Efficient Spatial Reasoning with Rectangular Cardinal Relations and Metric Constraints

Angelo Montanari[1], Isabel Navarrete[2], Guido Sciavicco[2], and Alberto Tonon[3]

[1] Department of Mathematics and Computer Science. University of Udine, Italy
[2] Department of Information Engineering. University of Murcia, Spain
[3] eXascale Infolab. University of Fribourg, Switzerland
angelo.montanari@uniud.it, {inava,guido}@um.es,
alberto.tonon@unifr.ch

Abstract. In many real-world applications of knowledge representation and reasoning formalisms, one needs to cope with a number of spatial aspects in an integrated and efficient way. In this paper, we focus our attention on the so-called Rectangular Cardinal Direction calculus for qualitative spatial reasoning on cardinal relations between rectangles whose sides are parallel to the axes of a fixed reference system. We show how to extend its convex tractable fragment with metric constraints preserving tractability. The resulting formalism makes it possible to efficiently reason about spatial knowledge specified by one qualitative constraint network and two metric networks (one for each spatial dimension). In particular, it allows one to represent definite or imprecise knowledge on directional relations between rectangles and to derive additional information about them, as well as to deal with metric constraints on the height/width of a rectangle or on the vertical/horizontal distance between the sides of two rectangles. We believe that the formalism features a good combination of simplicity, efficiency, and expressive power, making it adequate for spatial applications like, for instance, web-document query processing and automatic layout generation.

Keywords: Qualitative spatial reasoning, Quantitative spatial reasoning, Cardinal direction relations, Constraint satisfaction problems.

1 Introduction

Qualitative spatial representation and reasoning play an important role in various areas of computer science such as, for instance, geographic information systems, spatial databases, document analysis, layout design, and image retrieval. Different aspects of space, such as direction, topology, size, and distance, which must be dealt with in a coherent way in many real-world applications, have been modeled by different formal systems (see [5] for a survey). For practical reasons, a bidimensional space is commonly assumed, and spatial entities are represented by points, boxes, or polygons with a variety of shapes, depending on the required level of detail.

Information about spatial configurations is usually specified by constraint networks describing the allowed binary relations between pairs of spatial variables. The main problem in qualitative spatial reasoning is to decide whether or not a given network

J. Filipe and A. Fred (Eds.): ICAART 2012, CCIS 358, pp. 234–249, 2013.

has a solution, that is, to establish whether or not there exists an assignment of domain values to variables that satisfies all constraints (*consistency checking*).

Cardinal relations are directional relations that allow one to specify how spatial objects are placed relative to each other either by making use of a fixed reference system, e.g., to say that an object is to the "north" or "southwest" of another one in a geographic space, or, alternatively, by exploiting directions as "above" or "below and left" in a local space. The most expressive formalism with cardinal relations between plane regions is the *Cardinal Direction calculus* (*CD-calculus* for short) [10,14,22]. The consistency problem for the CD-calculus is NP-complete; moreover, in [12], it has been shown that there exists no tractable fragment of it containing all basic constraints. Such a restriction is a serious limitation when we have to deal with incomplete or indefinite information in spatial applications.

A restricted version of the CD-calculus, called *Rectangular Cardinal Direction calculus* (*RCD-calculus*), has been introduced in [19,18], where cardinal relations are defined only between rectangles whose sides are parallel to the axes of the Euclidean plane. Rectangles of this type (aka *boxes*) can be seen as *minimum bounding rectangles (MBRs)* that enclose plane regions (the actual spatial objects). On the one hand, approximating regions by rectangles implies a loss of accuracy in the representation of the relative direction between regions; on the other hand, reasoning tasks become more efficient. The RCD-calculus has a strong connection with the *Rectangle Algebra* (RA) [2], which can be viewed as a bidimensional extension of *Interval Algebra* (IA), the well-known temporal formalism for dealing with qualitative binary relations between time intervals [1]. A tractable fragment of the RCD-calculus, called *convex RCD-calculus*, has been identified in [18]. It includes all basic relations and a large number of disjunctive relations, making it possible to represent and reason about indefinite information efficiently.

In this paper, we extend the convex RCD-calculus with metric constraints. Metric constraints between points over a dense linear order have been dealt with by the Temporal Constraint Satisfaction Problem formalism (TCSP) [7]. In such a formalism, one can restrict the admissible values for the distance between a pair of points to a finite set of ranges. If each constraint consists of one range only, we get a tractable fragment of TCSP, called Simple Temporal Problem formalism (STP). In the following, we propose a metric extension to the convex RCD-calculus that allows one to represent available knowledge on directional relations between rectangles and to derive additional information about them, as well as to deal with metric constraints on the height/width of a rectangle or on the vertical/horizontal distance between rectangles. We will show that the resulting formalism is expressive enough to capture various scenarios of practical interest and still computationally affordable.

The rest of the paper is organized as follows. In Section 2, we provide background knowledge on qualitative calculi and we shortly recall Interval Algebra and Rectangle Algebra. In Section 3, we introduce the RCD-calculus and its convex fragment. In Section 4, we extend the convex RCD-calculus with metric features, and we devise a sound and complete polynomial algorithm for consistency checking. In Section 5, we apply the proposed formalism to a case study in the domain of web-document layout design. Conclusions provide an assessment of the work and outline future research directions.

2 Background

In this section, we briefly review basic notions on constraint networks and the main calculi regarding qualitative relations on points, intervals and rectangles.

Temporal (resp., spatial) knowledge is commonly represented in a qualitative calculus by means of a *qualitative network* consisting of a complete constraint-labeled digraph $N = (V, C)$, where $V = \{v_1, \ldots, v_n\}$ is a finite set of variables, interpreted over an infinite domain D, and the labeled edges in C specify the constraints defining qualitative temporal (resp., spatial) configurations. An edge from v_i to v_j labeled with R corresponds to the *constraint* $v_i\, R\, v_j$, where R denotes a binary relation over D which restricts the possible values for the pair of variables (v_i, v_j). The full set of relations of the calculus is usually taken as the powerset $2^{\mathcal{B}}$, where $\mathcal{B}$ is a finite set of binary *basic relations* that forms a partition of $D \times D$. Thus, a relation $R \in 2^{\mathcal{B}}$ is of the form $R = \{r_1, \ldots, r_m\}$, where each r_i is a basic relation, and R represents the union of the basic relations it contains. If $m = 1$, we call R a *basic relation*; otherwise ($m > 1$), we call it a *disjunctive relation*. A special case of disjunctive relation is the *universal relation*, denoted by '?', which contains all the basic relations. A *basic constraint* $v_i\{r\}v_j$ expresses definite knowledge about the values that the two variables v_i, v_j can take, while a *disjunctive constraint* $v_i\{r_1, \ldots, r_m\}v_j$ expresses indefinite or imprecise knowledge about these values. In particular, the *universal constraint* $v_i\, ?\, v_j$ states that the relation between v_i an v_j is totally unknown. From a logical point of view, a disjunctive constraint $v_i\{r_1, \ldots, r_m\}v_j$ can be viewed as the disjunction $v_i\ \{r_1\}\ v_j \vee \ldots \vee v_i\ \{r_m\}\ v_j$.

An *instantiation* of the constraints of a qualitative network N is a mapping ι representing an assignment of domain values to the variables of N. A constraint $v_i\, R\, v_j$ is said to be *satisfied* by an instantiation ι if the pair $(\iota(v_i), \iota(v_j))$ belongs to the binary relation represented by R. A *consistent instantiation*, or *solution*, of a network is an assignment of domain values to variables satisfying all the constraints. If such a solution exists, then the network is consistent, otherwise it is inconsistent.

The main reasoning task in qualitative reasoning is *consistency checking*, which amounts to deciding if a network is consistent. If all relations are considered, consistency checking is usually NP-hard. Hence, finding subsets of $2^{\mathcal{B}}$ for which consistency checking turns out to be polynomial (*tractable subsets*) is an important issue to address. Another common task in qualitative reasoning is computing the unique *minimal network* equivalent to a given one by determining, for each pair of variables, the *strongest* relation *(minimal relation*) entailed by the constraints of the network. It can be easily shown that each basic relation in a minimal network is *feasible*, i.e., it participates in some solution of the network. To deal with these tasks, constraint propagation techniques are usually exploited [25,23]. The most prominent method for constraint propagation in qualitative temporal reasoning is the so-called *path-consistency algorithm*, PC-algorithm for short [15]. Such an algorithm refines relations by successively applying the operation $R_{ij} \leftarrow R_{ij} \cap (R_{ik} \circ R_{kj})$ for every triple of variables (v_i, v_k, v_j), until a stable network is reached, where R_{ij}, R_{ik}, R_{kj} are the relations constraining the pair of variables (v_i, v_j), (v_i, v_k), (v_k, v_j), respectively ($\circ$ stands for the composition of relations). If the empty relation is obtained during the process, then the input network is inconsistent; otherwise, we can conclude that the output network is *path consistent*,

Relation	Symbol	Inverse	Illustration	Meaning
I before J	b	bi		$I^- < I^+ < J^- < J^+$
I overlaps J	o	oi		$I^- < J^- < I^+ < J^+$
I during J	d	di		$J^- < I^- < I^+ < J^+$
I meets J	m	mi		$I^- < I^+ = J^- < J^+$
I starts J	s	si		$I^- = J^- < I^+ < J^+$
I finishes J	f	fi		$J^- < I^- < I^+ = J^+$
I equals J	e	e		$I^- = J^- < I^+ = J^+$

Fig. 1. Basic relations of the Interval Algebra

which does not necessarily imply that it is consistent. In some special cases, the PC-algorithm can be used to decide the consistency of a qualitative network and to get the minimal one.

2.1 Interval Algebra and Point Algebra

Allen's *Interval Algebra* (IA) allows one to constrain the relative position of two time intervals [1]. An interval I is usually interpreted as a closed interval over the rational numbers $[I^-, I^+]$, whose *endpoints* I^- and I^+ satisfy the relation $I^- < I^+$. Let $\mathcal{B}_{ia}$ be the set of the thirteen *basic interval relations* capturing all possible ways to order the four endpoints of two intervals, usually denoted by the symbols $b, o, d, m, s, f, e, bi, oi$, di, mi, si, and fi. The semantics of basic IA-relations is defined in terms of ordering relations between the endpoints of the intervals, as shown in Figure 1. Notice that, given a basic relation r between two intervals I and J, the inverse relation ri is defined by simply exchanging the roles of I and J (see Figure 1). IA can be viewed as a constraint algebra defined by the power set $2^{\mathcal{B}_{ia}}$ and the operations of intersection, inverse ($^{-1}$), and composition ($\circ$) of relations.

IA subsumes *Point Algebra* (PA) [25], a simpler qualitative calculus whose binary relations specify the relative position of pairs of time points. PA binary relations are $<, >, =$ (basic) and $\leq, \geq, \neq, ?$ (disjunctive), plus the empty relation. The endpoint relations defining a basic IA-relation (Figure 1) are basic relations of PA.

2.2 Rectangle Algebra

Rectangle Algebra (RA), proposed by Balbiani et al. (1998), is an extension of IA to a bidimensional space. We assume here the domain of RA to consist of the set of rational rectangles whose sides are parallel to the axes of the Euclidean plane. To avoid a notational overload, with a little abuse of notation, hereafter we will denote by a, b both rectangles in the domain of RA and constraint (rectangle) variables. A rectangle a is completely characterized by a pair of intervals (a_x, a_y), where a_x and a_y are the projections of a onto the x- and y-axis, respectively. We call $\mathcal{B}_{ra}$ the set of basic relations of RA, which is obtained by considering all possible pairs of basic IA-relations.

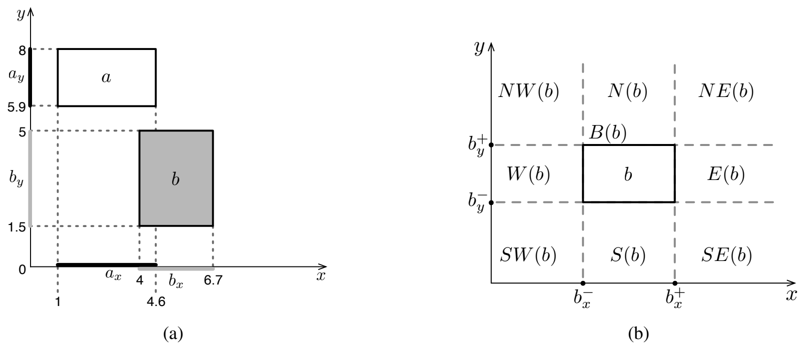

Fig. 2. (a) An illustration of the RA-relation $a\{(o, bi)\}b$. The corresponding RCD-relation is $a\,\{NW{:}N\}\,b$. (b) Cardinal tiles with respect to rectangle b.

Hence, a basic RA-relation r is denoted by a pair $r = (t, t')$ of basic IA-relations, representing the set of pairs of *rectangles* (a, b) such that $a\,(t, t')\,b$ holds if and only if, by definition, $a_x\,t\,b_x$ and $a_y\,t'\,b_y$ hold. Given a basic RA-relation $r = (t, t')$, let $t = \pi_x(r)$ and $t' = \pi_y(r)$ be the x- and y-projection of r, respectively.

Example 1. Figure 2-(a) shows a spatial realization of $a\,\{(o, bi)\}\,b$. Note that $\pi_x(o, bi) = o$, $\pi_y(o, bi) = bi$, a_x overlaps b_x, and a_y is after b_y. The left endpoints of the intervals assigned to a_x and a_y (1 and 5.9, respectively) and their right endpoints (4.6 and 8, respectively) are the coordinates of the lower-left and upper-right vertices of the given instantiation of a, respectively. The same for b. Thus, the values assigned to the endpoints of the projections of a and b represent an assignment for a and b that satisfies the constraint $a\,\{(o, bi)\}\,b$.

In the case of an arbitrary RA-relation $R \in 2^{\mathcal{B}_{ra}}$, the *projections of* R are defined as follows:

$$\pi_x(R) = \{\pi_x(r) \mid r \in R\} \quad \pi_y(R) = \{\pi_y(r) \mid r \in R\}.$$

Notice that, in general, $\pi_x(R) \times \pi_y(R)$ may be different from R or, equivalently, we may have $\pi_x(R_1) = \pi_x(R_2)$ and $\pi_y(R_1) = \pi_y(R_2)$ for some $R_1 \neq R_2$.

The mappings π_x and π_y can be generalized to RA-networks. We define the projections π_x and π_y of an RA-network $N = (V, C)$ as the two IA-networks $\pi_x(N) = (V_x, C_x)$ and $\pi_y(N) = (V_y, C_y)$, where V_x, V_y are the sets of interval variables corresponding to the rectangle variables in V and the set of IA-constraints C_x (resp., C_y) is obtained by replacing each relation R_{ij} in C by $\pi_x(R_{ij})$ (resp., by $\pi_y(R_{ij})$).

2.3 Convex Subalgebras

The consistency problem for both IA and RA is known to be NP-complete. Several tractable fragments of both calculi have been identified in the literature. In this paper, we focus our attention on convex tractable subsets of IA [24] and RA [2], which consist of the set of *convex IA-relations* and *convex RA-relations*, respectively. Convex relations are those relations that can be equivalently expressed as a set of convex PA-constraints (all PA-relations except $\neq$ are allowed) between the endpoints of interval

Table 1. Translation of convex PA-constraints to STP-constraints via the $toSTP$ mapping

Convex PA relation	STP constraint
$p_i < p_j$	$p_j - p_i \in]0, +\infty[$
$p_i \leq p_j$	$p_j - p_i \in [0, +\infty[$
$p_i = p_j$	$p_j - p_i \in [0, 0]$
$p_i > p_j$	$p_j - p_i \in]-\infty, 0[$
$p_i \geq p_j$	$p_j - p_i \in]-\infty, 0]$
$p_i \, ? \, p_j$	$p_j - p_i \in]-\infty, +\infty[$

variables (convex IA-relations) or between the endpoints of the projections of rectangle variables (convex RA-relations). It is worth to mention that a convex RA-relation is equivalently characterized as an RA-relation which can be obtained as the Cartesian product of two convex IA-relations. Both the consistency and the minimality problems in the convex fragments of PA, IA, and RA can be solved in $O(n^3)$, where n is the number of variables of the input network.

2.4 Simple Temporal Problem

The Simple Temporal Problem (STP) formalism was introduced in [7] to process metric information about time points. Formally, an STP is specified by a constraint network $S = (P, M)$, where P is a set of n point variables, whose values range over a dense domain (which we assume to be $\mathbb{Q}$), and M is a set of *binary metric* constraints over P. A metric constraint $M_{ij} = [q, q']$ (open and semi-open intervals can be used as well), with $q, q' \in \mathbb{Q}$, on the distance between (the values of) $p_i, p_j \in P$ states that $p_j - p_i \in [q, q']$, or, equivalently, that $q \leq p_j - p_i \leq q'$. Hence, the constraint M_{ij} defines the set of possible values for the distance $p_j - p_i$. In the constraint graph associated to S, $M_{ij} = [q, q']$ is represented by an edge from p_i to p_j labeled by the rational interval $[q, q']$. *Unary metric* constraints restricting the domain of a point variable p_i can be encoded as binary constraints between p_i and a special starting-point variable with a fixed value, e.g., 0. The *universal constraint* is $] - \infty, +\infty[$. The operations of composition ($\circ$) and inverse ($^{-1}$) of metric constraints are computed by means of interval arithmetic, that is, $[q_1, q_2] \circ [q_3, q_4] = [q_1 + q_3, q_2 + q_4]$ and $[q_1, q_2]^{-1} = [-q_2, -q_1]$. Intersection of constraints (intervals) is defined as usual.

Assuming such an interpretation of the operations of composition, inverse, and intersection, in [7] Dechter et al. show that any path-consistency algorithm can be exploited to compute the minimal STP equivalent to a given one, if any (if an inconsistency is detected, the algorithm returns an empty network). In the following, we will denote such an algorithm by $\mathbf{PC_{stp}}$. Making use of such a result, in [16], Meiri proposes a formalism to combine qualitative constraints between points and intervals with (possibly disjunctive) metric constraints between points (as in TCSP). An easy special case arises when only convex PA-constraints and STP-constraints are considered. Convex PA-constraints can be encoded as STP-constraints by means of the $toSTP$ translation function described in Table 1. The following result can be found in [16].

Theorem 1. *Let N be a network with convex PA-constraints and STP-constraints. If N is path-consistent, then N is also consistent and its metric constraints are minimal.*

$\mathbf{PC_{stp}}$ can thus be used to decide the consistency of a network N satisfying the conditions of the above theorem. To this end, it suffices to encode PA-constraints into equivalent STP-constraints.

3 Rectangular Cardinal Direction Calculus

The Rectangular Cardinal Direction calculus (*RCD-calculus*) [19,18] deals with cardinal direction relations between rectangles. It can be viewed as a restricted version of the CD-calculus over the domain of regular regions [10,14,22], which includes all rectangles aligned to the axes. The domain of the CD-calculus is the same as that of RA. Let b be a *reference* rectangle. We denote by b_x^- and b_x^+ (resp., b_y^- and b_y^+) the left and the right endpoint of the projection of b onto the x-axis (resp., y-axis), respectively. The straight lines $x = b_x^-,\ x = b_x^+,\ y = b_y^-,\ y = b_y^+$ divide the plane into nine tiles $\tau_i(b)$, with $1 \leq i \leq 9$, as shown in Figure 2-(b), where τ_i is a *tile symbol* from the set $TS = \{B, S, SW, W, NW, N, NE, E, SE\}$, denoting the cardinal directions in the Bounds of, to the South of, to the SouthWest of, to the West of, to the NorthWest of, to the North of, to the NorthEast of, to the East of, and to the SouthEast of, respectively.

Definition 1. *A* basic rectangular cardinal relation *(basic RCD-relation) is denoted by a* tile string $\tau_1{:}\tau_2{:}\ldots{:}\tau_k$*, where $\tau_i \in TS$, for $1 \leq i \leq k$, such that $a\ \tau_1{:}\tau_2{:}\ldots{:}\tau_k\ b$ holds iff for all $\tau_i \in \{\tau_1, \tau_2, \ldots, \tau_k\}$, $a^\circ \cap \tau_i(b) \neq \varnothing$, and for all $\tau_i \in TS \setminus \{\tau_1, \tau_2, \ldots, \tau_k\}$, $a^\circ \cap \tau_i(b) = \varnothing$, where a° is the interior of a. A* rectangular cardinal relation *(RCD-relation) is represented by a set $R = \{r_1, \ldots, r_m\}$, where each r_i is a basic RCD-relation.*

The set $\mathcal{B}_{rcd}$ of basic RCD-relations consists of 36 elements (see Figure 3). Qualitative networks with labels in $2^{\mathcal{B}_{rcd}}$, as well as the consistency problem for such networks, are defined in the standard way.

3.1 RCD-Calculus and Rectangle Algebra

The relationships between the RCD-calculus and the Rectangle Algebra have been systematically investigated by Navarrete et al. in [18]. Consider, for instance, the RCD-constraint $a\,\{NW{:}N\}\,b$. A possible instantiation of such a constraint is depicted in Figure 2-(a). The same pair of rectangles can be viewed as an instance of the RA-constraint $a\,\{(o, bi)\}\,b$ as well. However, there exists another possible instantiation of the constraint $a\,\{NW{:}N\}\,b$ that satisfies the RA-constraint $a\,\{(o, mi)\}\,b$. In general, for a given RCD-relation there exist more than one corresponding RA-relations, while for a given RA-relation there exists exactly one corresponding RCD-relation. This is due to the fact that RCD-relations are coarser than RA-relations. As an example, the RCD-calculus does not allow one to precisely state that two given rectangles are externally connected or strictly disconnected, or to constrain their sides to be (or to be not) vertically (resp., horizontally) aligned. As a general rule, given an RCD-relation, we

Basic RCD-relation ↦ RA-relation (I)	Basic RCD-relation ↦ RA-relation (II)
B ↦ {d, s, f, e} × {d, s, f, e}	W:NW ↦ {m, b} × {si, oi}
S ↦ {d, s, f, e} × {m, b}	E:SE ↦ {mi, bi} × {fi, o}
N ↦ {d, s, f, e} × {mi, bi}	NE:E ↦ {mi, bi} × {si, oi}
E ↦ {mi, bi} × {d, s, f, e}	S:SW:SE ↦ {di} × {m, b}
W ↦ {m, b} × {d, s, f, e}	NW:N:NE ↦ {di} × {mi, bi}
NE ↦ {mi, bi} × {mi, bi}	B:W:E ↦ {di} × {d, s, f, e}
NW ↦ {m, b} × {mi, bi}	B:S:N ↦ {d, s, f, e} × {di}
SE ↦ {mi, bi} × {m, b}	SW:N:NW ↦ {m, b} × {di}
SW ↦ {m, b} × {m, b}	NE:E:SE ↦ {mi, bi} × {di}
S:SW ↦ {fi, o} × {m, b}	B:S:SW:W ↦ {o, fi} × {o, fi}
S:SE ↦ {si, oi} × {m, b}	B:W:NW:N ↦ {o, fi} × {si, oi}
NW:N ↦ {fi, o} × {mi, bi}	B:S:E:SE ↦ {si, oi} × {o, fi}
N:NE ↦ {si, oi} × {mi, bi}	B:N:NE:E ↦ {si, oi} × {si, oi}
B:W ↦ {fi, o} × {d, s, f, e}	B:S:SW:W:NW:N ↦ {o, fi} × {di}
B:E ↦ {si, oi} × {d, s, f, e}	B:S:N:NE:E:SE ↦ {si, oi} × {di}
B:S ↦ {d, s, f, e} × {fi, o}	B:S:SW:W:E:SE ↦ {di} × {fi, o}
B:N ↦ {d, s, f, e} × {si, oi}	B:W:NW:N:NE:E ↦ {di} × {si, oi}
W:SW ↦ {m, b} × {fi, o}	B:S:SW:W:NW:N:NE:E:SE ↦ {di} × {di}

Fig. 3. Translation from basic RCD-relations to RA-relations via the $toRA$ mapping

can always determine the strongest RA-relation it implies. As an example, the strongest RA-relation implied by $NW{:}N$ is $\{fi, o\} \times \{mi, bi\}$. Notice that such an RA-relation, which is entailed by a basic RCD-relation, is not a basic RA-relation.

The weaker expressive power of the RCD-calculus with respect to RA is not necessarily a problem. As an example, if we are interested in pure cardinal information only, the expressiveness of RCD-relations suffices. Moreover, the constraint language of the RCD-calculus is closer to the natural language than the one of the RA. For example, stating that "rectangle a lies partly to the northwest and partly to the north of b" ($a\,\{NW{:}N\}\,b$) is much more natural than stating that "the x-projection of a is overlapping or finished by the x-projection of b, and the y-projection of b is …" ($a\{fi, o\} \times \{mi, bi\}b$).

Figure 3 describes a translation function, called $toRA$, to map a basic RCD-relation into the strongest entailed RA-relation. This mapping can be extended to translate arbitrary relations, constraints, and networks of the RCD-calculus to their RA counterparts, preserving consistency. More precisely, given a disjunctive relation R, $toRA(R)$ is obtained as the union of the translation of the basic relations in R, while, given an RCD-network $N = (V, C)$, the corresponding RA-network $toRA(N)$ is obtained by replacing each relation R_{ij} in C by $toRA(R_{ij})$. As the following theorem states, to decide the consistency of an RCD-network N, one can compute $toRA(N)$ and then apply to it any algorithm for deciding the consistency of RA-networks [18].

Theorem 2. *An RCD-network N is consistent if and only if $toRA(N)$ is consistent.*

3.2 The Convex Fragment of the RCD-Calculus

In [18], the authors show that the consistency problem for the RCD-calculus is NP-complete and they identify a large tractable subset of RCD-relations. Such a fragment, called *convex RCD-calculus*, consists of all and only the RCD-relations R whose translation $toRA(R)$ is a convex RA-relation. It is possible to show that there exist 400 *convex RCD-relations*. As we already pointed out, the convex subclasses of IA, PA, and RA are tractable. By exploiting the connection between these subclasses and the convex RCD-calculus, an $O(n^2)$ algorithm for consistency checking of convex RCD-networks has been proposed in [18]. In particular, it benefits from the following result about RA, stated in [2].

Theorem 3. *Let N be a convex RA-network. N is consistent if and only if its projections $\pi_x(N)$ and $\pi_y(N)$ are consistent.*

4 Convex-Metric RCD-Calculus

In this section, we propose a tractable metric extension of the convex RCD-calculus, called *convex-metric RCD-calculus* (*cmRCD-calculus*) to represent and to reason about both qualitative cardinal constraints between rectangles and metric constraints on the distance between the endpoints of their projections. The main tool we use to deal with metric information in the cmRCD-calculus is STP. More precisely, we use STP to elaborate information on the endpoints of MBR projections onto the Cartesian axes.

Integrating the convex RCD-calculus with STP makes it possible to express both directional constraints and metric constraints in a uniform framework. As an example, the resulting formalism allows one to constrain the position of a rectangle in the plane and to impose minimum and/or maximum values to the width/height of a given rectangle, or on the vertical/horizontal distances between the sides of two rectangles. Obviously, RCD-constraints and STP-constraints are not totally independent, that is, RCD-constraints entail some metric constraints and vice versa.

Example 2. Let a and b be two rectangles. We can use the metric constraint $0 \leq a_x^+ - a_x^- \leq 7$ to state that the maximum width of a is 7 and, similarly, we can exploit the metric constraint $2 \leq a_y^+ - a_y^-$ to state that the minimum height of a is 2 (leaving the maximum height unbounded). We can also express distance constraints between the boundaries of a and b. We can constrain the horizontal distance between the right side of a and the left side of b to be at least 3 by means of the constraint $3 \leq b_x^- - a_x^+$, and the vertical distance between the upper side of a and the bottom side of b to be greater than or equal to 0 by means of the constraint $0 \leq b_y^- - a_y^+$. The two constraints together entail the basic RCD constraint $a\,\{SW\}b$. Finally, some metric constraints can be entailed by RCD ones. For instance, the convex relation $a\,\{NW, N, NE, NW{:}N, NW{:}N{:}NE, N{:}NE\}\,b$ implies that $0 \leq a_y^- - b_y^+$.

If we allow one to combine arbitrary RCD-constraints with metric constraints, then checking the consistency of the resulting set of constraints turns out to be an NP-complete problem (the consistency problem for RCD-networks is already NP-complete). To preserve tractability, the cmRCD-calculus combines convex RCD-constraints with STP-constraints. Constraint networks in the cmRCD-calculus (*cmRCD-networks*) are defined as follows. Given a convex RCD-network $N_c = (V, C)$, we denote the sets of interval variables belonging to the projections $\pi_x(toRA(N_c))$ and $\pi_y(toRA(N_c))$ by V_x and V_y, respectively. Moreover, we denote by $P(V_x)$ and $P(V_y)$ the sets of point variables representing the endpoints of the interval variables in V_x and V_y, respectively.

Definition 2. *A* cmRCD-network *is an integrated qualitative and metric constraint network N consisting of three sub-networks $(N_c, \mathcal{S}_x, \mathcal{S}_y)$, where $N_c = (V, C)$ is a convex RCD-network, and $\mathcal{S}_x = \big(P(V_x), M_x\big)$ and $\mathcal{S}_y = \big(P(V_y), M_y\big)$ are two STPs.*

Algorithm 4.1. The algorithm **con-cmRCD**

Require: a cmRCD-network $N = (N_c, \mathcal{S}_x, \mathcal{S}_y)$
1: $N_r \leftarrow toRA(N_c)$;
2: $N_x \leftarrow \pi_x(N_r)$, $N_y \leftarrow \pi_y(N_r)$;
3: $N_x^P \leftarrow toPA(N_x)$, $N_y^P \leftarrow toPA(N_y)$;
4: $xSTP \leftarrow intersect(toSTP(N_x^P), \mathcal{S}_x)$;
5: $ySTP \leftarrow intersect(toSTP(N_y^P), \mathcal{S}_y)$;
6: if $xSTP$ or $ySTP$ is empty, then return 'inconsistent';
7: $xSTP^{min} \leftarrow PC_{stp}(xSTP)$;
8: $ySTP^{min} \leftarrow PC_{stp}(ySTP)$;
9: If $xSTP^{min}$ or $ySTP^{min}$ is empty, then return 'inconsistent'; otherwise, return 'consistent'.

The cmRCD-calculus subsumes both the convex RCD-calculus and the STP formalism. Moreover, it generalizes the convex fragment of the RA, since convex RA-relations are expressible as convex PA-relations, which can be encoded into an STP.

In the following, we describe an algorithm, called **con-cmRCD**, to solve the consistency problem for cmRCD-networks, that runs in $O(n^3)$. As a matter of fact, a similar combination of qualitative and quantitative networks is provided by preconvex-augmented rectangle networks. An $O(n^5)$ algorithm for checking the consistency of these networks, that subsume *cmRCD-networks*, is given in [6]. We exploit the trade-off between expressiveness and complexity to obtain a more efficient consistency checking algorithm for *cmRCD-networks*.

As a preliminary step, we extend the translation mapping $toSTP$ of Table 1 to encode a convex PA-network N^P into an STP S by replacing each relation R in the network N^P by $toSTP(R)$. First, **con-cmRCD** applies the mapping $toRA$ to the input convex RCD-network N_c to get the corresponding convex RA-network N_r. Then, it computes the projections N_x and N_y of N_r. Next, it applies the mapping $toPA$ to translate the convex IA-networks N_x and N_y into two equivalent PA-networks N_x^P and N_y^P with convex relations between points variables representing the projections of the intervals in N_x and N_y. Thereafter, making use of such an encoding of convex RCD-relations as PA-relations, it looks for possible inconsistencies between these constraints and the STP-constraints on the same variables given in S_x and S_y that can be detected at this stage. To this end, it translates the PA-network N_x^P (resp., N_y^P) into an STP-network by applying the extended function $toSTP$, and then it uses the function $intersect$ to compute the "intersection" between $toSTP(N_x^P)$ and S_x (resp., $toSTP(N_y^P)$ and S_y). This function simply intersects the constraints associated with the same pairs of variables in the two STPs. If an interval intersection produces an empty interval, then $intersect$ returns an empty network, and we can conclude that N is inconsistent. Otherwise, we apply the path-consistency algorithm to the two STPs computed at lines 4 and 5 independently. The following theorem proves that **con-cmRCD** is sound and complete.

Theorem 4. *Given a cmRCD-network $N = (N_c, \mathcal{S}_x, \mathcal{S}_y)$, the algorithm* $\mathbf{con\text{-}cmRCD}$ *returns 'consistent' if and only if N is consistent.*

Proof. We basically follow the steps of the algorithm. By Theorem 2, N_c is consistent if and only if N_r is consistent, and, by Theorem 3, N_r is consistent if and only if N_x and N_y are consistent (they can be checked independently). Next, N_x and N_y are consistent if and only if N_x^P and N_y^P are consistent, since there is no loss in information in the translations [24]. The consistency of N_x^P and N_y^P can be checked by computing the corresponding STPs and by applying PC_{stp}. However, we cannot apply PC_{stp} directly to the STPs $toSTP(N_x^P)$ and $toSTP(N_x^P)$ since the metric constraints of S_x and S_y must be taken into account. Hence, we compute $xSTP = intersect(toSTP(N_x^P), S_x)$ and $ySTP = intersect(toSTP(N_x^P), S_y)$. If one of them returns an empty network, then N is inconsistent. Otherwise, we independently apply PC_{stp} to $xSTP$ and $ySTP$. By Theorem 1, if one of the two applications of PC_{stp} returns an empty network, then N is inconsistent; otherwise, the path-consistent STPs $xSTP^{min}$ and $ySTP^{min}$ are consistent (and minimal), and thus N is consistent. □

Theorem 5. *The complexity of the algorithm* $\mathbf{con\text{-}cmRCD}$ *is $O(n^3)$, where n is the number of variables of the input network.*

Proof. The translation via $toRA$, the generation of a projection of a network, the transformation of an IA-network into an RA-network via $toPA$, and the last two encodings via $toSTP$ require $O(n^2)$ steps, since there are $O(n^2)$ constraints and each constraint can be translated in constant time. The function $toPA$ introduces two variables for each interval variable, so $xSTP$ and $ySTP$ have $O(n)$ variables each. Finally, PC_{stp} runs in $O(n^3)$ time, so the overall complexity is $O(n^3)$ time (for further details about the complexity of achieving path-consistency for combined networks see [16]). □

Once we have computed the path-consistent STPs $xSTP^{min}$ and $ySTP^{min}$ with algorithm $\mathbf{con\text{-}cmRCD}$, we can build a *solution* to the cmRCD-network N by computing a solution for the points in $xSTP^{min}$ and $ySTP^{min}$, since the assignment for point variables defines a consistent assignment for rectangle variables (see Example 1). To this end, the $O(n^3)$ algorithm STP-SOLUTION, by Gerevini and Cristani [9], can be used.

5 A Case Study for the cmRCD-Calculus

Given its distinctive features, the cmRCD-calculus is well-suited for all application areas where minimal bounding rectangles can be successfully exploited. This is the case, for instance, with spatial databases [10,21], information extraction from formatted documents [8,20], and 2D-layout design [3,4,17]. We would also like to mention the application of convex RCD-calculus (the qualitative fragment of cmRCD-calculus) to the problem querying and extracting data from web documents reported in [20]. We believe that, in view of its well-balanced combination of efficiency and expressiveness, cmRCD-calculus can be naturally and successfully applied to this class of problems as well.

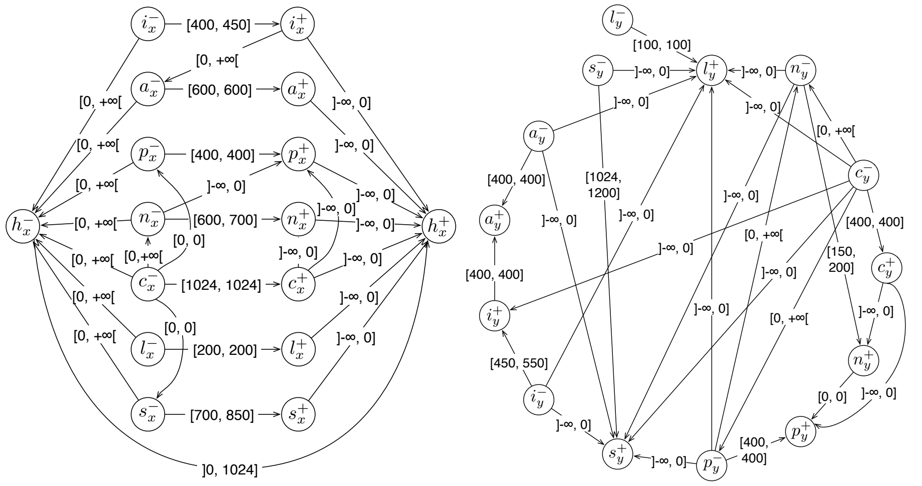

Fig. 4. Graph representation of part of the $xSTP$ and $ySTP$ of Example 3. For the sake of readability, constraints involving h in the $ySTP$ and the universal constraint are omitted.

In this section, we focus our attention on an application of the cmRCD-calculus to the automatic web page layout generation, inspired by a work by Borning et al. [4]. On the one hand, current web authoring tools do not allow home page designers to specify how the document should change in response to viewer's needs; on the other hand, web browsers do not really allow their users to express their requirements about the layout, except for those about the dimension of the font and few other features. The work by Borning et al. aims at allowing both the designer and the viewer to specify the positioning of the document elements by means of linear equalities and inequalities over their minimum bounding boxes, in such a way that the layout of the web page becomes the result of a negotiation between them (designer and viewer). In the following, we show how to apply cmRCD-calculus to allow the user (author or viewer) to specify both cardinal and metric constraints on the layout elements. Notice that, in doing that, we reduce the expressive power of Borning et al.'s proposal; nevertheless, the problem they consider is in fact an optimization problem, which is solved by means of a linear-programming algorithm that has an exponential worst-case time complexity.

Example 3. Let us consider a Facebook-like social network, that allows the user to personalize the contents of his/her home page by making use of directional constraints. We can assume each element the user can add to be represented by a MBR, or box, whose sides are parallel to the axes of the reference system centered at the lower left vertex of the home page. As an example, we may have a box containing information about his/her (gender, birthday, etc.), a box containing his/her profile picture, and so on. In addition, it makes sense to assume that the system requires all user pages to share some common presentation features like, for instance, the system logo and some general presentation directives.

In this scenario, we can imagine that a user enters the following requirements to some design tool that interacts with the layout designer (user's specifications are given in terms of directions such as “right” or “inside”, which are far more natural for a local space than their equivalent cardinal directions “to the East of” or “in the bounds of”):

1. my cover picture c has to lie on the top of the home page h, that is, the vertical distance between the top sides of c and h must be 0, and the dimension of c is 1024×400 px;
2. the box n containing my full name has to lie inside c;
3. the box i containing my personal information has to be somewhere below c, no matter what its horizontal position is (in terms of cardinal relations, this requirement should be understood as “somewhere between the south west and the south east zone of c”);
4. the box a containing the cover pictures of my photo albums has to lie to the right of i, no matter what its vertical position is; in addition, the vertical distance between the top sides of a and i must be 0;
5. the box p containing my profile picture has to lie inside c, and the horizontal distance between the left sides of of p and c must be 0; in addition, n has to be to the right of p, with the restriction that the vertical distance between the top sides of p and n must be 0;
6. I want to see the 5 most recent stories from my friends in a box s, which has to lie below the previous elements, no matter what its horizontal position is, and the horizontal distance between the left sides of s and c must be 0.

Besides these user constraints, we can imagine that the system imposes the following additional constraints:

7. somewhere at the bottom of the home page, there must be a logo l, whose dimension is 200×100 px;
8. the width of the home page cannot exceed 1024 px.

When the contents of all boxes are retrieved from the database server, the system provides lower and upper bounds to the size of the boxes, so that the layout manager has more chances to fit the contents on the basis of user preferences. In particular, we can assume that the following conditions hold:

9. n’s width can vary from 600 to 700 px, while n’s height can vary from 150 to 200 px;
10. i’s width can vary from 400 to 450 px, while i’s height can vary from 450 to 550 px;
11. s’s width can vary from 700 to 850 px, while s’s height can vary from 1024 to 1200 px;
12. a is 600×400 px;
13. p is 400×400 px.

A particular choice for the size of a box is automatically compensated by increasing/decreasing the font size.

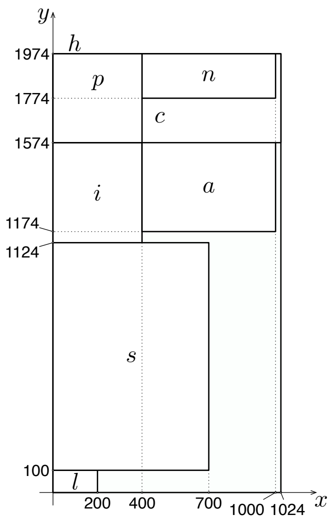

Fig. 5. A solution to the cmRCD-network for Example 3.

When the system has to deliver the web page, it must find a solution to a *cmRCD-network* consisting of the following qualitative constraints (that encode the above qualitative requirements):

1. Implicit: "boxes must be inside the homepage":
   $c\,B\,h, \quad n\,B\,h, \quad i\,B\,h, \quad a\,B\,h, \quad l\,B\,h, \quad p\,B\,h, \quad s\,B\,h$
2. $n\,B\,c$;
3. $i\,\{SW, S, SW{:}S, SW{:}S{:}SE, S{:}SE, SE\}\,c$;
4. $a\,\{NE, E, NE{:}E, NE{:}E{:}SE, E{:}SE, SE\}\,i$;
5. $p\,B\,c,\ n\,E\,p$;
6. for each box $b \in \{c, n, i, a, p\}$, $s\,\{SW, S, SW{:}S, SW{:}S{:}SE, S{:}SE, SE\}\,b$;
7. for each box b, $l\,\{SW, S, SW{:}S, SW{:}S{:}SE, S{:}SE, SE\}\,b$;

and of the following metric constraints (that encode the above metric requirements):

1. $h_y^+ - c_y^+ = 0,\ c_y^+ - c_y^- = 400,\ c_x^+ - c_x^- = 1024$;
2. $a_y^+ - i_y^+ = 0$;
3. $p_y^+ - n_y^+ = 0,\ p_x^- - c_x^- = 0$;
4. $s_x^- - c_x^- = 0$;
5. $l_x^+ - l_x^- = 200,\ l_y^+ - l_y^- = 100$;
6. $0 < h_x^+ - h_x^- \leq 1024$;
7. $600 \leq n_x^+ - n_x^- \leq 700,\ 150 \leq n_y^+ - n_y^- \leq 200$;
8. $400 \leq i_x^+ - i_x^- \leq 450,\ 450 \leq i_y^+ - i_y^- \leq 550$;
9. $700 \leq s_x^+ - s_x^- \leq 850,\ 1024 \leq s_y^+ - s_y^- \leq 1200$;
10. $a_x^+ - a_x^- = 600,\ a_y^+ - a_y^- = 400$;
11. $p_x^+ - p_x^- = 400,\ p_y^+ - p_y^- = 400$.

Meaningful portions of the constraint networks $xSTP$ and $ySTP$, generated by steps 4 and 5 of the algorithm **con-cmRCD**, respectively, are depicted in Figure 4. A possible solution to the network, that is, a possible layout of the user home page, is given in Figure 5. It is worth pointing out that the picture shows the minimum feasible values for point variables, and thus it immediately follows that the minimum dimension of h is 1024×1974 px.

6 Conclusions

In this paper, we have proposed a quite expressive, but tractable, metric extension of RCD-calculus, that integrates convex RCD-constraints and STP-constraints. The resulting cmRCD-calculus allows one to constrain the position of a rectangle in the plane, its width/height, and the vertical/horizontal distance between the sides of two rectangles, as well as to represent cardinal relations between rectangles. We have developed an $O(n^3)$ consistency-checking algorithm for such a calculus, and we have shown how a spatial realization of a *cmRCD-network* can be built.

As for future work, we plan to extend the cmRCD-calculus with topological relations to improve its expressiveness (similar results can be found in [11,13]). Moreover, since the problem of identifying maximal tractable subsets of RCD is still open, it makes sense to search for tractable classes strictly including the convex fragment. Finally, we are interested in developing heuristics and algorithms to check consistency and to find a solution in the cases of non-convex RCD-relations or disjunctive metric constraints. Since in these cases both problems turn out to be intractable, an empirical evaluation of the solutions is necessary to check scalability.

Acknowledgements. This work has been partially supported by the Spanish Ministry of Science and Innovation, the European Regional Development Fund of the European Commission under grant TIN2009-14372-C03-01, and the Spanish MEC through the project 15277/PI/10, funded by Seneca Agency of Science and Technology of the Region of Murcia within the II PCTRM 2007-2010. Guido Sciavicco and Angelo Montanari were also partially founded by the Spanish fellowship 'Ramon y Cajal' RYC-2011-07821 and by the Italian PRIN project *Innovative and multi-disciplinary approaches for constraint and preference reasoning*, respectively.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), 832–843 (1983)
2. Balbiani, P., Condotta, J.F., del Cerro, L.F.: A model for reasoning about bidimensional temporal relations. In: Proc. of KR 1998, pp. 124–130 (1998)
3. Baykan, C.A., Fox, M.D.: Spatial synthesis by disjunctive constraint satisfaction. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 11(4), 245–262 (1997)
4. Borning, A., Kuang-Hsu Lin, R., Marriott, K.: Constraint-based document layout for the web. Multimedia Syst. 8(3), 177–189 (2000)
5. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An overview. Fundamenta Informaticae 46(1-2), 1–29 (2001)
6. Condotta, J.F.: The augmented interval and rectangle networks. In: Proc. of KR 2000, pp. 571–579 (2000)

7. Dechter, R., Meiri, I., Pearl, J.: Temporal constraint networks. Artificial Intelligence 49(1-3), 61–95 (1991)
8. Gatterbauer, W., Bohunsky, P.: Table extraction using spatial reasoning on the CSS2 visual box model. In: Proc. of AAAI 2006, pp. 1313–1318 (2006)
9. Gerevini, A., Cristani, M.: On finding a solution in temporal constraint satisfaction problems. In: Proc. of IJCAI 1997, vol. 2, pp. 1460–1465 (1997)
10. Goyal, R., Egenhofer, M.: Consistent queries over cardinal directions across different levels of detail. In: Proc. of DEXA 2000, pp. 876–880 (2000)
11. Gerevini, A., Renz, J.: Combining topological and size information for spatial reasoning. Artificial Intelligence 137(1-2), 1–42 (2002)
12. Liu, W., Li, S.: Reasoning about cardinal directions between extended objects: The NP-hardness result. Artificial Intelligence 175(18), 2155–2169 (2011)
13. Liu, W., Li, S., Renz, J.: Combining RCC-8 with qualitative direction calculi: Algorithms and complexity. In: Proc. of IJCAI 2009, pp. 854–859 (2009)
14. Liu, W., Zhang, X., Li, S., Ying, M.: Reasoning about cardinal directions between extended objects. Artificial Intelligence 174, 951–983 (2010)
15. Mackworth, A.K.: Consistency in Networks of Relations. Artificial Intelligence 8(1), 99–118 (1977)
16. Meiri, I.: Combining qualitative and quantitative constraints in temporal reasoning. Artificial Intelligence 87(1-2), 343–385 (1996)
17. Montanari, A., Navarrete, I., Sciavicco, G., Tonon, A.: A tractable formalism for combining rectangular cardinal relations with metric constraints. In: Proc. of ICAART 2012, pp. 154–163 (2012)
18. Navarrete, I., Morales, A., Sciavicco, G., Cardenas, M.A.: Spatial reasoning with rectangular cardinal relations utility package for rectangular cardinal relations. Technical Report TR-DIIC 2/11, Universidad de Murcia (2011)
19. Navarrete, I., Sciavicco, G.: Spatial reasoning with rectangular cardinal direction relations. In: Proc. of the ECAI 2006 Workshop on Spatial and Temporal Reasoning, pp. 1–10 (2006)
20. Oro, E., Ruffolo, M., Staab, S.: SXPath - extending XPath towards spatial querying on web documents. Proc. of VLDB 2010 4(2), 129–140 (2010)
21. Papadias, D., Theodoridis, Y.: Spatial relations, minimum bounding rectangles, and spatial data structures. International Journal of Geographical Information Science 11(2), 111–138 (1997)
22. Skiadopoulos, S., Koubarakis, M.: On the consistency of cardinal directions constraints. Artificial Intelligence 163(1), 91–135 (2005)
23. van Beek, P.: Reasoning about qualitative temporal information. Artificial Intelligence 58(1-3), 297–326 (1992)
24. van Beek, P., Cohen, R.: Exact and approximate reasoning about temporal relations. Computation Intelligence 6(3), 132–147 (1990)
25. Vilain, M.B., Kautz, H.: Constraint propagation algorithms for temporal reasoning. In: Proc. of AAAI 1986, pp. 377–382 (1986)