Vertex Incremental Path Consistency for Qualitative Constraint Networks†

Michael Sioutis and Jean-François Condotta

Université Lille-Nord de France, Artois, CRIL-CNRS UMR 8188, Lens, France
{sioutis,condotta}@cril.fr

Abstract. The Interval Algebra (IA) and a subset of the Region Connection Calculus, namely, RCC-8, are the dominant Artificial Intelligence approaches for representing and reasoning about qualitative temporal and topological relations respectively. Such qualitative information can be formulated as a Qualitative Constraint Network (QCN). In this framework, one of the main tasks is to compute the path consistency of a given QCN. We propose a new algorithm that applies path consistency in a vertex incremental manner. Our algorithm enforces path consistency on an initial path consistent QCN augmented by a new temporal or spatial entity and a new set of constraints, and achieves better performance than the state-of-the-art approach. We evaluate our algorithm experimentally with QCNs of RCC-8 and show the efficiency of our approach.

1 Introduction

Spatial and temporal reasoning is a major field of study in Artificial Intelligence; particularly in Knowledge Representation. This field is essential for a plethora of areas and domains that include dynamic GIS, cognitive robotics, spatiotemporal design, and reasoning and querying with semantic geospatial query languages [3, 6, 8]. The Interval Algebra (IA) [1, 2] and a subset of the Region Connection Calculus [9], namely, RCC-8, are the dominant Artificial Intelligence approaches for representing and reasoning about qualitative temporal and topological relations respectively.

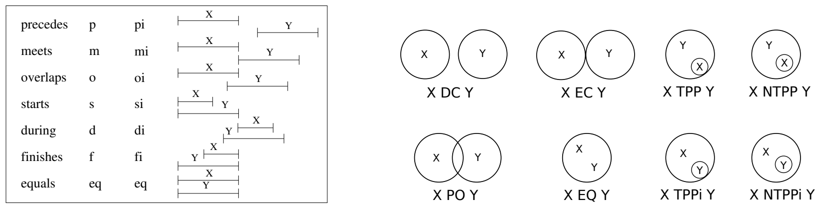
The state-of-the-art technique to decide whether a set of IA or RCC-8 relations is *path consistent* [13], considers the underlying complete graph of the respective constraint network all at once. However, due to the recent work of Huang [5] who showed that given a path consistent IA or RCC-8 network one can extend it arbitrarily with the addition of new temporal or spatial entities respectively, we could as well decide the path consistency of a constraint network by beginning with a subnetwork comprising a single temporal or spatial entity and extending it with a new entity at each step. This would allow us to work with a smaller underlying graph for each addition of a temporal or spatial entity, as opposed to considering the underlying graph of the entire constraint network

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

for all entities. The latter case is well described in the work of Gerevini [4, chapt. 3] for qualitative temporal reasoning who applies path consistency in an edge incremental manner obtaining a time complexity of $O(n^2 \cdot (n-1)) = O(n^3)$, where n is the number of the temporal entities. In short, to decide the path consistecy of a constraint network of n entities, edge incremental path consistency considers $O(n^2)$ constraints and for each contraint applies path consistency on the underlying complete graph of the network which is of degree $n-1$. The edge incremental path consistency described in the work of Gerevini, has been established as the state-of-the-art path consistency approach up to now. We will often refer to it simply as one-shot path consistency, since it can be performed in a single appliance of a path consistency algorithm that uses a queue initialized with all $O(n^2)$ constraints to reason with a network of n entities. Our approach is different and complementary to that of Gerevini, in that we process the constraint network in a vertex incremental manner, deciding or maintaining its path consistency bit by bit. To construct a path consistent network of n temporal or spatial entities, we apply path consistency $n-1$ times, one for every temporal or spatial entity that is added in the initial single-entity subnetwork. At each appliance, the underlying complete graph of the subnetwork along with the new entity has degree 1, ..., $n-1$ respectively, and the new entity also brings $O(1)$, ..., $O(n-1)$ constraints respectively, resulting in $O(1 \cdot 1 + \ldots + (n-1) \cdot (n-1))$ operations. Thus, we increase on average the performance of the path consistency algorithm, but do not improve its worst-case complexity which remains $O(n^3)$. In this paper, we make the following contributions: (*i*) we present an algorithm that maintains or decides the path consistency of an initial path consistent constraint network augmented by a new temporal or spatial entity and its accompanying constraints, and (*ii*) we implement our algorithm and evaluate it experimentally with QCNs of RCC-8, showing the efficiency of our approach.

2 Preliminaries

A (binary) qualitative temporal or spatial constraint language [11] is based on a finite set B of *jointly exhaustive and pairwise disjoint* (JEPD) relations defined on a domain D, called the set of base relations. The set of base relations B of a particular qualitative constraint language can be used to represent definite knowledge between any two entities with respect to the given level of granularity. B contains the identity relation Id, and is closed under the converse operation ($^{-1}$). Indefinite knowledge can be specified by unions of possible base relations, and is represented by the set containing them. Hence, 2^{B} will represent the set of relations. 2^{B} is equipped with the usual set-theoretic 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 two relations s and t for a set of base relations B is defined as the strongest relation $r \in 2^{\mathsf{B}}$ which contains $s \circ t$, or formally, $s \diamond t = \{b \in \mathsf{B} \mid b \cap (s \circ t) \neq \emptyset\}$, where $s \circ t = \{(x,y) \mid \exists z : (x,z) \in s \wedge (z,y) \in t\}$ is the relational composition. As illustration, consider the qualitative temporal constraint language IA [2], and



(a) The base relations of IA (b) The base relations of RCC-8

Fig. 1: IA and RCC-8 constraint languages

the qualitative spatial constraint language RCC-8 [9]. The set of base relations of IA is the set $\{eq, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$. These thirteen relations represent the possible relations between *time intervals*, as depicted in Figure 1a. The set of base relations of RCC-8 is the set $\{dc, ec, po, tpp, ntpp, tppi, ntppi, eq\}$. These eight relations represent the binary topological relations between *regions* that are non-empty regular subsets of some topological space, as depicted in Figure 1b (for the $2D$ case). IA and RCC-8 networks are qualitative constraint networks (QCNs), with relation eq being the identity relation in both cases.

Definition 1. *A* RCC-*8, or* IA*, network is a pair* $\mathcal{N} = (V, C)$ *where* V *is a finite set of variables and* C *a mapping associating a relation* $C(v, v') \in 2^{\mathsf{B}}$ *to each pair* (v, v') *of* $V \times V$*.* C *is such that* $C(v, v) \subseteq \{eq\}$ *and* $C(v, v') = (C(v', v))^{-1}$*.*

Given a QCN $\mathcal{N} = (V, C)$ and a new temporal or spatial entity α accompanied by mapping C' that associates a relation $C(\alpha, v) \in 2^{\mathsf{B}}$ to each pair (α, v) of $\{\alpha\} \times V$, $\mathcal{N} \uplus \alpha$ denotes the QCN $\mathcal{N}'' = (V'', C'')$, where $V'' = V \cup \{\alpha\}$, and C'' is a mapping that associates a relation $C(v, v') \in 2^{\mathsf{B}}$ to each pair (v, v') of $V \times V$ and a relation $C(\alpha, v) \in 2^{\mathsf{B}}$ to each pair (α, v) of $\{\alpha\} \times V$. In what follows, $C(v_i, v_j)$ will be also denoted by C_{ij}. Checking the consistency of a QCN of IA or RCC-8 is $\mathcal{NP}$-hard in general [7, 12]. However, there exist large maximal tractable subsets of IA and RCC-8 which can be used to make reasoning much more efficient even in the general $\mathcal{NP}$-hard case. These maximal tractable subsets are the sets $\hat{\mathcal{H}}_8, \mathcal{C}_8$, and $\mathcal{Q}_8$ for RCC-8 [10] and $\mathcal{H}_{\mathsf{IA}}$ for IA [7]. Consistency checking is then realised by a path consistency algorithm that iteratively performs the following operation until a fixed point $\overline{C}$ is reached: $\forall i, j, k$ do $C_{ij} \leftarrow C_{ij} \cap (C_{ik} \diamond C_{kj})$, where variables i, k, j form triangles that belong to the underlying complete graph of the input network [13]. If $C_{ij} = \emptyset$ for a pair (i, j) then C is inconsistent, otherwise $\overline{C}$ is *path consistent*. If the relations of the input QCN belong to some tractable subset of relations, path consistency implies consistency, otherwise a backtracking algorithm decomposes the initial relations into subrelations belonging to some tractable subset of relations spawning a branching search tree [14]. Thus, the performance of path consistency is crucial for the overall performance of a reasoner, since path consistency can be used to solve tractable networks, and can be run as the preprocessing and the consistency checking step of a backtracking algorithm.

3 iPC+ algorithm

In this section we present a new algorithm that enforces path consistency in a vertex increment manner. We call our algorithm iPC+, where symbol + is only used to differentiate it from the edge incremental path consistency algorithm of Gerevini [4, chapt. 3], as we consider extensions of a given QCN with a new temporal or spatial entity accompanied by new sets of constraints.

Function iPC+($\mathcal{N} \uplus \alpha$)

in : A QCN $\mathcal{N} \uplus \alpha = (V'', C'')$, where $\mathcal{N} = (V, C)$ is the initial path consistent QCN augmented by a new temporal or spatial entity α.

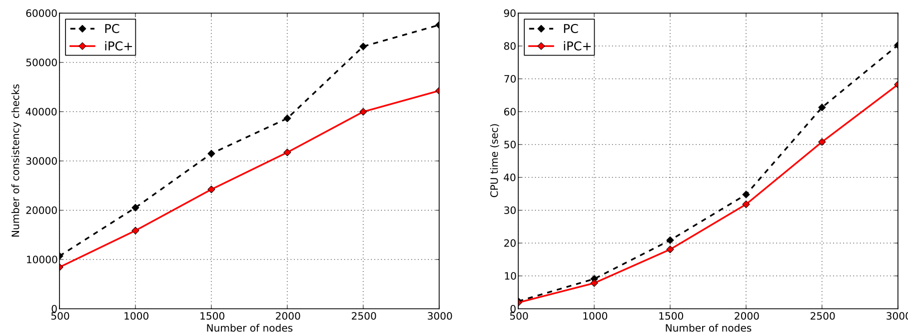
output : False if network $\mathcal{N} \uplus \alpha$ results in a trivial inconsistency (contains the empty relation), True if the modified network $\mathcal{N} \uplus \alpha$ is path consistent.

```
1  begin
2      Q ← {(i, j) | (i, j) ∈ V × {α}};
3      while Q ≠ ∅ do
4          (i, j) ← Q.pop();
5          foreach k ← 1 to V″, (i ≠ k ≠ j) do
6              t ← C″_ik ∩ (C″_ij ⋄ C″_jk);
7              if t ≠ C″_ik then
8                  if t = ∅ then return False;
9                  C″_ik ← t; C″_ki ← t^{-1};
10                 Q ← Q ∪ {(i, k)};
11             t ← C″_kj ∩ (C″_ki ⋄ C″_ij);
12             if t ≠ C″_kj then
13                 if t = ∅ then return False;
14                 C″_kj ← t; C″_jk ← t^{-1};
15                 Q ← Q ∪ {(k, j)};
16     return True;
```

iPC+ receives as input a QCN $\mathcal{N} \uplus \alpha = (V'', C'')$, where $\mathcal{N} = (V, C)$ is the initial path consistent QCN augmented by a new temporal or spatial entity α. The output of algorithm iPC+ is False if network $\mathcal{N} \uplus \alpha$ results in a trivial inconsistency, and True if the modified network $\mathcal{N} \uplus \alpha$ is path consistent. The queue data structure is instatiated by the set of edges $(i, j) \in V \times \{\alpha\}$ (line 2), i.e., the set of edges corresponding to the new temporal or spatial entity α. Path consistency is then realised by iteratively performing the following operation until a fixed point $\overline{C''}$ is reached: $\forall i, j, k$ perform ${C''}_{ij} \leftarrow {C''}_{ij} \cap ({C''}_{ik} \diamond {C''}_{kj})$, where edges $(i, k), (k, j) \in V'' \times V''$ (line 5).

Theorem 1 *For a given* QCN $\mathcal{N} \uplus \alpha = (V'', C'')$ *of* RCC-*8, or* IA*, where* $\mathcal{N} = (V, C)$ *is the initial path consistent* QCN *augmented by a new temporal or spatial entity* α*, function* iPC+ *correctly enforces path consistency on* QCN $\mathcal{N} \uplus \alpha$*.*

If we start with a single-entity QCN and extend it one entity at a time applying iPC+ in total $n-1$ times, it follows that we will obtain a time complexity of $O(1 \cdot 1 + \ldots + (n-1) \cdot (n-1)) = O(1/6 \cdot (n-1) \cdot n \cdot (2n-1))$ for constructing a



(a) Number of consistency checks

(b) CPU time (sec)

Fig. 2: Performance of iPC+ and PC for QCNs of RCC-8

QCN of n temporal or spatial entities, which is an improvement on average over the strict $O(n^3)$ complexity of the one-shot path consistency algorithm (PC).

4 Experimental evaluation

We generated random RCC-8 networks using the $\mathsf{A(n, d, l)}$ model [13]. In short, model $\mathsf{A(n, d, l)}$ creates random networks of size n, degree d, and an average number l of RCC-8 relations per edge. We considered network sizes between 500 and 3000 with a 500 step and $l = 4$ $(= |\mathsf{B}|/2)$ relations per edge. For each size series we created 70 networks that span over a degree d between 8.0 and 11.0 with a 0.5 step, i.e., 10 network instances were generated for each degree. For model $\mathsf{A(n, d, l)}$, a degree d between 8 and 11 belongs to the phase transition of RCC-8 relations, and, hence, guarantees hard and more time consuming, in terms of solubility, instances for the path consistency algorithm [13]. The experiments were carried out on a computer with an Intel Core 2 Duo P7350 processor with a CPU frequency of 2.00 GHz, 4 GB RAM, and the Lucid Lynx x86_64 OS (Ubuntu Linux). The python implementations of iPC+ and PC, were run with the CPython interpreter (`http://www.python.org/`), which implements Python 2. Only one of the CPU cores was used for the experiments. Regarding iPC+, we begin with a single node and grow the network one node at a time.[1]

A *consistency check* takes place whenever we apply the intersection operator ($\cap$) between two constraints (lines 6 and 11). This parameter is critical as the consistency check operation lies in the core of a path consistency algorithm. Results on the average number of consistency checks that each algorithm performs are shown in Figure 2a. On average, iPC+ performs 22.5% less consistency checks than PC, and 23.2% less in the final step in particular, where the networks of 3000 nodes are considered. Let us now see how all these numbers translate to CPU time. A diagrammatic comparison on the CPU time for each algorithm is

[1] All tools and datasets used here can be acquired upon request from the authors or found online in the following address: `http://www.cril.fr/~sioutis/work.php`.

shown in Figure 2b. On average, iPC+ runs 14.4% faster than PC, and 15.0% faster in the final step in particular (68 sec for iPC+ and 80 sec for PC), where the networks of 3000 nodes are considered. Similar results were obtained for IA that we omit to present here due to space constraints.

5 Conclusion and Future work

In this paper we presented an algorithm, viz., iPC+, for maintaining or deciding the path consistency of an initial path consistent constraint network augmented by a new temporal or spatial entity and its accompanying constraints. Experimental evaluation with QCNs of RCC-8 showed that iPC+ is able to perform better than PC for random networks of model $\mathsf{A(n, d, l)}$. Future work consists of evaluating our approach more thoroughly with structured and real datasets, and using chordal graphs to obtain a vertex incremental *partial* path consistency variant of our algorithm.

References

1. Allen, J.F.: An Interval-Based Representation of Temporal Knowledge. In: IJCAI (1981)
2. Allen, J.F.: Maintaining knowledge about temporal intervals. CACM 26, 832–843 (1983)
3. Bhatt, M., Guesgen, H., Wölfl, S., Hazarika, S.: Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions. Spatial Cognition & Computation 11, 1–14 (2011)
4. Gerevini, A.: Incremental qualitative temporal reasoning: Algorithms for the point algebra and the ord-horn class. Artif. Intell. 166(1-2), 37–80 (2005)
5. Huang, J.: Compactness and its implications for qualitative spatial and temporal reasoning. In: KR (2012)
6. Koubarakis, M., Kyzirakos, K.: Modeling and Querying Metadata in the Semantic Sensor Web: The Model stRDF and the Query Language stSPARQL. In: ESWC (2010)
7. Nebel, B.: Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class. In: ECAI (1996)
8. Open Geospatial Consortium: OGC GeoSPARQL - A geographic query language for RDF data. OGC® Implementation Standard (2012)
9. Randell, D.A., Cui, Z., Cohn, A.: A Spatial Logic Based on Regions and Connection. In: KR (1992)
10. Renz, J.: Maximal Tractable Fragments of the Region Connection Calculus: A Complete Analysis. In: IJCAI (1999)
11. Renz, J., Ligozat, G.: Weak Composition for Qualitative Spatial and Temporal Reasoning. In: CP (2005)
12. Renz, J., Nebel, B.: Spatial Reasoning with Topological Information. In: Spatial Cognition (1998)
13. Renz, J., Nebel, B.: Efficient Methods for Qualitative Spatial Reasoning. JAIR 15, 289–318 (2001)
14. Renz, J., Nebel, B.: Qualitative Spatial Reasoning Using Constraint Calculi. In: Handbook of Spatial Logics, pp. 161–215 (2007)