN N is consistent. We denote the set of all scenarios



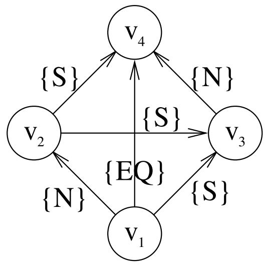
(a) N_1: contains 119 consistent scenarios



(b) N_2: inconsistent



(c) N_3: contains 3 consistent scenarios



(d) a scenario of N_3

Fig. 1. Input QCNs N_1 to N_3 and a scenario of N_3.

(not necessarily consistent ones) of N as $\langle\!\langle N \rangle\!\rangle$ and the set of all *consistent* scenarios as $[\![N]\!]$. $\mathcal{QCN}$ will refer to the constraint network in which all constraints C_{ij} are U for a given set of variables V and a given calculus $\mathcal{C}$ and, hence, $\langle\!\langle \mathcal{QCN} \rangle\!\rangle$ stands for the set of all possible scenarios (not necessarily consistent ones) given V and $\mathcal{C}$. Figs. 1(a)–1(d) show several exemplary QCNs with cardinal direction constraints. The QCN in Fig. 1(d) is a scenario of the QCN in Fig. 1(c).

Deciding consistency of QCNs is NP-complete for many calculi but often tractable subalgebras are known. There exist two main methods for deciding consistency, both based on techniques developed for discrete CSPs. The so-called algebraic closure algorithm enforces a local consistency called path-consistency [13] and runs in $O(n^3)$ time for n variables. If algebraic closure is not sufficient to decide consistency for the relations occurring in the network, a backtracking search is performed [14] that recursively splits the constraints, until a level is reached which can be checked with algebraic closure.

2.2 Conceptual Neighborhood

Our merging approach is based on the notion of similarity or distance between QCNs. Similarity is related to how the relations of the QCN can change, an aspect which is described by the notion of *conceptual neighborhood* introduced in [3]. Two base relations of a spatial calculus are *conceptually neighbored*, if they can be continuously transformed into each other without resulting in a third relation in between. For instance, N is conceptually neighbored to NW but not to W as one would have to pass through at least one other base relation (e.g., NW). The concrete conceptual neighborhood relation depends on the concrete set of continuous transformations one considers [3, 7] which in turn need to be grounded in spatial change over time [15]. For this work, it is sufficient to assume that a suitable conceptual neighbor relation has been defined which is irreflexive and symmetric. It can be represented by the so-called *conceptual neighborhood graph* $\mathcal{CNG}$ as illustrated in Fig. 2. Typically, when using the graph to measure the distance between two base relations in terms of their shortest path distance, all edges are assumed to have a uniform weight of 1 but approaches with different weights per edge have also been proposed, e.g. in [16, 17].

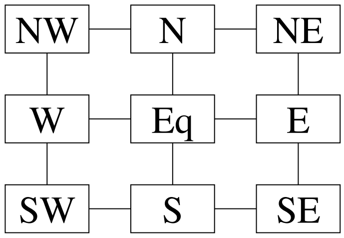


Fig. 2. A conceptual neighborhood graph for the cardinal direction calculus [18].

2.3 The QCN merging problem

The QCN merging problem addressed in this paper, can be formalized as follows (adopting a notation similar to that used in [11]):

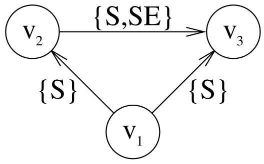
Definition 1 (QCN Merging Problem) *Given an input set $\mathcal{N} = (N_1, N_2, ...N_n)$ of QCNs $N_k = (V, C_k)$ over the same set of variables V and the same qualitative spatial calculus $\mathcal{C}$, representing information from different sources about a static arrangement of objects, the task is to derive a consistent QCN $N' = \Delta(\mathcal{N})$ according to some well-defined merging operator $\Delta(\mathcal{N})$.*

An exemplary merging problem could be to merge the QCNs N_1, N_2, N_3 from Fig. 1. We here only make the restriction that all N_k are over the same set of variables V for convenience. As long as the correct correspondences between variables are known, a preprocessing step renaming variables and, if needed, adding new ones connected via the univesal relation U to all other variables can be employed to transform the QCNs into QCNs over the same variable sets.

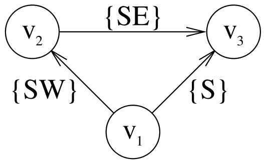
Obviously, the above definition specifies the input and output of the QCN merging problem but everything else depends on the chosen merging operator $\Delta(\mathcal{N})$. Therefore, we will in the next section look into merging operators in more detail and discuss criteria for suitable merging operators. For now we restrict ourselves to the one main criterion that the output network should be consistent, meaning it needs to have at least one consistent scenario.

Two straightforward ways of combining QCNs are integrating them conjunctively or disjunctively. Combining the QCNs conjunctively (later written as $N_1 \cap N_2$) means to construct a new QCN by taking the intersection of the relations making up corresponding constraints. Obviously, the resulting QCN can be inconsistent, even when the input networks themselves are all consistent. This is the case if the QCNs do not share a consistent scenario. Hence, intersection in general is too strict to serve as a suitable merging operator. On the other hand, combining networks disjunctively (later written as $N_1 \cup N_2$) means to take the union over corresponding relations. Using the union, consistent scenarios of the input networks are preserved but new ones may appear and the result will often be very unspecific reducing its usability. In addition, if all input networks are inconsistent, the resulting QCN may still be inconsistent.

In this work, we are interested in merging operators that are guaranteed to return a consistent result even when the input QCNs are not consistent (we only assume that all $C_{ij} \neq \emptyset$). Adopting the idea of *distance-based merging* [10, 9], we want our solution to be based on those models (consistent scenarios in our case) that are as close as possible to all input networks simultaneously in a way that we will explain below. A main difference to existing work on merging QCNs [11, 12] is that we assume that all relations in the QCN can be considered independent and equally reliable information pieces that can have an effect on the result of the merging, while in the other approaches the result only depends on relations belonging to consistent scenarios. Consequentially, we will refer to our approach as *relation-based* in contrast to the *model-based* paradigm employed in the other approaches. To make this difference more clear, consider the example shown in Fig. 3. For convenience we introduce a compact notation for QCNs with three variables: A QCN $N = (V, C)$ with variables v_1, v_2, and v_3 is written as a triple of constraints $N = (C_{12}, C_{13}, C_{23})$. The input set in the example consists of the two QCNs $N_1 = (\{\text{S}\}, \{\text{S}\}, \{\text{S}, \text{SE}\})$ and $N_2 = (\{\text{SW}\}, \{\text{S}\}, \{\text{SE}\})$. N_1 has one consistent scenario, namely $(\{\text{S}\}, \{\text{S}\}, \{\text{S}\})$, while N_2 itself is a consistent scenario. As the model-based approach presented in [11] basically ignores relations that are not part of a consistent scenario such as SE in C_{23} of N_1, it would consider the consistent scenarios $(\{\text{SW}\}, \{\text{S}\}, \{\text{SE}\})$ and $(\{\text{S}\}, \{\text{S}\}, \{\text{S}\})$ as equally plausible resolutions of the conflicts between the two QCNs. However,



(a) N_1



(b) N_2

Fig. 3. Merging example: Input QCNs N_1 and N_2.

while the former scenario can be explained by a single small observation error in N_1 (C_{12} should have been SW instead of S), the latter would mean that there have been two small observation errors in N_2. In contrast to such model-based operators, the merging operators we are going to define in Sec. 4 will treat the scenario ({SW}, {S}, {SE}) as a more plausible explanation.

Before we introduce the operators themselves, we start by formulating rationality criteria for relation-based QCN merging operators. As mentioned in the introduction, one particularity of the QCN merging scenario distinguishing it, for instance, from merging problems in propositional logic is that it is not always possible to combine models (or here consistent scenarios) into a single representation without obtaining additional models. Unfortunately, in many situations (e.g., merging databases) maintaining multiple hypotheses is undesirable or infeasible because of the additional complexity of tracking multiple hypotheses about the state of the world simultaneously. Hence, we define relation-based merging operators $\Delta(\mathcal{N})$ with $\mathcal{N} = \{N_1, ...N_n\}$ that take an input set and return a single QCN and investigate how this requirement and the established rationality criteria fit together.

3 Rationality Criteria

To define the rationality criteria for our merging scenario, we follow criteria developed for information merging in a propositional setting (criteria (A1)–(A6) in [19] and (IC1)–(IC6) in [9]). Due to the special properties of QCNs and the fact that we are aiming at merging operators which are relation-based instead of model-based, we have to adapt the criteria leading to (Q1)–(Q6). The resulting set of criteria turns out to be a specialization of the generic criteria for QCN merging (N1)–(N6) described in [12] but without assuming consistency of the input QCNs. We will point out where we make stronger demands tailored towards our particular merging approach.

The most basic requirement is that the merging result is a consistent QCN. Therefore we demand that $\Delta(\mathcal{N})$ always has to be consistent. In contrast, instantiating (N1) in [12] for our case would only demand that a QCN is returned but not necessarily a consistent one.

(Q1) $consistent(\Delta(\mathcal{N}))$

If the intersection of the input QCNs already is consistent, Δ should yield exactly this intersection. Again, the corresponding criterion (N2) in [12] would only make a weaker demand allowing the merging result to be inconsistent.

(Q2) $consistent(\bigcap N_i) \Rightarrow \Delta(\mathcal{N}) = \bigcap N_i$

The third criterion defined in [19] formalizes the 'irrelevance of syntax'. Concerned with defining criteria for model-based merging operators, they demand that the result of merging should only depend on the models of the input knowledge bases. In our relation-based case, it only makes sense to demand a significantly weakened version of the third criterion, basically claiming that the order of input networks should not affect the result. Thus, we define when two input sets are equivalent.

Definition 2 (Equivalence (≡) of input sets) *Two input sets of QCNs* $\mathcal{N} = (N_1, ..., N_n)$ *and* $\mathcal{N}' = (N'_1, ..., N'_n)$ *are equivalent* ($\mathcal{N} \equiv \mathcal{N}'$) *iff there exists a bijection* f *between* $\mathcal{N}$ *and* $\mathcal{N}'$ *such that* N_k *and* $N'_l = f(N_k)$ *have the same scenarios, i.e.* $\langle\langle N_k \rangle\rangle = \langle\langle f(N_k) \rangle\rangle$ *for* $1 \leqslant k \leqslant n$.

(Q3) $\mathcal{N}_1 \equiv \mathcal{N}_2 \Rightarrow \Delta(\mathcal{N}_1) = \Delta(\mathcal{N}_2)$

The fourth criterion is concerned with fairness of the merging operator stating that it must not give preference to one of the input knowledge bases. Since we do not consider one knowledge base to be more reliable than another, when merging two QCNs N_1 and N_2 and there is a (not necessarily consistent) scenario s part of the merging result which is also a scenario of N_1, the same must hold for a scenario t of N_2. Intuitively, since there has to be a scenario in N_2 that is at least as close to N_1 as s is to N_2, not having a scenario from N_2 in the result would mean giving preference to N_1 over N_2.

(Q4) $\exists s : s \in \langle\!\langle \Delta((N_1, N_2)) \rangle\!\rangle \wedge s \in \langle\!\langle N_1 \rangle\!\rangle \Leftrightarrow \exists t : t \in \langle\!\langle \Delta((N_1, N_2)) \rangle\!\rangle \wedge t \in \langle\!\langle N_2 \rangle\!\rangle$

With the fifth property we demand that if we merge two input sets $\mathcal{N}_1$ and $\mathcal{N}_2$ individually and there is a scenario s part of both merging results, this scenario must also be part of the result of merging the input set resulting from combining the QCNs from $\mathcal{N}_1$ and $\mathcal{N}_2$ into a single set (written as $\mathcal{N}_1 \sqcup \mathcal{N}_2$).

(Q5) $\forall s : (s \in \langle\!\langle \Delta(\mathcal{N}_1) \rangle\!\rangle \wedge s \in \langle\!\langle \Delta(\mathcal{N}_2) \rangle\!\rangle \Rightarrow s \in \langle\!\langle \Delta(\mathcal{N}_1 \sqcup \mathcal{N}_2) \rangle\!\rangle)$

Finally, in (Q6) we demand that if $\Delta(\mathcal{N}_1)$ and $\Delta(\mathcal{N}_2)$ have a common scenario, the reverse direction of (Q5) is also true. Taken together (Q5) and (Q6) state that if for two input sets the merging agrees on certain scenarios, these scenarios should be exactly the scenarios of the resulting QCN of the combined input set.

(Q6) $\exists t : t \in \langle\!\langle \Delta(\mathcal{N}_1) \rangle\!\rangle \wedge t \in \langle\!\langle \Delta(\mathcal{N}_2) \rangle\!\rangle \Rightarrow (s \in \langle\!\langle \Delta(\mathcal{N}_1 \sqcup \mathcal{N}_2) \rangle\!\rangle \Rightarrow s \in \langle\!\langle \Delta(\mathcal{N}_1) \rangle\!\rangle \wedge s \in \langle\!\langle \Delta(\mathcal{N}_2) \rangle\!\rangle)$

We now proceed by defining our relation-based merging operators for QCNs and will later discuss to what extent they satisfy the rationality criteria defined here.

4 Merging Operators and Algorithm

4.1 Neighborhood distance based measures and operators

Above, we introduced the conceptual neighborhood graph as a way to measure the distance or similarity of the base relations of a calculus[3], assuming that variations are caused by imperfect observations. Seeing the conceptual neighborhood graph $\mathcal{CNG}_\mathcal{C}$ of a calculus $\mathcal{C}$ as an undirected graph (cmp. Sec. 2.2), we now define the distance $d_{B\leftrightarrow B}$ between the two base relations $b_i, b_j \in \mathcal{B}_\mathcal{C}$ as the shortest path distance between the corresponding nodes in the graph:

$$d_{B\leftrightarrow B}(b_i, b_j) = \text{shortest path distance between } b_i \text{ and } b_j \text{ in } \mathcal{CNG}_\mathcal{C} \qquad (1)$$

The next step is to define the distance between two atomic qualitative constraint networks $s = (V, C)$ and $s' = (V, C')$ over the same set of m variables and the same calculus. For this we need an aggregation operator that determines how the distances between constraints in s_i, s_j given by $d_{B\leftrightarrow B}(b_i, b_j)$ are combined. Natural candidates for this aggregation operator which we will denote as $\circledcirc$ are the sum or the max operator. The distance itself is defined as:

$$d^{\circledcirc}_{S\leftrightarrow S}(s, s') = \underset{1\leqslant i<j\leqslant m}{\circledcirc} d_{B\leftrightarrow B}(C_{ij}, C'_{ij}) \qquad (2)$$

The notion behind our merging operators $\Delta(\mathcal{N})$ is that the resulting QCN is built from the consistent scenarios that are closest to the input networks together with all inconsistent scenarios that are at most as distant as these consistent scenarios. Therefore, we further need to define the distance between a scenario and a general constraint network and based on that the distance between a scenario and the set of input networks $(\mathcal{N})$.

[3] We note that, in general, any distance measure can be applied. However, the choice of distance measure may affect the properties of $\Delta(\mathcal{N})$.

For determining how close a scenario s is to a constraint network N we consider all scenarios of N and take the distance to the closest one. The resulting distance $d^{\circledcirc}_{S\leftrightarrow N}(s,N)$ is then given by

$$d^{\circledcirc}_{S\leftrightarrow N}(s,N) = \min_{s' \in \langle\!\langle N \rangle\!\rangle} d^{\circledcirc}_{S\leftrightarrow S}(s,s') \tag{3}$$

To measure the distance between a scenario s and the set $\mathcal{N}$ of all input networks N_i, we need to aggregate over the individual distances $d^{\circledcirc}_{S\leftrightarrow N}(s,N_i)$. To do this, we introduce another aggregation operator $\circledast$. Again, sum and max seem to be natural candidates for this aggregation operator. In the general case, the resulting distance is given by

$$d^{\circledcirc,\circledast}_{S\leftrightarrow \mathcal{N}}(s,\mathcal{N}) = \underset{1\leqslant k\leqslant n}{\circledast} d^{\circledcirc}_{S\leftrightarrow N}(s,N_k) \tag{4}$$

To construct the final merging result we take the set $S^{\circledcirc,\circledast}(\mathcal{N})$ of all scenarios ($\langle\!\langle \mathcal{QCN} \rangle\!\rangle$) that are closer or as close to $\mathcal{N}$ as the closest consistent scenarios ($[\![\mathcal{QCN}]\!]$) wrt. $d^{\circledcirc,\circledast}_{S\leftrightarrow \mathcal{N}}$.

$$S^{\circledcirc,\circledast}(\mathcal{N}) = \{s \in \langle\!\langle \mathcal{QCN} \rangle\!\rangle \mid \forall s' \in [\![\mathcal{QCN}]\!] : d^{\circledcirc,\circledast}_{S\leftrightarrow \mathcal{N}}(s',\mathcal{N}) \geqslant d^{\circledcirc,\circledast}_{S\leftrightarrow \mathcal{N}}(s,\mathcal{N})\} \tag{5}$$

As the final step, the resulting QCN is constructed by taking the union of all the scenarios, i.e. union of the corresponding networks, in $S^{\circledcirc,\circledast}(\mathcal{N})$.

$$\Delta^{\circledcirc,\circledast}(\mathcal{N}) = \bigcup_{s \in S^{\circledcirc,\circledast}(\mathcal{N})} s \tag{6}$$

As discussed previously, the final union step may lead to additional scenarios in $\Delta^{\circledcirc,\circledast}(\mathcal{N})$ that are not contained in $S^{\circledcirc,\circledast}(\mathcal{N})$, i.e., having a larger distance to the given networks, which is the price one has to pay to end up with a single QCN.

4.2 Properties of the Operators

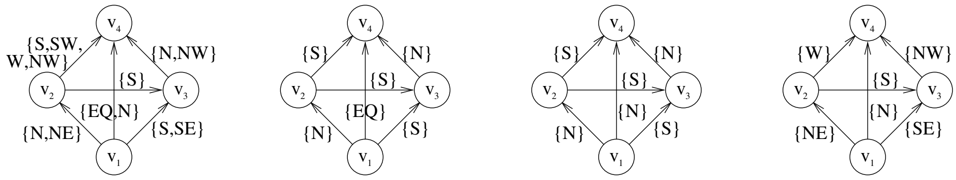
Table 1 summarizes the properties of the merging operators introduced in the previous section wrt. the rationality criteria given in Section 3 as proven and discussed in detail in [8]. So far, we have mainly been concerned with the question whether our merging operator $\Delta^{\circledcirc,\circledast}(\mathcal{N})$ satisfies the criteria for all combinations of $\circledcirc, \circledast \in \{\sum, max\}$ [4]. This is the case for the fundamental criteria (Q1)–(Q4). For (Q5) the situation is more complicated: In [8] we presented a counter example showing that $\Delta^{\Sigma,\Sigma}(\mathcal{N})$ does not satisfy the criterion. We then proved that in general (so for $\circledcirc, \circledast \in \{\sum, max\}$) (Q5) is satisfied by the scenarios in $S^{\circledcirc,\circledast}(\mathcal{N})$. This means that the loss of this property is a direct consequence of the final union step based on the demand that the merging result has to be a single QCN.

For (Q6), we presented a counter example that shows that the criterion is not satisfied by $\Delta^{\Sigma,\Sigma}(\mathcal{N})$ and $\Delta^{\Sigma,max}(\mathcal{N})$ and also not in $S^{\Sigma,\Sigma}(\mathcal{N})$ and $S^{\Sigma,max}(\mathcal{N})$. We currently do not know whether a similar counter example can be constructed for $\Delta^{max,\circledast}(\mathcal{N})$ (and $S^{max,\circledast}(\mathcal{N})$). It seems possible that this is the result of including inconsistent scenarios with a distance smaller than that of the closest consistent scenario. While it would be counterintuitive not to include these scenarios when inconsistent scenarios with a larger distance are included, the properties of such alternatives need to be investigated. In addition, the question on whether suitable operators can be found which do not suffer from similar problems caused by the final union step as we saw for (Q5) needs to be investigated as part of future research.

4.3 An Algorithm to Compute $\Delta^{\circledcirc,\circledast}(\mathcal{N})$

In [8], we described an algorithm to compute $\Delta^{\circledcirc,\circledast}(\mathcal{N})$. While the time complexity is still exponential in the worst-case, the algorithm is an improvement over brute-force methods described in [11] and based on the following two notions to significantly improve its performance in practice, in particular when the input QCNs are rather close to each other: (1) candidate scenarios are considered

[4] Merging operators are often classified into majority or arbitration operators. We briefly state here without proof that $\Delta^{\circledcirc,\max}$ satisfies the notion of an arbitration operator (see (A7) in [19]), while $\Delta^{\circledcirc,\Sigma}$ is preferable if a majority-based resolution is desired (see (M7) in [19]).



(a) $\Delta^{\Sigma,\Sigma}(\mathcal{N})$: has 64 scenarios, 6 consistent (b) consistent scenario with $d^{\Sigma,\Sigma}(s,\mathcal{N}) = 4$ (c) consistent scenario with $d^{\Sigma,\Sigma}(s,\mathcal{N}) = 4$ (d) consistent scenario with $d^{\Sigma,\Sigma}(s,\mathcal{N}) > 4$

Fig. 4. The solution of $\Delta^{\Sigma,\Sigma}(\mathcal{N})$ according to Alg.1.

in order of increasing distance to $\mathcal{N}$ as given by $d^{\circledcirc,\circledast}_{S\leftrightarrow\mathcal{N}}(s,\mathcal{N})$[5], and (2) the expensive consistency checking is delayed as long as possible and does not have to consider individual scenarios. For generation of scenarios in the order of increasing distance in the merging algorithm (see Alg. 1), we need *relax* functions at several levels: (a) the constraint level, (b) the network level, and (c) the level of sets of networks. Although, we develop them bottom up, i.e., from (a) to (c), they will be applied top-down in the algorithm, i.e., (c) to (a).

The function $relaxC(C, d_C)$ yields the relation consisting of all base relations which have minimal distance d_C for $d_C \geqslant 0$ to a base relation in constraint C_{ij}. That is, $relaxC(C, d_C)$ generates a D-coarse network as introduced by Li and Li [20].

$$relaxC(C_{ij}, d_C) = \left\{ b \in \mathcal{B}_\mathcal{C} \mid \min_{b' \in C_{ij}} d_{B\leftrightarrow B}(b, b') = d_C \right\} \quad [6] \tag{7}$$

Based on it, the function $relaxN^{\circledcirc}(N, d_N)$ yields a set of networks in which constraints C_{ij} have been changed using $relaxC$ with parameter e_{ij} so that aggregation with $\circledcirc$ over e_{ij} yields d_N.

$$\begin{aligned} &relaxN^{\circledcirc}(N = (V, C), d_N) = \\ &\left\{ N' = (V, C') \mid C'_{ij} = relaxC(C_{ij}, e_{ij}) \wedge \underset{1 \leqslant i,j \leqslant m}{\circledcirc} e_{ij} = d_N \right\} \end{aligned} \tag{8}$$

Thus, each scenario $s \in \langle\!\langle \mathcal{QCN} \rangle\!\rangle$ is a scenario of a network in $relaxN^{\circledcirc}(N, d_N)$ iff $d^{\circledcirc}_{S\leftrightarrow N}(s, N) = d_N$.

The function $relax\mathcal{N}^{\circledcirc,\circledast}(\mathcal{N}, d_\mathcal{N})$ then yields a set of modified input sets where each modified N_k has been modified using $relaxN^{\circledcirc}(N_k, e_k)$ such that the e_k are aggregated with $\circledast$ to $d_\mathcal{N}$.

$$\begin{aligned} &relax\mathcal{N}^{\circledcirc,\circledast}(\mathcal{N} = (N_1, N_2, ...N_n), d_\mathcal{N}) = \\ &\left\{ (N'_1, ..., N'_n) \mid N'_k = relaxN^{\circledcirc}(N_k, e_k) \wedge \underset{1 \leqslant k \leqslant n}{\circledast} e_k = d_\mathcal{N} \right\} \end{aligned} \tag{9}$$

Given this set of *relax* functions the basic version of our algorithm proceeds as follows (see Algorithm 1): The outer loop increases $d_\mathcal{N}$ and considers scenarios s with $d^{\circledcirc,\circledast}_{S\leftrightarrow\mathcal{N}}(s,\mathcal{N}) = d_\mathcal{N}$. The algorithm stops when there is at least one consistent scenario among these. All scenarios are

[5] In [11] at any stage all possible scenarios have to be considered

[6] We here assume that the distance function is integer-based, so that constraints can be relaxed in discrete steps (see line 12 of Algorithm 1). An adaptation to other distance functions is possible though.

merging operator	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**
$\Delta^{\Sigma,\Sigma}(\mathcal{N})$	✓	✓	✓	✓	✗ (holds for $S^{\Sigma,\Sigma}$)	✗ (also not for $S^{\Sigma,\Sigma}$)
$\Delta^{\Sigma,max}(\mathcal{N})$	✓	✓	✓	✓	? (holds for $S^{\Sigma,max}$)	✗ (also not for $S^{\Sigma,max}$)
$\Delta^{max,\Sigma}(\mathcal{N})$	✓	✓	✓	✓	? (holds for $S^{max,\Sigma}$)	?
$\Delta^{max,max}(\mathcal{N})$	✓	✓	✓	✓	? (holds for $S^{max,max}$)	?

Table 1. The properties of the merging operators

Algorithm 1 Merging algorithm

procedure $\Delta^{\circledtriangle,\circledast}(\mathcal{N})$
1: $S \leftarrow (V, C)$ with all $C_{ij} = \varnothing$
2: $d_{\mathcal{N}} \leftarrow 0$; $consistent \leftarrow false$
3: **repeat**
4: $\quad R \leftarrow relax\mathcal{N}^{\circledtriangle,\circledast}(\mathcal{N}, d_{\mathcal{N}})$
5: $\quad$ **for all** $(N'_1, \ldots N'_n) \in R$ **do**
6: $\quad\quad I \leftarrow \bigcap_{i=1}^{n} N'_i$
7: $\quad\quad$ **if** $\forall i, j : C_{ij} \neq \varnothing$ holds for I **then**
8: $\quad\quad\quad S \leftarrow S \cup I$
9: $\quad\quad\quad$ **if** $consistent(I)$ **then** $consistent \leftarrow true$ **end if**
10: $\quad\quad$ **end if**
11: $\quad$ **end for**
12: $\quad d_{\mathcal{N}} \leftarrow d_{\mathcal{N}} + 1$
13: **until** $consistent$
14: **return** S

collected in the QCN S which in the end will contain the merging result. To find the scenarios with $d^{\circledtriangle,\circledast}_{S\leftrightarrow\mathcal{N}}(s, \mathcal{N}) = d_{\mathcal{N}}$, the following happens inside the outer loop: Relaxed input sets are generated with $relax\mathcal{N}^{\circledtriangle,\circledast}(\mathcal{N}, d_{\mathcal{N}})$ for the given $d_{\mathcal{N}}$. For every element $N' = (N'_1, \ldots N'_n)$ of the resulting set, the algorithm takes the intersection over all N'_k. It can be shown, that $d^{\circledtriangle,\circledast}_{S\leftrightarrow\mathcal{N}}(s, \mathcal{N}) = d_{\mathcal{N}}$ holds for a scenario s iff s is a scenario of one of the networks I generated in this inner loop. Hence, when the intersection does not contain empty constraints (which would mean it does not have scenarios at all), we add all scenarios of I to S through the union operation in line 8. In addition, it is checked whether I is consistent using standard QSTR techniques. If consistent, we know that we have found at least one consistent scenario and the algorithm will stop after all remaining tuples in R have been processed (see line 5). The algorithm is available in the current version of the SparQ reasoning toolbox[7].

Fig. 4(a) shows an example of employing Algorithm 1 to compute $\Delta^{\Sigma,\Sigma}$ for merging the QCNs N_1 to N_3 from Fig. 1. The resulting network contains 64 scenarios, six of them consistent (for comparison the input QCNs have 2187, 6, and 18 scenarios). The minimum distance is three (four inconsistent scenarios) and the maximum distance is four. Two of the consistent scenarios (Figs. 4(b) and 4(c)) have distance four, the other four consistent scenarios with distance > 4 result from the final union step (e.g. Fig. 4(d)).

4.4 Complexity Considerations

Investigation of our algorithm has shown that the question of complexity is not simple to answer as it is highly dependent on the choice of aggregation operators and the calculus $\mathcal{C}$ used. To the best of our knowledge, the same holds for the QCN merging problem in general. Condotta et al. have described a naive brute-force approach that always checks the exponential worst-case number of $|\mathcal{B}_{\mathcal{C}}|^{|C|}$ scenarios individually [11], with $|\mathcal{B}_{\mathcal{C}}|$ denoting the number of base relations and $|C| = \frac{m(m-1)}{2}$ denoting the number of constraints. Each scenario has to be checked for consistency ($O(m^3)$ for algebraic-closure or worse). We assume that the same worst-case complexity applies to our algorithm, but as our scenario generation is iteratively built on the basis of the $\mathcal{CNG}$ and typically several scenarios are considered simultaneously in a single consistency check, the average-case performance of our algorithm is better. If the input networks already contain a common consistent scenario, our algorithm only computes the intersection and checks its consistency (best-case). Similarly, if the networks are already rather close to each other, i.e., only few constraints of the provided networks differ and are close in the $\mathcal{CNG}$, only few iterations (line 3) are necessary to find at least one consistent scenario.

[7] https://github.com/dwolter/SparQ

5 Concluding Discussions

We presented the QCN merging problem and a family of distance-based operators. The operators are relation-based in the sense that they treat every relation as an independent piece of information that may affect the result and can be applied to inconsistent input networks. Deviations from the criteria are partially due to the fact that QCNs cannot express all disjunctions of scenarios without leading to additional scenarios. Nevertheless, alternatives need to be evaluated as part of future research. We also presented an algorithm for computing the merging result by incrementally relaxing the input networks and delaying expensive consistency checking as long as possible in order to increase the average-case efficiency compared to a naive implementation. However, to our knowledge the complexity of the overall problem is still not well understood. We also expect that significant progress still can be achieved in designing more efficient merging approaches, once this problem and related neighborhood-based operations and reasoning tasks are given more attention.

References

1. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An overview. Fundamenta Informaticae **46**(1-2) (2001) 1–29
2. Renz, J., Nebel, B.: Qualitative spatial reasoning using constraint calculi. In Aiello, M., Pratt-Hartmann, I.E., van Benthem, J.F., eds.: Handbook of Spatial Logics. Springer (2007) 161–215
3. Freksa, C.: Conceptual neighborhood and its role in temporal and spatial reasoning. In Singh, M., Travé-Massuyès, L., eds.: Decision Support Systems and Qualitative Reasoning. (1991) 181 – 187
4. Hué, J., Westphal, M.: Revising qualitative constraint networks: Definition and implementation. In: IEEE 24th International Conference on Tools with Artificial Intelligence, ICTAI 2012, Athens, Greece, November 7-9, 2012. (2012) 548–555
5. Bruns, H.T., Egenhofer, M.J.: Similarity of spatial scenes. In: Seventh International Symposium on Spatial Data Handling, Taylor & Francis (1996) 173–184
6. Moratz, R., Freksa, C.: Spatial reasoning with uncertain data using stochastic relaxation. In Brauer, W., ed.: Fuzzy-Neuro Systems 98. Infix; Sankt Augustin (1998) 106–112
7. Dylla, F., Wallgrün, J.O.: Qualitative spatial reasoning with conceptual neighborhoods for agent control. Journal of Intelligent and Robotic Systems **48**(1) (2007) 55–78
8. Wallgrün, J.O., Dylla, F.: A relation-based merging operator for qualitative spatial data integration and conflict resolution. Technical Report 022-06/2010, SFB/TR 8 Spatial Cognition (2010)
9. Grégoire, É., Konieczny, S.: Logic-based approaches to information fusion. Information Fusion **7**(1) (2006) 4–18
10. Lin, J., Mendelzon, A.O.: Merging databases under constraints. International Journal of Cooperative Information Systems **7** (1996) 55–76
11. Condotta, J.F., Kaci, S., Schwind, N.: A framework for merging qualitative constraints networks. In Wilson, D., Lane, H.C., eds.: Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference, May 15-17, 2008, Coconut Grove, Florida, USA, AAAI Press (2008) 586–591
12. Condotta, J.F., Kaci, S., Marquis, P., Schwind, N.: Merging qualitative constraints networks using propositional logic. In: 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty(ECSQARU'09). (2009)
13. Mackworth, A.K.: Consistency in networks of relations. Artificial Intelligence **8** (1977) 99–118
14. Nebel, B.: Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ord-horn class. CONSTRAINTS **3**(1) (1997) 175–190
15. Galton, A.: Qualitative Spatial Change. Oxford University Press (2000)
16. Wallgrün, J.O., Yang, J., Klippel, A., Dylla, F.: Investigations into the cognitive conceptualization and similarity assessment of spatial scenes. In: Geographic Information Science - 7th International Conference Columbus, OH, USA, September 18-21, 2012. Proceedings. (2012) 212–225
17. Li, B., Fonseca, F.: TDD: A comprehensive model for qualitative spatial similarity assessment. Spatial Cognition and Computation **6**(1) (2006) 31–62
18. Goyal, R.K., Egenhofer, M.J.: Similarity of cardinal directions. In Goos, G., Hartmanis, J., van Leeuwen, J., eds.: Advances in Spatial and Temporal Databases. Volume 2121 of LNCS. Springer-Verlag Berlin Heidelberg (2001) 36–55
19. Konieczny, S., Pérez, R.P.: On the logic of merging. In: Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98). (1998) 488–498
20. Li, J.J., Li, S.: On finding approximate solutions of qualitative constraint networks. In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, Herndon, VA, USA, November 4-6, 2013. (2013) 30–37