Artificial Intelligence 174 (2010) 72–104

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

# An AGM-style belief revision mechanism for probabilistic spatio-temporal logics

John Grant [a,c], Francesco Parisi [b], Austin Parker [c], V.S. Subrahmanian [c,*]

[a] Towson University, Towson, MD 21252, USA
[b] Università della Calabria, 87036 Rende (CS), Italy
[c] University of Maryland, College Park, MD 20742, USA

ARTICLE INFO

*Article history:*
Received 27 February 2009
Received in revised form 21 September 2009
Accepted 6 October 2009
Available online 14 October 2009

*Keywords:*
Belief revision
Reasoning about motion
Spatio-temporal logics
Uncertainty management

ABSTRACT

There is now extensive interest in reasoning about moving objects. A probabilistic spatio-temporal (PST) knowledge base (KB) contains atomic statements of the form "Object $o$ is/was/will be in region $r$ at time $t$ with probability in the interval $[\ell, u]$". In this paper, we study mechanisms for belief revision in PST KBs. We propose multiple methods for revising PST KBs. These methods involve finding maximally consistent subsets and maximal cardinality consistent subsets. In addition, there may be applications where the user has doubts about the accuracy of the spatial information, or the temporal aspects, or about the ability to recognize objects in such statements. We study belief revision mechanisms that allow changes to the KB in each of these three components. Finally, there may be doubts about the assignment of probabilities in the KB. Allowing changes to the probability of statements in the KB yields another belief revision mechanism. Each of these belief revision methods may be epistemically desirable for some applications, but not for others. We show that some of these approaches cannot satisfy AGM-style axioms for belief revision under certain conditions. We also perform a detailed complexity analysis of each of these approaches. Simply put, all belief revision methods proposed that satisfy AGM-style axioms turn out to be intractable with the exception of the method that revises beliefs by changing the probabilities (minimally) in the KB. We also propose two hybrids of these basic approaches to revision and analyze the complexity of these hybrid methods.



## 1. Introduction

There are numerous applications where we need to reason about probabilistic spatio-temporal applications. A shipping company may be interested in continuously tracking the locations of its vehicles. As RFID tags become ever more common, companies (pharma, automotive, electronics) are interested in tracking supply items and in understanding where these items are now, and where they might be in the future. Military agencies are interested in tracking where vehicles might be — now and in the future. Cell phone companies are interested in when and where cell phones might be in the future in order to determine how best to balance load on cell towers. Moreover, all these applications have an essential component involving uncertainty. Predicting where a cell phone might be in the future may be derived probabilistically from past logs showing the phones' location. Likewise, predicting where and when an RFID tag will be is subject to uncertainty. Where and when a ship will reach a given geolocation is also subject to many forces that cannot be accurately specified, even when a schedule is available.

* Corresponding author.
*E-mail addresses:* jgrant@towson.edu (J. Grant), fparisi@deis.unical.it (F. Parisi), austinjp@cs.umd.edu (A. Parker), vs@cs.umd.edu (V.S. Subrahmanian).

0004-3702/$ – see front matter 
doi:10.1016/j.artint.2009.10.002

Methods to reason about probabilistic spatio-temporal (PST) information have emerged in recent years, both in databases [20] and in AI [5,18]. In this paper, we build upon the results of [20]. Our PST knowledge base contains a set of facts of the form $loc(id, r, t)[\ell, u]$ which informally states that "Object $o$ is somewhere in region $r$ at time $t$ with a probability between $\ell$ and $u$ (inclusive)". A more formal description will be provided shortly in the paper.

One important aspect of applications such as those mentioned above is that there is *continuous change*. As objects move, they encounter unexpected situations, leading to a continuous revision of estimates of where they might be in the future, as well as a revision of where they might have been in the past. Surprisingly, to date, we are not aware of any effort to handle revisions to such PST knowledge bases. A PST knowledge base $\mathcal{K}$ can be revised in many different ways. Clearly, when the insertion of a fact $ra$ into the knowledge base leads to no inconsistency, i.e. $\mathcal{K} \cup \{ra\}$ is consistent, then $ra$ can just be added to $\mathcal{K}$. However, when $\mathcal{K} \cup \{ra\}$ is inconsistent, then many different belief revision operations are possible.

In this paper, we focus on several different ways in which to revise $\mathcal{K}$ based on various epistemic intuitions.

- Following on much work in classical reasoning about inconsistency in AI, our revision could try to find a maximal (w.r.t. either subset inclusion or cardinality) subset $\mathcal{K}'$ of $\mathcal{K}$ that is consistent with $ra$ — in this case, the revision of $\mathcal{K}$ w.r.t. the update $ra$ is $\mathcal{K}' \cup \{ra\}$. We study these two revision strategies, show that they satisfy AGM-style axioms (defined in Section 2), and that they lead to computational intractability.
- It is also possible to revise $\mathcal{K}$ when $ra$ is inserted by minimally modifying the spatial, temporal, or object components in $\mathcal{K}$.
  - We first propose a revision mechanism based on just modifying the object ids in a PST KB. An application user or developer may wish to use this strategy for an application when there is reason to believe that the object ids are likely to be incorrect. This may occur, for instance, when the PST KB is generated using an image processing program (e.g. a car license plate reader) that may be "off".
  - We also propose a revision mechanism based on just modifying the temporal component in a PST KB. An application user or developer may wish to use this strategy for an application when there is reason to believe that the times reported are "off". This may be due to historical skepticism such as the belief that the clocks used to automatically generate PS KBs in the application are flawed.
  - We also propose a revision mechanism based on just modifying the spatial component in a PST KB. An application user or developer may wish to use this strategy for an application when there is reason to believe that the regions are inaccurate. This may be due to the fact that GPS transponders exhibited errors previously.

  We develop all these revision mechanisms. We show that spatial revisions may not satisfy AGM-style axioms and that the other mechanisms — though they satisfy AGM-style axioms — are computationally intractable.
- We also propose three revision mechanisms based on revising the probability intervals in a PST KB. In one, only the lower bound is modified, in another, only the upper bound is modified, and in the third, both may be simultaneously modified. We show that the last mechanism not only satisfies AGM-style axioms, but that there is a polynomial time algorithm to compute this update mechanism. An end user may use this mechanism when there is reason to believe that the probabilities in a PST KB are likely to be incorrect.
- We also propose a revision mechanism that allows simultaneous changes to each of the spatial, temporal, object, and probability components in PST KBs. The user can specify how unlikely each of these mechanisms may be wrong by setting appropriate weights.

Fig. 1 summarizes the types of revision methods proposed in this paper — it also states whether the method satisfies AGM-style axioms or not, and the computational complexity involved. As probabilistic revision is polynomial, we spend a fair amount of time focusing on speeding this up via the use of a suite of heuristics.

| Technique | AGM | Complexity |
|---|---|---|
| MAX-SUBSET (Def. 9) | All | coNP-complete |
| Indeterminate MAX-SUBSET (Sec. 3.3) | All but **(A6)** | PTIME |
| MAX-CARD (Def. 10) | Yes | coNP-complete |
| Spatial Revision (Def. 12) | No | N/A |
| Temporal Revision (Def. 13) | No | coNP-complete |
| Infinite time | No | coNP-hard |
| Fully connected reachability | Yes | coNP-hard |
| Object Revision (Def. 14) | No | coNP-complete |
| Lower Bound Revision (Def. 15) | No | N/A |
| Upper Bound Revision (Def. 15) | No | N/A |
| Probabilistic Revision (Def. 17) | Yes | PTIME |
| Hybrid Revision (Def. 21) | No | coNP-complete |

**Fig. 1.** A summary of this paper's results. Unless otherwise stated, all results assume that $T$ is finite.

This paper builds on some of our previous work on representing and reasoning about probabilistic spatio temporal (PST) knowledge bases [23]. Section 2 formalizes the notion of a PST KB from some of our past work, overviews AGM axioms for updating logical theories, and provides a linear programming based algorithm to check the consistency of PST KBs. Section 3 provides many possible ways of revising a PST KB and shows which of these methods satisfy AGM-style axioms, and also analyzes the computational complexity of implementing these methods. Section 4 provides two *hybrid* methods to revise PST KBs based on the basic methods proposed previously. We study whether these hybrid methods satisfy AGM-style axioms, as well as the computational complexity of these methods. Section 5 identifies a partitioning strategy that may be used to more quickly find an approximate solution to probabilistic revision problems. Section 6 compares our work with related work in the scientific literature. At the end, Section 7 identifies directions for future work on reasoning about PST KBs.

All proofs not given in the text are given in Appendices A and B.

## 2. Background: Formal model

[20,21] proposes a framework for probabilistic spatio-temporal reasoning in which we can reason about statements of the form "Object $o$ is/was/will be in region $r$ at time $t$ with a probability within the interval $[\ell, u]$". We assume the existence of some *finite* set *ID* of object ids. We generally assume a *finite* convex set $\mathcal{S}$ of points in a 2-dimensional space[1]; however, in some cases the 2-dimensionality of space is irrelevant and we simply deal with a set of points, $|\mathcal{S}| > 1$. We use $distance(p, q)$ to represent the Euclidean distance between $p$ and $q$. We assume that time, $T$, is represented by the set of all non-negative integers. However, in some cases, as specified later, we will assume that $T$ is a finite set: $T = \{0, 1, \ldots, N\}$ for some integer $N$. Though time may be infinite in theory, virtually all real world applications are only intended to last some number of years and so, for practical applications, it is reasonable to assume that PST knowledge bases are only intended to last for a finite window of time in the future. We also assume that an object cannot be at two different points at the same time, although two objects may be at the same point at the same time.

**Definition 1.** If $id \in ID$, $t \in T$, $r \subseteq \mathcal{S}$ $(r \neq \emptyset)$, then $loc(id, r, t)$ is called an ***ST atom***. A ***PST atom*** is an ST atom annotated with probabilistic parameter $[\ell, u]$, where $0 \leqslant \ell \leqslant u \leqslant 1$, and is denoted as $loc(id, r, t)[\ell, u]$.

Intuitively, an ST atom $loc(id, r, t)$ says that the object with the given *id* is somewhere on a location in region $r$ at time $t$. Let $Pr(loc(id, r, t))$ denote the probability that event $loc(id, r, t)$ occurs. The PST atom $loc(id, r, t)[\ell, u]$ means that $\ell \leqslant Pr(loc(id, r, t)) \leqslant u$. Soon we will define the concept of a PST knowledge base which contains a set of PST atoms, the probabilistic information about the location of objects at various times, as well as constraints on the movement of objects.

We now introduce the PST atoms for an example that will be used throughout the paper to illustrate various concepts.

**Example 1.** Fig. 2 shows (rectangular) regions $R_0, \ldots, R_4$ in Beijing. A worker living in the home at location $(3, 3)$ commutes to the factory at $(8, 7)$ every weekday from 7:30am to 8:00am. We will therefore use the time series $0, 1, 2, \ldots, 30$ to denote the 31 possible minutes between 7:30am and 8am (i.e. time 5 is 7:35). The worker can take one of many paths to work, but through observation, we determined that he is almost always in the vicinity of work after 7:58. In PST-syntax: $loc(worker, R_2, 28)[0.9, 1]$, $loc(worker, R_2, 29)[0.9, 1]$, $loc(worker, R_2, 30)[0.9, 1]$. Further, the worker is in the vicinity of work by 7:45 half the time: $loc(worker, R_2, 15)[0.5, 0.5]$. We know that the worker almost never travels anywhere in $R_1$: $loc(worker, R_1, 0)[0, 0.01], \ldots, loc(worker, R_1, 30)[0, 0.01]$, and that half the time, the worker gets breakfast at a place somewhere in $R_3$ at 7:45: $loc(worker, R_3, 15)[0.5, 0.5]$. We denote the knowledge base containing these atoms as $\mathcal{K}_{Beijing}$.

Probabilistic intervals such as $[\ell, u]$ allow for a more flexible approach that subsumes the commonly used single probability for each point: one can always use the singleton interval $[p, p]$. With probabilistic intervals, there are fewer restrictions on the data generation process — the probability need not be determined exactly, but only within a range. Furthermore, even if all data about basic events only contained point probabilities, the implied probabilities of complex (e.g. conjunctive or disjunctive) events end up being intervals unless additional assumptions such as independence assumptions are made. For instance, suppose an object is in region $r$ with 50% probability. Then we can infer that the object is in a region $r'$ disjoint from $r$ with a probability in the interval $[0, 0.5]$. We would not, however, be able to infer anything if all our inferences were limited to a notation allowing only point probabilities. Therefore we continue the tradition of the probabilistic spatial temporal logics in [20,21] with our use of probability intervals.

In the initial work in [20,21], it was assumed that there were no velocity constraints on moving objects — obviously, this was not realistic. Moreover, it was assumed that all points in $\mathcal{S}$ are reachable from all other points by all objects — obviously this is not always a valid assumption as well. For example, a 2-dimensional representation of the world (or even just of the state of Maryland), consists of regions that are reachable by some vehicles (e.g., cars) but not by others (e.g., boats) and vice

[1] The framework is easily extensible to higher dimensions. The finiteness requirement ensures that a potentially continuous 2-dimensional space is somehow discretized. Virtually all existing real world geographic information systems [27] assume that space is discretized into a grid of size $M \times N$ for some $M, N \geqslant 1$ and most geographic data structures such as quadtrees, R-trees, etc. supported by GISs make the same assumption.

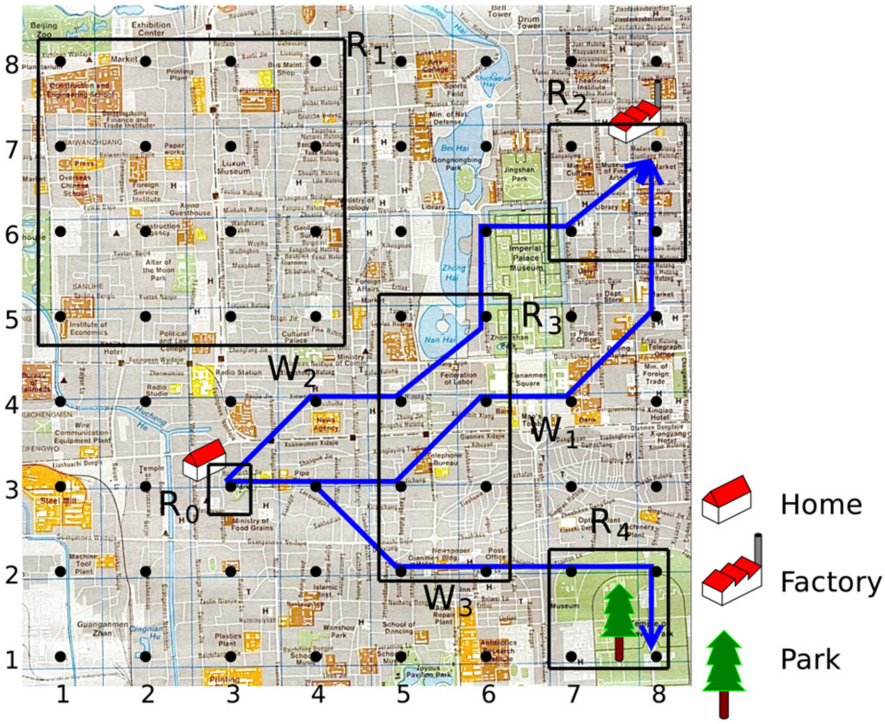


**Fig. 2.** An example PST knowledge base representing possible locations of a commuter in Beijing, China. The lines represent possible paths that can be taken to the factory and to the park.

versa. We provide a very general notion of a *reachability atom* and a *reachability definition* below that allows us to capture these concepts and use a more real-world framework than that provided in [20,21].

**Definition 2.** If $p_1, p_2 \in \mathcal{S}$ and $id \in ID$, then $reachable_{id}(p_1, p_2)$ is called a *reachability atom*. A *reachability definition RD*, is a finite set of reachability atoms.

Intuitively, the reachability atom $reachable_{id}(p_1, p_2)$ says that it is possible for the object *id* to reach location $p_2$ from location $p_1$ in one unit of time. Note that this definition is very general — what is reachable in one time point depends not only on the place $p_1$, but also the object *id*. For example, the same object may move at a very low speed when going from $p_1$ to $p_2$ (e.g., a steep upward slope) and at a different speed when going from $p_2$ to $p_3$. Likewise, different vehicles may also exhibit different maximal speeds depending upon the type of vehicle (trucks might go slower than Ferraris for instance). We also assume that the points in $\mathcal{S}$ are close enough together or the time units are long enough, so that an adjacent[2] point never requires more than one time unit to be reached if it can be reached at all.

Though the definition of *RD* appears to require explicit storage, this is not necessary when implementing a PST KB. *RD* may be dynamically determined — for instance by invoking a third party code base such as Google Maps. Without loss of generality, we assume that *RD* contains $reachable_{id}(p, p)$ for all $p \in \mathcal{S}$ and $id \in ID$. This merely says that if *id* is at *p*, it can reach the same point *p* within one time unit.

**Definition 3.** Given a reachability definition *RD*, we define $connected_{id}(p_1, p_2)$ as the transitive closure of $reachable_{id}(p_1, p_2)$.

A reachability definition is *fully connected* if $connected_{id}(p_1, p_2)$ holds for every $id \in ID$ and $p_1, p_2 \in \mathcal{S}$.

Thus $connected_{id}(p_1, p_2)$ means that *id* can get from $p_1$ to $p_2$ but not necessarily in one unit of time. We now present a simple example.

**Example 2.** If we allow the worker from Example 1 to travel at most a Manhattan distance of 2 each time point,[3] this creates a natural reachability definition $RD_{worker}$ where $reachable_{id}((x_1, y_1), (x_2, y_2)) \in RD_{worker}$ iff $|x_1 - x_2| + |y_1 - y_2| \leqslant 2$. This simple reachability definition is clearly fully connected.

[2] Points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ are adjacent if $|x_1 - x_2| + |y_1 - y_2| = 1$.

[3] Manhattan distance is the standard $L_1$ norm: for $(x_1, y_1)$ and $(x_2, y_2)$ it is $|x_1 - x_2| + |y_1 - y_2|$.

We now formally define a PST KB.

**Definition 4** *(PST KB).* A ***PST knowledge base*** is a pair $(\mathcal{K}, RD)$ where $\mathcal{K}$ is a finite set of PST atoms and $RD$ is a reachability definition.

**Example 3.** Using the set $\mathcal{K}_{Beijing}$ from Example 1, and the reachability definition $RD_{worker}$ from Example 2, we have the PST knowledge base: $(\mathcal{K}_{Beijing}, RD_{worker})$.

*Throughout this paper, we assume the existence of an arbitrary, but fixed reachability definition, RD, and refer to $\mathcal{K}$ as a PST KB.* We define the semantics of PST KBs through worlds.

**Definition 5** *(World).* A world $w$ is a function, $w : ID \times T \to \mathcal{S}$ such that for all objects $id$, points $p_1$, $p_2$, and time points $t_1$, $t_2$ with $t_2 = t_1 + 1$ if $w(id, t_1) = p_1$ and $w(id, t_2) = p_2$ then $reachable_{id}(p_1, p_2) \in RD$. $\mathcal{W}$ is the set of all worlds. To simplify formulas later we will assume that $w$ ranges over $\mathcal{W}$.

A world specifies where in space $\mathcal{S}$ an object $o$ is at time $t$. A world $w$ can be represented by the set of ST atoms $loc(id, \{p\}, t)$ such that $w(id, t) = p$. We write $w \models loc(id, r, t)$ iff $w(id, t) \in r$. Example worlds $W_1$, $W_2$, and $W_3$ can be seen in Fig. 2. An interpretation assigns a probability to each world.

**Definition 6** *(Interpretation).* An interpretation $I$ is a probability distribution over $\mathcal{W}$.

$I(w)$ is the probability that $w$ describes the actual locations of all the objects at all the time values.

**Example 4.** Continuing with Example 1, two possible commutes the worker may take are shown in Fig. 2 as paths $W_1$ and $W_2$. There is a further path the worker may take to the park labeled $W_3$. The function corresponding to $W_1$ has the worker staying at home until 7:40am $W_1(worker, t) = (3, 3)$ for $t = 0, \dots, 10$, then $W_1(worker, 11) = (4, 3)$, and $W_1(worker, 12) = (5, 3)$. At this point the worker stops for breakfast: $W_1(worker, t) = (6, 4)$ for $t = 13, \dots, 26$. Then the worker continues to work: $W_1(worker, 27) = (7, 4)$, $W_1(worker, 28) = (8, 5)$, $W_1(worker, 29) = (8, 6)$, and $W_1(worker, 30) = (8, 7)$. The function for $W_2$ has the worker move directly along the path and staying at work, i.e. $W_2(worker, 0) = (3, 3)$, $W_2(worker, 1) = (4, 4)$, $W_2(worker, 2) = (5, 4)$, $W_2(worker, 3) = (6, 5)$, $W_2(worker, 4) = (6, 6)$, $W_2(worker, 5) = (7, 6)$, and $W_2(worker, 6 \dots 30) = (8, 7)$.

An example interpretation $I$ assigns probability 0.5 to world $W_1$, probability 0.5 to world $W_2$, and probability 0 to all other worlds including $W_3$.

The definition of satisfaction of a PST atom by an interpretation is as follows.

**Definition 7** *(Satisfaction/Entailment).* Interpretation $I$ satisfies the PST atom $a = loc(id, r, t)[\ell, u]$, denoted $I \models a$, iff $\sum_{w \models loc(id,r,t)} I(w) \in [\ell, u]$. $I$ satisfies $\mathcal{K}$, denoted $I \models \mathcal{K}$, iff $I$ satisfies all $a \in \mathcal{K}$. $\mathcal{K}$ entails $\mathcal{K}'$, denoted $\mathcal{K} \models \mathcal{K}'$ (resp. $\mathcal{K}$ entails $a$, denoted $\mathcal{K} \models a$) iff all $I$ satisfying $\mathcal{K}$ also satisfy $\mathcal{K}'$ (resp. $a$).

$\mathcal{K}$ is *consistent* iff there is an interpretation $I$ that satisfies it. $\mathcal{K}$ and $\mathcal{K}'$ are *equivalent* (denoted $\mathcal{K} \equiv \mathcal{K}'$) iff for all interpretations $I$, $I \models \mathcal{K}$ iff $I \models \mathcal{K}'$. For instance, the interpretation from Example 4 satisfies the PST knowledge base $\mathcal{K}_{Beijing}$ from Example 1. Hence $\mathcal{K}_{Beijing}$ is consistent. *We will assume in all our work that we start with a consistent KB.*

A PST atom $a$ *is consistent with* $\mathcal{K}$ iff $\mathcal{K} \cup \{a\}$ is consistent.

**Example 5.** The atom $loc(worker, R_4, 29)[0.75, 0.75]$ is not consistent with the knowledge base $\mathcal{K}_{Beijing}$ from Example 1 due to the fact that $\mathcal{K}_{Beijing}$ states that the worker is in region $R_2$ at time 29 with probability in $[0.9, 1]$ and $R_2$ is disjoint from $R_4$. The total probability of the worker being on the map at time 29 would then exceed 1.

However, $loc(worker, R_4, 29)[0.1, 0.1]$ is consistent with $\mathcal{K}_{Beijing}$ — consider for instance an interpretation that gives to the world $W_3$ a probability 0.1 (making the probability of being in region $R_4$ also 0.1) and world $W_1$ a probability of 0.9 (making the probability of being in region $R_2$ 0.9 as $\mathcal{K}_{Beijing}$ requires).

We are interested in studying the revision of PST KBs when a revision atom $ra$ is added to $\mathcal{K}$. We start by presenting AGM-style postulates [1] for this purpose. A revision operator $\dotplus$ is a binary function that takes $\mathcal{K}$ and $ra$ as input, and produces $\mathcal{K} \dotplus ra$ as output. $\dotplus$ is required to satisfy these AGM-style axioms[4] expressed in our framework as given below.

[4] As PST KBs are atomic, we do not discuss AGM axioms involving negation and disjunction.

(A1) $\mathcal{K} \dotplus ra$ is a PST KB.
(A2) $\mathcal{K} \dotplus ra \models ra$.
(A3) $(\mathcal{K} \cup \{ra\}) \models (\mathcal{K} \dotplus ra)$.
(A4) If $ra$ is consistent with $\mathcal{K}$ then $(\mathcal{K} \dotplus ra) \models (\mathcal{K} \cup \{ra\})$.
(A5) $\mathcal{K} \dotplus ra$ is inconsistent iff $\{ra\}$ is inconsistent.
(A6) If $ra \equiv ra'$ then $\mathcal{K} \dotplus ra \equiv \mathcal{K} \dotplus ra'$.

We say a revision strategy is *AGM-compliant* or that it satisfies the AGM axioms if it satisfies Axioms (A1)–(A6).

### 2.1. Consistency checking

We can check the consistency of a PST KB by solving a linear program. Because linear programs can be solved in time polynomial in their input, consistency checking will run in polynomial time when the number of time points is bounded *a priori*.

The linear program we use contains variables of the form $v_{id,t,p,q}$, each representing the probability that object $id$ will be at point $p$ at time $t$ and then at point $q$ at time $t+1$ (i.e. the probability of $loc(id,\{p\},t) \wedge loc(id,\{q\},t+1)$).

For convenience, let $minT(\mathcal{K})$ (resp. $maxT(\mathcal{K})$) be the minimum (resp. maximum) time point referenced in $\mathcal{K}$. When $T$ is finite we have *a priori* bounds for $minT(\mathcal{K})$ and $maxT(\mathcal{K})$. For technical reasons, we include the time point $maxT(\mathcal{K})+1$ in $T$. A detailed explanation of the constraints is given after the definition.

**Definition 8** *($LP(\mathcal{K})$).* $LP(\mathcal{K})$, the linear program for $\mathcal{K}$ contains the following constraints where $minT(\mathcal{K}) \leqslant t \leqslant maxT(\mathcal{K})$:

(1) For all $loc(id,r,t)[\ell,u] \in \mathcal{K}$:

$$\ell \leqslant \sum_{p\in r}\sum_{q\in\mathcal{S}} v_{id,t,p,q} \quad \text{and} \quad u \geqslant \sum_{p\in r}\sum_{q\in\mathcal{S}} v_{id,t,p,q}.$$

(2) For all $id$, $t$: $\sum_{p\in\mathcal{S}}\sum_{q\in\mathcal{S}} v_{id,t,p,q} = 1$.
(3) For all $p,q \in \mathcal{S}$ and all $id$, $t$: $v_{id,t,p,q} \geqslant 0$.
(4) For all $p,q \in \mathcal{S}$ and all $id$, $t$ such that $reachable_{id}(p,q) \notin RD$: $v_{id,t,p,q} = 0$.
(5) For all $p \in \mathcal{S}$ and $id$, $t$: $\sum_{q\in\mathcal{S}} v_{id,t,q,p} = \sum_{q\in\mathcal{S}} v_{id,t+1,p,q}$.

The constraints each serve their purpose. Constraint (1) ensures that a solution places the object in $r$ with a probability between $\ell$ and $u$, as required by the atom $loc(id,r,t)[\ell,u]$. Constraints (2) and (3) ensure that for each $id$ and $t$, the $v_{id,t,p,q}$ variables jointly represent a proper probability distribution (i.e. sum to 1 and have non-negative probabilities). Constraint (4) enforces the reachability definition by assigning a probability of 0 to travel points $p$ and $q$ that cannot be reached in one time step. Note that if there is a third point $p'$ such that $reachable_{id}(p,p')$ and $reachable_{id}(p',q)$, then the object could still travel between $p$ and $q$ because $v_{id,t,p,p'}$ and $v_{id,t+1,p',q}$ can both be non-zero. It would just take two time steps, which is sensible when reachability says the object cannot travel from $p$ to $q$ in one time step. Constraint (5) is central to the correctness of our variable formulation. Since each variable $v_{id,t,p,q}$ gives the probability of being at $p$ and moving to $q$, we need something to ensure that $id$ can in fact be at $p$ at time $t$ with the probability given by $v_{id,t,p,q}$. Constraint (5) provides that insurance, by forcing the probability of an object entering any point to be the same as the object leaving that point.

Note that when a constraint of the form $v_{id,t,p,q} = 0$ is included in $LP(\mathcal{K})$ because of clause (4) above, the redundant constraint $v_{id,t,p,q} \geqslant 0$ generated by clause (3) above can be eliminated, as well as all occurrences of this variable in other constraints. We take advantage of these simplifications in our implementation and solve linear programs consisting only of variables $v_{id,t,p,q}$ for which $reachable_{id}(p,q)$ holds.

The following result gives a one to one correspondence between the problem of checking the consistency of $\mathcal{K}$ and the problem of checking the solvability of the constraints $LP(\mathcal{K})$.

**Theorem 1.** *(See Proposition 4 from [22].) $LP(\mathcal{K})$ has a solution iff $\mathcal{K}$ is consistent.*

**Proof.** We present a sketch of the proof here. A complete proof can be found in [22].

($\Rightarrow$): Let $\theta$ be a solution satisfying $LP(\mathcal{K})$. To construct a satisfying interpretation $I$, let $\alpha[id,p]$ be the probability that $id$ is at $p$ at the first time point, $minT(\mathcal{K})$ computed from $\theta$ as follows: $\alpha[id,p] = \sum_{p'\in\mathcal{S}} v[id, minT(\mathcal{K}), p, p']\theta$, where we write $v[id,t,p,q]\theta$ for the value assigned by $\theta$ to $v_{id,t,p,q}$. Now define $\delta[id,t,p,p']$ to be the probability of moving from $p$ to $p'$ at time $t$, or:

$$\delta[id,t,p,p'] = \frac{v[id,t,p,p']\theta}{\sum_{p''\in\mathcal{S}} v[id,t,p,p'']\theta}$$

(when $\sum_{p''\in\mathcal{S}} v[id,t,p,p'']\theta = 0$, $\delta$ is defined to be 0 as well). We can now define $I$ for all $w \in \mathcal{W}$ as:

$$I(w) = \prod_{id\in ID} \alpha\big[id, w(id,0)\big] \prod_{t,t+1\in T} \delta\big[id, t, w(id,t), w(id,t+1)\big].$$

$I$ can be shown to be an interpretation satisfying $\mathcal{K}$.

($\Leftarrow$): Let $I$ be an interpretation which satisfies $\mathcal{K}$. Define a variable assignment $\theta$ such that: $v_{id,t,p,p'} = \sum_{w\in\mathcal{W},\, w(id,t)=p,\, w(id,t+1)=p'} I(w)$. $\theta$ can be shown to be a solution to $LP(\mathcal{K})$. $\square$

The theorem yields a straightforward consistency checking algorithm: check if $LP(\mathcal{K})$ has a solution using standard linear programming solvers.

To determine the running time of this algorithm, we count the number of variables and equations in $LP(\mathcal{K})$. The number of variables is dependent upon the number of ids in the knowledge base, which is at most $|\mathcal{K}|$, the number of points in space, which is $|\mathcal{S}|$, and the number of time points $n_t = maxT(\mathcal{K}) - minT(\mathcal{K})$. This gives an upper bound of $O(|\mathcal{K}| \cdot |\mathcal{S}|^2 \cdot n_t)$ variables in $LP(\mathcal{K})$.

The number of constraints in $LP(\mathcal{K})$ is $2 \cdot |\mathcal{K}|$ for (1), plus one constraint per *id* and *t* for (2), plus at most $|\mathcal{S}|^2$ constraints per *id* and *t* for (3) and (4), plus $|\mathcal{S}|$ constraints per *id* and *t* for (5) giving $O(|\mathcal{K}| \cdot |\mathcal{S}|^2 \cdot n_t)$ constraints. The entire linear program's size can be bounded by the number of variables times the number of constraints, that is, $O((|\mathcal{K}| \cdot |\mathcal{S}|^2 \cdot n_t)^2)$.

Linear programs are solvable in time cubic in the size of the linear constraints [15]. The running time required to find a solution to $LP(\mathcal{K})$, thereby determining the consistency of $\mathcal{K}$, is $O((|\mathcal{K}| \cdot |\mathcal{S}|^2 \cdot n_t)^6)$. Therefore consistency checking is polynomial in the size of the input knowledge base.

## 3. Basic belief revision strategies

In this section, we present six "basic" ways of revising consistent PST KBs. The first two ways use the standard method studied extensively in logical reasoning — eliminate entire atoms from $\mathcal{K}$ when the *ra* being inserted conflicts with $\mathcal{K}$. To adhere to the principle of minimal change, we can either remove a minimal subset of $\mathcal{K}$ or the smallest number of possible PST atoms from $\mathcal{K}$, leading to the first and second basic belief revision strategies (called MAX-SUBSET and MAX-CARD revision strategies, respectively).

Subsequently, we note that it might be possible to restore consistency by changing the region, or the time, or the id, or the probability bounds associated with PST atoms in $\mathcal{K}$ when a new PST atom *ra* is inserted. These lead to another four types of basic ways of revising PST KBs when insertions occur.

We study these methods in the rest of this section.

### 3.1. Maximal consistent subset revision

We can define a revision operator $\dot{+}_m$ based on maximal consistent subsets as follows.

**Definition 9.** Suppose $\mathcal{K}$ is a PST KB, *ra* a PST atom, and $\mathcal{K}' \subseteq \mathcal{K}$. Then $\mathcal{K}' \cup \{ra\}$ *accomplishes the revision of* $\mathcal{K}$ *by adding ra via the subset strategy* iff $\mathcal{K}'$ is a subset of $\mathcal{K}$ and $\mathcal{K}' \cup \{ra\}$ is consistent. We say that $\mathcal{K}' \cup \{ra\}$ *accomplishes the revision of* $\mathcal{K}$ *by adding ra via the max-subset strategy* iff it accomplishes the revision of $\mathcal{K}$ by adding *ra* via the subset strategy and there is no other $\mathcal{K}'' \cup \{ra\}$ that accomplishes the same revision such that $\mathcal{K}' \subsetneq \mathcal{K}''$.

This definition does not necessarily determine a unique revision. To achieve uniqueness a strict total ordering can be induced on all $\mathcal{K}'$ satisfying the above definition and the minimal element picked. *Throughout the rest of this paper, we assume such a strict total ordering $O_T$ is available*. We further assume that $O_T$ is polynomially computable: that is, one can determine the relationship between $\mathcal{K}$ and $\mathcal{K}'$ according to $O_T$ in polynomial time. We use the notation $\mathcal{K} \dot{+}_m ra$ to denote *the* $\mathcal{K}' \cup \{ra\}$ that accomplishes the revision of $\mathcal{K}$ by adding *ra* via the *max-subset* strategy such that $\mathcal{K}'$ is minimal under order $O_T$.

The following example exhibits both how maximal subset revision and the total ordering $O_T$ will function.

**Example 6.** We revisit the set of atoms $\mathcal{K}_{Beijing}$ from Example 1 with the reachability definition $RD_{worker}$ from Example 2.

Suppose, in that example, that a camera in the park shows the worker there exactly 50% of the time at 7:45am. This conflicts with the knowledge base $\mathcal{K}_{Beijing}$, according to which the worker is either near the factory or in $R_3$ both with 50% probability, and the worker cannot be in three different places each with 50% probability at the same time! The revision atom here is: $loc(worker, R_4, 15)[0.5, 0.5]$.

For explication purposes, we list the relevant subset of $\mathcal{K}_{Beijing}$:

$$\big\{loc(worker, R_3, 15)[0.5, 0.5], loc(worker, R_2, 15)[0.5, 0.5]\big\}.$$

To achieve consistency with the revision atom, only one of these atoms must be removed from $\mathcal{K}_{Beijing}$. Therefore there are two possible maximal subset revisions:

- $\mathcal{K}_1 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_3, 15)[0.5, 0.5]\}$,
- $\mathcal{K}_2 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_2, 15)[0.5, 0.5]\}$

(note that other sorts of revision, where the probabilities, the regions, the time, or even the ID are changed, will be addressed later in this paper). Both $\mathcal{K}_1$ and $\mathcal{K}_2$ are consistent with the revision atom $loc(worker, R_4, 15)[0.5, 0.5]$, so to determine which of $\mathcal{K}_1$ and $\mathcal{K}_2$ will be $\mathcal{K}_{Beijing} \dot{+}_m loc(worker, R_4, 15)[0.5, 0.5]$, we consult the ordering $O_T$. If $\mathcal{K}_1$ is "smaller" than $\mathcal{K}_2$ according to $O_T$, then it is the answer. Suppose that in this case $O_T$ prefers atoms associated with $R_2$ to atoms associated with $R_3$, making $\mathcal{K}_1$ the "smaller" of the two revisions. Therefore $\mathcal{K}_{Beijing} \dot{+}_m loc(worker, R_4, 15)[0.5, 0.5] = \mathcal{K}_1 \cup \{loc(worker, R_4, 15)[0.5, 0.5]\}$.

We now verify that $\dot{+}_m$ satisfies the AGM axioms.

**Proposition 1.** *$\dot{+}_m$ is AGM-compliant.*

**Algorithm 1.** Computes the maximal subset revision according to $O_T$.

**MaxSubset**($\mathcal{K}$, *ra*)
  Let $list = [\mathcal{K}]$ {*list* is the list of maximal subsets of $\mathcal{K}$}
  **while** *list* is not empty **do**
    $nextList = []$ {initialize the list for the next iteration.}
    {Traverse *list* in order.}
    **for** $\mathcal{K}' \in list$ **do**
      **If** $\mathcal{K}' \cup \{ra\}$ is consistent **then** continue to next $\mathcal{K}'$.
      {Remove each possible element from $\mathcal{K}'$}
      **for** $atm \in \mathcal{K}'$ **do**
        add $\mathcal{K}' \setminus \{atm\}$ to *nextList*.
      **end for**
    **end for**
    Set $list = nextList$.
    remove duplicates from *list* {Don't check the same revision twice.}
  **end while**
  **return** the minimal member of *list* according to $O_T$.

The **MaxSubset** algorithm shown as Algorithm 1 correctly computes the $\dot{+}_m$ revision operator. However, as the following result shows, the decision problem associated with computing $\dot{+}_m$ is intractable (so long as $P \neq NP$).

**Theorem 2.** *Given PST KBs $\mathcal{K}$ and $\mathcal{K}'$, and revision atom ra, determining if $\mathcal{K}' \cup \{ra\} = \mathcal{K} \dot{+}_m ra$ is coNP-complete.*

**Proof Sketch.** Membership follows from the fact that if $\mathcal{K}'$ is not the best revision, then a witness $\mathcal{K}''$ can be determined to be a better revision in polynomial time (since consistency is a polynomial time operation). That the problem is coNP-hard is done by reduction from the MCSS problem (Definition 29 in Appendix A), which, according to Lemma 1 (also in Appendix A), is coNP-hard. □

### 3.2. $O_T$

The total ordering $O_T$ plays an essential role in the definition of the $\dot{+}$ operators used in this paper. The template we use for defining the various revision methods will be to introduce a method by which the knowledge base can be changed to create consistency (e.g. taking subsets of the knowledge base as above or changing some aspect of the data in the knowledge base). It will then generally be infeasible, without compromising the generality of the approach, to say which of the potential consistent changes is the best revision. We can easily imagine different subsets preferred in different application domains, perhaps based on how the atoms were created, or their importance if true, etc. We therefore abstract out that last choice with $O_T$, letting the user-supplied total ordering say which of the potential solutions is "best" for the given application. This is not an original approach, as it is similar to the selection mechanisms in the original work on AGM revision [1].

For our purposes, we suppose that $O_T$ runs in polynomial time in the representation and comparison of any two knowledge bases. This allows many possibilities for $O_T$: it could prefer knowledge bases with regions closer to a given point of interest, or $O_T$ could prefer knowledge bases with atoms further in the future, or prefer knowledge bases with tighter probability intervals, and so on.

Apart from establishing a unique answer to a given revision, the ordering $O_T$ will ensure the satisfaction of Axiom (A6). That axiom requires that even when the revision atoms differ syntactically, revision by either atom results in the same answer. Since consistency with the revision atom decides the set of possible revisions, the $O_T$ minimal revision will be the same regardless of the syntax of semantically equivalent revision atoms.

For these reasons $O_T$ will be used as part of the $\dotplus$ operators we define throughout this paper.

### *3.3. Indeterminate MAX-SUBSET revision is polynomial*

When we examine the proof of Theorem 2, we note that the source of the complexity of maximal subset revision is the total ordering $O_T$. The proof that determining if $\mathcal{K}'$ is $\mathcal{K} \dotplus \{ra\}$ is coNP-complete relies on the construction of a specific total ordering $O_T$. However, if we are willing to give up *uniqueness* and ignore $O_T$ (as has been commonly done in various papers in the past [3]), then the result below shows that this problem can be solved in polynomial time.

**Theorem 3.** *Given PST KBs $\mathcal{K}$ and $\mathcal{K}'$ and PST atom ra, determining whether $\mathcal{K}' \cup \{ra\}$ accomplishes the revision of $\mathcal{K}$ by adding ra via the max-subset strategy according only to Definition* 9 (irrespective of the order $O_T$) *can be accomplished in polynomial time w.r.t. the size of $\mathcal{K}$.*

To prove this theorem, we give a polynomial-time algorithm that returns true iff $\mathcal{K}'$ is the max-subset revision of $\mathcal{K}$ w.r.t. *ra*, and returns false iff $\mathcal{K}'$ is not the max-subset revision of $\mathcal{K}$ (note that $O_T$ is used to establish *which* max-subset revision is $\mathcal{K} \dotplus_m ra$, and is therefore inconsequential).

### *3.4. Maximal cardinality (MAX-CARD) revision*

Max-subset revision turned out to be coNP-complete, but only because of the total ordering $O_T$. In this section, we introduce max-cardinality revision, where instead of ensuring that there is no superset consistent with the revision, one must ensure that there is no larger-cardinality subset of the original knowledge base consistent with the revision atom.

**Definition 10.** Suppose $\mathcal{K}$ is a PST KB, *ra* is a PST atom, and $\mathcal{K}' \subseteq \mathcal{K}$. We say that $\mathcal{K}' \cup \{ra\}$ *accomplishes the revision of $\mathcal{K}$ by adding ra via the max-cardinality strategy* iff it accomplishes the revision of $\mathcal{K}$ by adding *ra* via the subset strategy and there is no other $\mathcal{K}'' \cup \{ra\}$ that accomplishes the revision via the subset strategy and $|\mathcal{K}'| < |\mathcal{K}''|$.

Again, the issue of uniqueness is resolved through a strict total ordering $O_T$. We denote the $O_T$-minimal knowledge base accomplishing max-cardinality revision of $\mathcal{K}$ with respect to *ra* as: $\mathcal{K} \dotplus_c ra$. However, unlike max-subset revision, max-cardinality revision is coNP-hard regardless of $O_T$.

**Theorem 4.** *Determining if $\mathcal{K}'$ accomplishes max-cardinality revision of $\mathcal{K}$ by adding ra is coNP-complete.*

**Proof Sketch.** Membership can be denied by a witness $\mathcal{K}''$ that has larger cardinality than $\mathcal{K}'$ and is consistent with the revision, so the problem is in coNP. coNP-hardness is established by reduction from the MCSS problem (Definition 29 in Appendix A), which, according to Lemma 1 (also in Appendix A), is coNP-hard.

Max cardinality subset revision may be expensive to compute; however, it still satisfies the AGM axioms. □

**Proposition 2.** *For knowledge base $\mathcal{K}$ and revision atom ra, $\mathcal{K} \dotplus_c ra$ is an AGM-compliant revision function.*

We now show that we can find maximal cardinality revisions by solving a *mixed integer linear program*.[5]

Consider a modified version of *LP*($\mathcal{K}$) (see Definition 8) where for each PST atom $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i]$ in $\mathcal{K}$, we include a binary (integer) variable $\delta_i$. Our intention is that $\delta_i = 0$ implies that $a_i$ is in the revised KB, while $\delta_i = 1$ means $a_i$ is not in the revised KB.

**Definition 11** *(Maximal Cardinality Subset Revision Program).* Let the Maximal Cardinality Subset Revision Program for $\mathcal{K}$ and *ra*, *MCSRP*($\mathcal{K}$, *ra*), contain the following constraints:

1. For each $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}$:
   (a) $\ell_i - \delta_i \leqslant (\sum_{p \in r_i} \sum_{q \in \mathcal{S}} v_{id_i, t_i, p, q}) \leqslant u_i + \delta_i$,
   (b) $\delta_i \in \{0, 1\}$.

[5] A linear program is a set of linear constraints and objective function where all variables range over the reals. An integer program is a set of linear constraints (and objective function) where all variables range over integers. A mixed integer linear program is one where some variables may range over integers, while others may range over the reals.

2. For $ra = loc(id', r', t')[\ell, u]$:
   (a) $\ell \leqslant \sum_{p \in r'} \sum_{q \in \mathcal{S}} v_{id',t',p,q} \leqslant u$.
3. For each *id* in the knowledge base and each $t$ in $T$:
   (a) For all $p, q \in \mathcal{S}$, $v_{id,t,p,q} \geqslant 0$.
   (b) $\sum_{p \in \mathcal{S}} \sum_{q \in \mathcal{S}} v_{id,t,p,q} = 1$.
   (c) For all $p, q \in \mathcal{S}$, if $reachable_{id}(p,q) \notin RD$: $v_{id,t,p,q} = 0$.
   (d) For all $p \in \mathcal{S}$: $\sum_{q \in \mathcal{S}} v_{id,t,q,p} = \sum_{q \in \mathcal{S}} v_{id,t+1,p,q}$.

The following result shows that there is a one to one correspondence between the solutions of a linear program associated with $MCSRP(\mathcal{K}, ra)$ and a revision that accomplishes the insertion of *ra* into $\mathcal{K}$ using the max-cardinality strategy.

**Theorem 5.** *Suppose that $\mathcal{K}$ is a PST KB and ra a PST atom. Let $\theta$ be a solution of the optimization problem*

$$\textbf{minimize} \sum_{a_i \in \mathcal{K}} \delta_i \quad \textbf{subject to } MCSRP(\mathcal{K}, ra).$$

*Then $\mathcal{K}' = \{a_i \in \mathcal{K} \mid \delta_i \theta = 0\} \cup \{ra\}$ accomplishes the revision of $\mathcal{K}$ by adding ra via the max-cardinality strategy.*

The preceding result provides a straightforward algorithm to find a max-cardinality revision. Just construct $MCSRP(\mathcal{K}, ra)$, solve the linear program stated in the preceding theorem, and use the solution as indicated in the preceding theorem.

### *3.5. Minimizing spatial change*

Now we can consider revising $\mathcal{K}$ by changing the spatial component $r$ of PST atoms in $\mathcal{K}$. A *spatial revision* of PST atom $a = loc(id, r, t)[\ell, u]$ is an atom of the form $a' = loc(id, r', t)[\ell, u]$.

We define the distance $d_S(a, a')$ as the Euclidean distance between points in the regions given by: $(\sum_{p \in r} \min_{p' \in r'} distance(p, p')) + (\sum_{p' \in r'} \min_{p \in r} distance(p, p'))$.[6] A spatial revision $\mathcal{K}'$ of $\mathcal{K}$ contains at most one spatial revision of each atom in $\mathcal{K}$. The distance, $(d_S(\mathcal{K}, \mathcal{K}'))$, is the sum of the distances between the individual atoms and their associated spatial revision.

**Definition 12.** A spatial revision $\mathcal{K}'$ of $\mathcal{K}$ by adding *ra* is *optimal* iff $\mathcal{K}' \cup \{ra\}$ is consistent and there is no other spatial revision $\mathcal{K}''$ of $\mathcal{K}$ by adding *ra* such that $\mathcal{K}'' \cup \{ra\}$ is consistent and $d_S(\mathcal{K}, \mathcal{K}'') < d_S(\mathcal{K}, \mathcal{K}')$.

As in the case of other revision strategies, there may be multiple optimal spatial revision strategies and we use a total ordering $O_T$ to obtain uniqueness. We write $\mathcal{K} \dotplus_s ra$ to denote this optimal spatial revision $\mathcal{K}'$. We now give an example of spatial change.

**Example 7.** Consider again the revision from Example 6 where the knowledge base $\mathcal{K}_{Beijing}$ is being revised by the atom: $loc(worker, R_4, 15)[0.5, 0.5]$. It could be that the data collection procedures determined the regions via some potentially erroneous method, and that therefore the inconsistency can be fixed by changing the regions in the knowledge base with spatial revision. One possible spatial revision of $\mathcal{K}_{Beijing}$ replaces the atom $loc(worker, R_3, 15)[0.5, 0.5]$ with the atom $loc(worker, R_3 \cup \{(7,2)\}, 15)[0.5, 0.5]$. We call this spatial revision $\mathcal{K}^s_{Beijing}$. We note that $d_S(\mathcal{K}_{Beijing}, \mathcal{K}^s_{Beijing})$ is one, which is the minimal possible non-zero value for $d_S$. Therefore, $\mathcal{K}^s_{Beijing}$ is an optimal spatial revision. In this case, there are no other optimal spatial revisions, however, when there are other optimal spatial revisions, we use $O_T$ to choose between them. For instance, if we used $d^*_S$ from footnote 6 as the spatial distance function, then $\mathcal{K}^s_{Beijing}$ would have the same spatial distance from $\mathcal{K}_{Beijing}$ as the spatial revision $\mathcal{K}^{s2}_{Beijing}$, which substitutes $loc(worker, R_3 \cup \{(8,2)\}, 15)[0.5, 0.5]$ for $loc(worker, R_3, 15)[0.5, 0.5]$. In this case, both $\mathcal{K}^s_{Beijing}$ and $\mathcal{K}^{s2}_{Beijing}$ would be optimal spatial revisions (they both have the same distance from $\mathcal{K}_{Beijing}$ according to $d^*_S$), but only the one deemed minimal according to $O_T$ would be $\mathcal{K}_{Beijing} \dotplus_s loc(worker, R_4, 15)[0.5, 0.5]$. We would expect a reasonable $O_T$ to prefer knowledge bases with connected regions to knowledge bases with unconnected regions, meaning that $\mathcal{K}_{Beijing} \dotplus_s loc(worker, R_4, 15)[0.5, 0.5]$ would be $\mathcal{K}^s_{Beijing} \cup \{loc(worker, R_4, 15)[0.5, 0.5]\}$.

The following theorem characterizes the cases where a spatial revision satisfying the AGM Axioms (A1)–(A6) is possible.

**Theorem 6.** *Let $\mathcal{K}$ be any knowledge base and $|\mathcal{S}| > 2$. An AGM-compliant spatial revision is possible for every atom $ra = loc(id, r, t)[\ell, u]$ where r is a strict subset of $\mathcal{S}$ iff for all $a_i = loc(id, r_i, t)[\ell_i, u_i] \in \mathcal{K}$ either $l_i = 0$ or $u_i = 1$.*

---

[6] Many other distance functions can also be defined (e.g. we could set the distance to be $d^*_S(a, a') = |r \cup r'| - |r \cap r'|$); however, the results in this section hold irrespective of the specific distance function.

Next we give a specific example where no spatial revision is possible.

**Example 8.** We revisit the knowledge base $\mathcal{K}_{Beijing}$ from Example 1. Recall that $\mathcal{K}_{Beijing}$ contains the atom $loc(worker, R_3, 15)[0.5, 0.5]$. When given the revision atom $loc(worker, \{(5,3)\}, 15)[1,1]$, which is inconsistent with $\mathcal{K}_{Beijing}$, one might suppose that it is inconsistent with the knowledge base due to some problem with the regions associated with the atoms, and that simply by fixing those regions, we might regain consistency. However, it is not that easy in this case: the revision atom is inconsistent with $loc(worker, R_3, 15)[0.5, 0.5]$, since $(5,3) \in R_3$. The revision atom forces a probability mass of 1 for $R_3$ while the knowledge base forces a probability mass of 0.5 for $R_3$. However, even if we changed the region in the atom $loc(worker, R_3, 15)[0.5, 0.5]$ from $R_3$ to $R_3 \setminus \{(5,3)\}$ (or in fact, to any $r \subset \mathcal{S}$ that does not contain $(5,3)$), we find that the revision is still inconsistent with the revision atom. Thus there is no spatial revision in this case.

This theorem and the example show that spatial revision is brittle and that, unless the knowledge base satisfies some fairly restrictive requirements, we cannot always revise KBs by using spatial revision alone. In KBs like that from Example 8, one inserted PST atom is capable of causing such a severe conflict between PST atoms in the KB that even drastic changes in every PST atom's region does not lead to a possible resolution of the inconsistency. We take this result as evidence that spatial revision should not be relied on by itself as a revision technique for PST knowledge bases.

### *3.6. Minimizing temporal change*

In this section, we study what happens when we revise a PST KB by changing only time stamps. We will consider both the infinite case where $T$ is the set of all non-negative integers (this is how $T$ was defined) as well as where $T = \{0, \dots, N\}$ for some positive integer $N$, making $T$ finite. A *temporal revision* of PST atom $a = loc(id, r, t)[\ell, u]$ is a PST atom of the form $a' = loc(id, r, t')[\ell, u]$. We define the distance between them as $d_T(a, a') = |t - t'|$.

A temporal revision of $\mathcal{K}$ is a PST KB $\mathcal{K}'$ containing at most one temporal revision of each atom in $\mathcal{K}$. The distance between $\mathcal{K}$ and $\mathcal{K}'$, $(d_T(\mathcal{K}, \mathcal{K}'))$, is the sum of the distances between the individual atoms and their associated temporal revisions.

**Definition 13.** A temporal revision $\mathcal{K}'$ of $\mathcal{K}$ by adding *ra* is *optimal* iff $\mathcal{K}' \cup \{ra\}$ is consistent and there is no other temporal revision $\mathcal{K}''$ of $\mathcal{K}$ by adding *ra* such that $\mathcal{K}'' \cup \{ra\}$ is consistent and $d_T(\mathcal{K}, \mathcal{K}'') < d_T(\mathcal{K}, \mathcal{K}')$.

As in the case of the previous two revision strategies, there can be multiple optimal temporal revisions — we assume the existence of a strict total ordering $O_T$ to achieve uniqueness. We denote this optimal temporal revision of $\mathcal{K}$ by adding *ra* as $\mathcal{K} \dot{+}_t ra$.

**Example 9.** Again, we consider the revision from Example 6 where the knowledge base $\mathcal{K}_{Beijing}$ is being revised by the atom: $loc(worker, R_4, 15)[0.5, 0.5]$. In this example, however, we suppose the error creating the inconsistency arises due to temporal inaccuracies in our knowledge base, implying that the inconsistency can be fixed by changing the timestamps in the knowledge base. The relevant atoms from $\mathcal{K}_{Beijing}$ are:

$$\{loc(worker, R_3, 15)[0.5, 0.5], loc(worker, R_2, 15)[0.5, 0.5]\}.$$

However, since it takes the worker at least 2 time periods to travel from $R_4$ to $R_2$ (recall the worker can travel Manhattan distance two each time step), and it only takes the worker 1 time period to travel to $R_3$ from $R_4$, the revisions changing $loc(worker, R_3, 15)[0.5, 0.5]$ to either $loc(worker, R_3, 14)[0.5, 0.5]$, or $loc(worker, R_3, 16)[0.5, 0.5]$ will both be consistent with $loc(worker, R_4, 15)[0.5, 0.5]$ and will be optimal temporal revisions (we could also change $loc(worker, R_2, 15)[0.5, 0.5]$ to time points 13 or 17 to achieve consistency, but the resulting distance $d_T$ would be larger than the above revisions and therefore not optimal). If $O_T$ prefers knowledge bases with higher time points to knowledge bases with lower time points, then the optimal temporal revision, $\mathcal{K}_{Beijing} \dot{+}_t loc(worker, R_4, 15)[0.5, 0.5]$ will change $loc(worker, R_3, 15)[0.5, 0.5]$ to $loc(worker, R_3, 16)[0.5, 0.5]$.

Now we show by examples that temporal revisions need not exist when either $T$ is finite (Example 10) or when the reachability definition is not fully connected (Example 11).

**Example 10.** For $T = \{0, 1\}$, let

$$\mathcal{K} = \{loc(id, \{p_1\}, 0)[0.6, 0.6], loc(id, \{p_2\}, 1)[0.6, 0.6]\}$$

and let the revision atom be $ra = loc(id, \{p_1\}, 0)[0.7, 0.7]$. All possible temporal revisions of $\mathcal{K}$ are inconsistent with *ra*.

**Example 11.** For infinite $T$ a two point space $\mathcal{S} = \{p_1, p_2\}$, and *RD* where $\{reachable(p, p) \mid p \in \mathcal{S}\} = RD$, let $\mathcal{K} = \{loc(id, \{p_1\}, 0)[1,1]\}$ and $ra = loc(id, \{p_2\}, 1)[1,1]$. No matter what integer $t \in T$ is used for the temporal revision: $\mathcal{K}' = \{loc(id, \{p_1\}, t)[1,1]\}$, $\mathcal{K}'$ will still be inconsistent with *ra* because *ra* requires that the object always be at $p_2$ while $\mathcal{K}'$ requires that the object always be at $p_1$.

We can guarantee the existence of a temporal revision if we eliminate the causes of these two counterexamples.

**Proposition 3.** *If $T$ is infinite and RD is fully connected, then there is an AGM-compliant minimal temporal revision $\mathcal{K}' = \mathcal{K} \dotplus_t ra$.*

The basic idea involves changing the time points for all atoms in $\mathcal{K}$ such that the difference in time between any two atoms is sufficient for the object to travel form any point in space to any point in space.

When a temporal revision exists, the following result shows that when $T$ is finite deciding if a temporal revision is optimal is coNP-complete. From a practical point of view, there is no loss of utility in the finiteness assumption as the size of $T$ can be an arbitrarily large finite number.

**Theorem 7.** *Given a PST atom ra, and PST KBs $\mathcal{K}$ and $\mathcal{K}'$, deciding whether $\mathcal{K}'$ is a temporally optimal revision of $\mathcal{K}$ by ra is coNP-hard. Further, if $T$ is finite then deciding whether $\mathcal{K}'$ is a temporally optimal revision of $\mathcal{K}$ by adding ra is coNP-complete.*

The proof proceeds by reduction to the MCSS problem (Definition 29 in Appendix A). The following corollary of Theorem 7 states that checking whether a PST KB is *the* optimal temporal revision of a given knowledge base is coNP-complete even when we *take the order $O_T$ into account*.

**Corollary 1.** *Given a PST atom ra, and PST KBs $\mathcal{K}$, $\mathcal{K}'$ where $T$ is finite, checking whether $\mathcal{K}' = \mathcal{K} \dotplus_t ra$ is coNP-complete.*

We now introduce Algorithm 2 to compute temporal revisions. This algorithm works via *unary* temporal revisions where $loc(id, r, t')[\ell, u]$ is a unary temporal revision of $loc(id, r, t)[\ell, u]$ iff $abs(t - t') = 1$. The algorithm creates a search tree — each node $N$ in the search tree has an $N.KB$ field. The root of the search tree is initialized to $Root.KB = \mathcal{K}$. Every child $C$ of a node $N$ is just like $N$ except that exactly one PST atom in $N.KB$ is replaced by a unary temporal revision. Further, each child $KB$ is required to be further (according to $d_T$) from $\mathcal{K}$ than its parent. When visiting a node $N$, the algorithm checks if $N.KB \cup \{ra\}$ is consistent. By creating and visiting this tree in breadth first order, we are guaranteed that the first node that satisfies this consistency check is an optimal temporal revision of $\mathcal{K}$ that accomplishes the insertion of *ra*.

**Algorithm 2.** ***TemporalRevision***$(\mathcal{K}, ra)$ Find $\mathcal{K} \dotplus_t ra$.

**If** $\{a\}$ is inconsistent, **return** error.
Get new node *Root*. Set $Root.KB = \mathcal{K}$;
TODO = [Root]. {TODO is an ordered list.}
**while** True **do**
  Let nextTODO be an empty list.
  {iterate over TODO in order.}
  **for** N in TODO **do**
    **if** $N.KB \cup \{ra\}$ is consistent **return** $N.KB \cup \{ra\}$.
    Insert each child of $N$ into nextTODO.
  **end for**
  Let TODO = nextTODO.
  sort TODO according to strict total ordering $O_T$.
**end while**

**Theorem 8.** *If there exists a minimal temporal revision for $\mathcal{K}$ with respect to ra, then Algorithm 2 is correct, i.e.* ***TemporalRevision***$(\mathcal{K}, ra)$ *returns $\mathcal{K} \dotplus_t ra$. Moreover,* ***TemporalRevision***$(\mathcal{K}, a)$ *is AGM-compliant.*

The ***TemporalRevision*** algorithm takes time exponential in the size of the knowledge base (as expected due to Theorem 7).

### 3.7. Minimizing object Id change

In this section, we introduce a revision operator that changes the object ids of PST atoms. IDs of moving objects can be incorrect for any number of reasons. For example, if the data was inserted manually then an id might be erroneous because of a typing error. Alternatively, if the ids were detected using an image processing program which may be used in some traffic cameras,[7] errors might occur because of occlusion when the image is captured, or due to poor lighting conditions,

[7] Many cities in the US and Europe have traffic cameras to track offenders. Some of these use automated methods to read license plates, while others just register an image of the license plate that is then manually inserted into the database by someone who looks at the image.

or wind conditions (e.g. leaves blowing across the field of view of the camera), or just error in the image recognition algorithms.

In order to evaluate the distance between a KB and its revisions, we assume the existence of a metric $d_O(id_1, id_2)$ whose value is the cost of changing an object id from $id_1$ to $id_2$. For example, this function could use the edit-distance cost of changing the name of the object identified by $id_1$ into the name of the object identified by $id_2$. Edit distance might work well for identifying typographical errors. Versions of edit distance that take proximity on a typewriter keyboard are also available.

An *object id revision* of PST atom $a = loc(id, r, t)[\ell, u]$ is an atom of the form $a' = loc(id', r, t)[\ell, u]$. The distance $d_O(a, a')$ is given by $d_O(id, id')$.

An object id revision of $\mathcal{K}$ is a $\mathcal{K}'$ containing at most one object id revision of each atom in $\mathcal{K}$. The distance between a PST KB and its object id revision ($d_O(\mathcal{K}, \mathcal{K}')$) is the sum of the distances between the individual atoms and their associated object id revisions.

**Definition 14.** An object id revision $\mathcal{K}'$ of $\mathcal{K}$ by adding *ra* is *optimal* iff $\mathcal{K}' \cup \{ra\}$ is consistent and there is no other object id revision $\mathcal{K}''$ of $\mathcal{K}$ by adding *ra* such that $\mathcal{K}'' \cup \{ra\}$ is consistent and $d_O(\mathcal{K}, \mathcal{K}'') < d_O(\mathcal{K}, \mathcal{K}')$.

As in the case of the previous revision strategies, there can be multiple optimal revisions — we assume the existence of a strict total ordering $O_T$ to obtain uniqueness. We denote this optimal object id revision of $\mathcal{K}$ by adding *ra* as $\mathcal{K} \dot{+}_O ra$. We now give an example of object ID change.

**Example 12.** Again we use the knowledge base $\mathcal{K}_{Beijing}$ from Example 1 and the revision atom specifying the worker is in the park with 50% probability at time 15: $loc(worker, R_4, 15)[0.5, 0.5]$. Recall that the revision is inconsistent with $\mathcal{K}_{Beijing}$ due to the following atoms:

$$loc(worker, R_3, 15)[0.5, 0.5],\ loc(worker, R_2, 15)[0.5, 0.5],$$

which together specify that the worker could not be in $R_4$ at time 15 with any probability. Either of those atoms could be mistaken — maybe it is the worker's brother who is seen in $R_3$ at time 15, and due to familial resemblance is mistaken for the worker, or maybe it is a different worker that shows up to work in $R_2$. This suggests two possible object id revisions:

- $\mathcal{K}_B^1 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_3, 15)[0.5, 0.5]\} \cup \{loc(brother, R_3, 15)[0.5, 0.5]\}$,
- $\mathcal{K}_B^2 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_2, 15)[0.5, 0.5]\} \cup \{loc(worker_2, R_2, 15)[0.5, 0.5]\}$.

We will assume these two of the many possible revisions to have the same (optimal) distance according to $d_O$, i.e. $d_O(\mathcal{K}_{Beijing}, \mathcal{K}_B^1) = d_O(\mathcal{K}_{Beijing}, \mathcal{K}_B^2)$ and therefore that $O_T$ will be needed to decide which is $\mathcal{K}_{Beijing} \dot{+}_O loc(worker, R_4, 15)[0.5, 0.5]$.

However, it turns out that object ID change is again not always possible. The next example shows that if *ID* is finite and determined *a priori*, there need not always be an object id revision.

**Example 13.** Let $ID = \{id_1, \dots, id_n\}$ and $r \subsetneq \mathcal{S}$. Let $\mathcal{K} = \{a_1, \dots, a_n\}$ where $a_i = loc(id_i, r, 0)[\frac{1}{i}, \frac{1}{i}]$ for $1 \leqslant i \leqslant n$. Let $ra = loc(id_1, r, 0)[0, 0]$. No object id revision of $\mathcal{K}$ is consistent with *ra*, so Axiom (A5) is not satisfied.

Even so, when there is an object ID change, it is hard to find. The following theorem states that checking for optimal object id revisions of a given knowledge base is coNP-complete.

**Theorem 9.** *Given a PST atom ra, and PST KBs $\mathcal{K}$, $\mathcal{K}'$ where $T$ is finite, deciding whether $\mathcal{K}'$ is an optimal object id revision of $\mathcal{K}$ is coNP-complete.*

As with other coNP-complete proofs in this paper, the MCSS problem (Definition 29 in Appendix A) is used in the reduction. The full proof is given in Appendix B. A corollary states that checking whether a PST KB is *the* optimal object id revision of a given knowledge base is also coNP-complete (here the order $O_T$ is considered).

**Corollary 2.** *Given PST atom ra, and PST KBs $\mathcal{K}$, $\mathcal{K}'$ where $T$ is finite, deciding whether $\mathcal{K}' = \mathcal{K} \dot{+}_o ra$ is coNP-complete.*

### *3.8. Minimizing probability change*

In this section, we propose a belief revision operator that replaces PST atoms of the form $loc(id, r, t)[\ell, u]$ in $\mathcal{K}$ by PST atoms $loc(id, r, t)[\ell', u']$ where $[\ell, u] \subseteq [\ell', u']$. In other words, these belief revision operators expand the probability bounds of PST atoms in $\mathcal{K}$ in order to retain consistency when *ra* is added.

First we examine changing only the lower or upper bound (i.e. changing $\ell$ to $\ell'$ or $u$ to $u'$ respectively), then we consider the possibility of changing both. In all cases we want to minimize the expansion of the probability interval $[\ell, u]$ to $[\ell', u']$.

*3.8.1. Minimizing lower/upper bound change*

One may wish to modify a PST KB by changing only the lower (resp. upper) bounds of the PST atoms. We show here how to do this for lower bounds. Doing upper bound change is analogous.

**Definition 15.** Suppose $a = loc(id, r, t)[\ell, u]$ is a PST atom and $\ell' \leqslant \ell$. Then the PST atom $a' = loc(id, r, t)[\ell', u]$ is called a *lower bound revision* of $a$. The distance, $d_\ell(a, a')$ between $a$ and $a'$ is defined as $(\ell - \ell')$.

A lower bound revision of $\mathcal{K}$ is a $\mathcal{K}'$ containing at most one lower bound revision of each atom in $\mathcal{K}$. The distance between $\mathcal{K}$ and $\mathcal{K}'$, $(d_\ell(\mathcal{K}, \mathcal{K}'))$, is the sum of the distances between the individual atoms and their associated lower bound revision.

**Definition 16.** A lower bound revision $\mathcal{K}'$ of $\mathcal{K}$ by adding $ra$ is *optimal* iff $\mathcal{K}' \cup \{ra\}$ is consistent and there is no other lower bound revision $\mathcal{K}''$ of $\mathcal{K}$ by adding $ra$ such that $\mathcal{K}'' \cup \{ra\}$ is consistent and $d_\ell(\mathcal{K}, \mathcal{K}'') < d_\ell(\mathcal{K}, \mathcal{K}')$.

As in the case of the previous revision strategies, there can be multiple optimal lower bound revisions and we assume the existence of a strict total ordering $O_T$ to obtain uniqueness. We denote this minimal optimal lower bound revision of $\mathcal{K}$ by adding $ra$ as $\mathcal{K} \dotplus_\ell ra$.

The following theorem characterizes the cases where a lower bound revision is possible.

**Theorem 10.** *Let $\mathcal{K}$ be any knowledge base. Lower bound revision of $\mathcal{K}$ for any atom ra is AGM-compliant iff for all $a_i \in \mathcal{K}$, $u_i = 1$.*

The following is a particular example where no lower bound revision exists.

**Example 14.** For $\mathcal{K}_{Beijing}$ from Example 1 let $ra = loc(worker, (7, 6), 15)[1, 1]$. There is no lower bound revision of $\mathcal{K}_{Beijing}$ consistent with $ra$.

It is therefore impossible to guarantee the existence of an AGM-compliant revision when using lower bound revision. An analogous result holds for upper bound revision.

*3.8.2. Minimizing probability interval change*

In this subsection we examine changing both the lower and the upper bounds of the probability intervals in PST atoms. We will see that unlike attempts to change either of the bounds individually, revision strategies changing both bounds can be guaranteed to exist. Further, we will present a polynomial time algorithm for making AGM-compliant revisions by changing the probability intervals.

Our definition of probability interval revision combines both lower bound and upper bound revisions.

**Definition 17.** Suppose $a = loc(id, r, t)[\ell, u]$ is a PST atom, $u \leqslant u'$, and $\ell' \leqslant \ell$. The PST atom $a' = loc(id, r, t)[\ell', u']$ is called a *probabilistic revision* of $a$. The distance, $d_P(a, a')$ between $a$ and $a'$ is defined as $(\ell - \ell') + (u' - u)$.

A probabilistic revision of $\mathcal{K}$ is a $\mathcal{K}'$ containing at most one probabilistic revision of each atom in $\mathcal{K}$. The distance between $\mathcal{K}$ and $\mathcal{K}'$, $(d_P(\mathcal{K}, \mathcal{K}'))$, is the sum of the distances between the individual atoms and their associated probabilistic revision.

**Definition 18.** A probabilistic revision $\mathcal{K}'$ of $\mathcal{K}$ by adding $ra$ is *optimal* iff $\mathcal{K}' \cup \{ra\}$ is consistent and there is no other probabilistic revision $\mathcal{K}''$ of $\mathcal{K}$ by adding $ra$ such that $\mathcal{K}'' \cup \{ra\}$ is consistent and $d_P(\mathcal{K}, \mathcal{K}'') < d_P(\mathcal{K}, \mathcal{K}')$.

As in the case of the previous revision strategies, there can be multiple optimal probabilistic revisions and we assume the existence of a strict total ordering $O_T$ to get uniqueness. We denote this optimal probabilistic revision of $\mathcal{K}$ with respect to atom $ra$ as $\mathcal{K} \dotplus_p ra$. We now give an example of probabilistic revision.

**Example 15.** Consider again $\mathcal{K}_{Beijing}$ from Example 1 and a revision atom specifying the worker in the park with 50% probability at time 15: $loc(worker, R_4, 15)[0.5, 0.5]$. This revision atom is inconsistent with $\mathcal{K}_{Beijing}$ due to the subset:

$$\{loc(worker, R_3, 15)[0.5, 0.5], loc(worker, R_2, 15)[0.5, 0.5]\}.$$

The probability intervals for either of those atoms could be wrong, and in probabilistic revision, we modify those bounds to create consistency. This suggests three of many possible probability interval revisions:

- Suppose that $loc(worker, R_3, 15)[0.5, 0.5]$ was mistaken:
  $\mathcal{K}_B^1 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_3, 15)[0.5, 0.5]\} \cup \{loc(worker, R_3, 15)[0, 0.5]\}$.

- Suppose that $loc(worker, R_2, 15)[0.5, 0.5]$ was mistaken:
  $\mathcal{K}_B^2 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_2, 15)[0.5, 0.5]\} \cup \{loc(worker, R_2, 15)[0, 0.5]\}$.
- Suppose both $loc(worker, R_3, 15)[0.5, 0.5]$ and $loc(worker, R_2, 15)[0.5, 0.5]$ were partially mistaken:
  $\mathcal{K}_B^3 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_3, 15)[0.5, 0.5], loc(worker, R_2, 15)[0.5, 0.5]\} \cup \{loc(worker, R_3, 15)[0.25, 0.5], loc(worker, R_2, 15)[0.25, 0.5]\}$.

(of course, any change to the lower bounds so that the lower bounds of all three atoms adds to 1 would work). Note that all mentioned revisions have a distance of 0.5 from $\mathcal{K}_{Beijing}$ (i.e. $d_P(\mathcal{K}_{Beijing}, \mathcal{K}_B^1) = d_P(\mathcal{K}_{Beijing}, \mathcal{K}_B^2) = d_P(\mathcal{K}_{Beijing}, \mathcal{K}_B^3)$). Thus $O_T$ will be needed to distinguish between them. If we suppose that $O_T$ prefers not to change atoms associated with $R_2$, then $\mathcal{K}_{Beijing} \dotplus_P loc(worker, R_4, 15)[0.5, 0.5]$ will be $\mathcal{K}_B^1 \cup \{loc(worker, R_4, 15)[0.5, 0.5]\}$.

We can find an optimal probabilistic revision by setting up a linear program similar to $LP(\mathcal{K})$ (Definition 8). We again use the variables $v_{id,t,p,q}$, each representing the probability of an object *id* being at location $p$ at time $t$ and at location $q$ at time $t+1$. We limit the range of *id* to those objects mentioned in the database and the range of $t$ to the bounded set $T$ provided *a priori* (we assume a bounded set of time points $T$ for probabilistic revision). For each PST atom $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i]$ in $\mathcal{K}$, we also include variables $low_i$ and $up_i$ representing the atoms' modified lower and upper bounds.

**Definition 19** *(Probability Revision Linear Program (PRLP)).* Let $PRLP(\mathcal{K}, ra)$ contain the following constraints:

1. For each $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}$:
   (a) $0 \leqslant (\sum_{p\in r_i} \sum_{q\in\mathcal{S}} v_{id_i,t_i,p,q}) - low_i$.
   (b) $0 \geqslant (\sum_{p\in r_i} \sum_{q\in\mathcal{S}} v_{id_i,t_i,p,q}) - up_i$.
   (c) $\ell_i \geqslant low_i$, $low_i \geqslant 0$, $u_i \leqslant up_i$, and $up_i \leqslant 1$.
2. For $ra = loc(id', r', t')[\ell, u]$:
   (a) $\ell \leqslant \sum_{p\in r'} \sum_{q\in\mathcal{S}} v_{id',t',p,q}$ and $u \geqslant \sum_{p\in r'} \sum_{q\in\mathcal{S}} v_{id',t',p,q}$.
3. For each *id* in $\mathcal{K}$ and each $t$ in $T$:
   (a) For all $p, q \in \mathcal{S}$, $v_{id,t,p,q} \geqslant 0$.
   (b) $\sum_{p\in\mathcal{S}} \sum_{q\in\mathcal{S}} v_{id,t,p,q} = 1$.
   (c) For all $p, q \in \mathcal{S}$, if $reachable_{id}(p, q) \notin RD$: $v_{id,t,p,q} = 0$.
   (d) For all $p \in \mathcal{S}$: $\sum_{q\in\mathcal{S}} v_{id,t,q,p} = \sum_{q\in\mathcal{S}} v_{id,t+1,p,q}$.

We now compute an optimal revision of $\mathcal{K}$ by minimizing the distance function subject to $PRLP(\mathcal{K}, ra)$. As in the case of all our revision strategies, when there are multiple solutions to this linear program, we assume there is a mechanism to deterministically pick one. We are now able to define a probabilistic revision strategy.

**Definition 20** *(Probabilistic Revision).* Suppose $\mathcal{K}$ is a PST KB and *ra* a PST atom. Let $\theta$ be a (deterministically) selected solution of the linear program

$$\textbf{minimize} \sum_{a_i\in\mathcal{K}} \big((\ell_i - low_i) + (up_i - u_i)\big) \quad \textbf{subject to } PRLP(\mathcal{K}, ra).$$

Return the PST KB defined as

$$\big\{loc(id_i, r_i, t_i)[low_i\theta, up_i\theta] \mid loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}\big\} \cup \{ra\}.$$

The size of $PRLP(\mathcal{K})$ has the same big-O bound as $LP(\mathcal{K})$: $O((|\mathcal{K}| \cdot |\mathcal{S}|^2 \cdot n_t)^2)$ (the only difference is the addition of $2 \cdot |\mathcal{K}|$ extra variables for the $low_i$ and $up_i$). Since solving linear programs is also polynomial [15], and we can assume our mechanism for picking a solution deterministically runs in polynomial time[8]; hence the above procedure computes $\mathcal{K} \dotplus_p ra$ in polynomial time.

This polynomial time probabilistic revision strategy is also AGM-compliant.

**Proposition 4.** *$\mathcal{K} \dotplus_p ra$ satisfies* (A1)–(A6).

## 4. Hybrid belief revision strategy

The previous section dealt with the revision of a PST KB by deleting atoms or modifying a single component of atoms. Here we consider combinations of these strategies that we call hybrid belief revision strategies. In the first subsection we allow only modifications of atoms; the second subsection allows deletions of atoms as well.

[8] Such mechanisms exist: consider a strict total ordering over the variables which specifies the order with which the linear program solver should minimize variables.

### 4.1. Minimizing weighted hybrid change

In this subsection, we consider a technique which allows all parts of an atom to be revised. A hybrid revision of a PST atom $a = loc(id, r, t)[\ell, u]$ is a PST atom $a' = loc(id', r', t')[\ell', u']$. The distance between $a$ and $a'$ is a weighted sum of the distances already defined:

$$d_H(a, a') = w_O \cdot d_O(a, a') + w_S \cdot d_S(a, a') + w_T \cdot d_T(a, a') + w_P \cdot d_P(a, a').$$

Here $d_O$, $d_S$, $d_T$, $d_P$ are the distance functions between atoms defined in the last section that quantify distances between objects, regions, time points, and probability intervals respectively. Likewise, $w_O$, $w_S$, $w_T$, and $w_P$ are weights, specifying the relative importance of the associated type of change. The weighted hybrid distance function uses the distances between two atoms along each of these four parameters to compute a distance between them.

The weights are non-zero, non-negative, and allow a special value $\infty$. Assigning a weight $\infty$ to any $w$ is equivalent to forcing no change of the appropriate type because any such change would make the distance between two PST atoms infinite. (We assume $0 \times \infty = 0$, to allow other types of changes.)

**Definition 21.** Any PST atom $a' = loc(id', r', t')[\ell', u']$ can be considered to be a *weighted hybrid revision* of any PST atom $a = loc(id, r, t)[\ell, u]$. The distance between $a$ and $a'$ is defined to be $d_H(a, a')$.

For $\mathcal{K}$ and $\mathcal{K}'$, $\mathcal{K}'$ is a *weighted hybrid revision* of $\mathcal{K}$ iff there is a bijection $\beta$ from $\mathcal{K}$ to $\mathcal{K}'$ such that for all $a \in \mathcal{K}$, $\beta(a)$ is a hybrid revision of $a$. The hybrid revision distance between $\mathcal{K}$ and $\mathcal{K}'$ is defined to be the minimum distance of all the bijections between them or

$$d_H(\mathcal{K}, \mathcal{K}') = \min_{\beta:\, \mathcal{K} \leftrightarrow \mathcal{K}'} \left( \sum_{a \in \mathcal{K}} d_H(a, \beta(a)) \right).$$

We say that $\mathcal{K}'$ is an optimal weighted hybrid revision of $\mathcal{K}$ by adding $ra$ iff $\mathcal{K}' \cup \{ra\}$ is consistent and there is no $\mathcal{K}''$ such that $\mathcal{K}'' \cup \{ra\}$ is consistent and $d_H(\mathcal{K}, \mathcal{K}') > d_H(\mathcal{K}, \mathcal{K}'')$.

Again there can be multiple optimal weighted hybrid revisions for a given KB and revision atom. We use the strict total ordering $O_T$ to obtain uniqueness. The optimal weighted hybrid revision $\mathcal{K}'$ that is minimal according to $O_T$ is *the* weighted hybrid revision returned by the $\dot{+}_h$ operator. That is, $\mathcal{K} \dot{+}_h ra = \mathcal{K}' \cup \{ra\}$ where $\mathcal{K}'$ is the $O_T$-minimal optimal weighted hybrid revision. We now give an example of hybrid revision relating to the knowledge base from Example 1.

**Example 16.** Consider revising $\mathcal{K}_{Beijing}$ from Example 1 with the revision atom $loc(worker, R_4, 15)[0.6, 0.6]$. This revision atom is inconsistent with $\mathcal{K}_{Beijing}$ due to the subset: $\{loc(worker, R_3, 15)[0.5, 0.5], loc(worker, R_2, 15)[0.5, 0.5]\}$. In hybrid revision, we are able to change the objects, regions, time points, and probabilities. We now list several possible weighted hybrid revisions.

- If the time and identity of $loc(worker, R_3, 15)[0.5, 0.5]$ is off: $\mathcal{K}_B^1 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_3, 15)[0.5, 0.5]\} \cup \{loc(brother, R_3, 16)[0.5, 0.5]\}$.
- If the probabilities for both $loc(worker, R_3, 15)[0.5, 0.5]$ and $loc(worker, R_2, 15)[0.5, 0.5]$ were partially mistaken: $\mathcal{K}_B^2 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_3, 15)[0.5, 0.5], loc(worker, R_2, 15)[0.5, 0.5]\} \cup \{loc(worker, R_3, 15)[0.2, 0.5], loc(worker, R_2, 15)[0.2, 1]\}$.
- If the region and probabilities of $loc(worker, R_3, 15)[0.5, 0.5]$ are wrong: $\mathcal{K}_B^3 = \mathcal{K}_{Beijing} \setminus \{loc(worker, R_3, 15)[0.5, 0.5]\} \cup \{loc(worker, R_3 \cup \{(7, 2)\}, 15)[0.5, 0.6]\}$.

Note that all of these are valid revisions as they are all consistent with $loc(worker, R_4, 15)[0.6, 0.6]$. Depending on the weights in $d_H$, any one of these could be an optimal hybrid revision. If we suppose $w_O = w_T = 10$, $w_S = 0.1$ and $w_P = 1$, then $\mathcal{K}_B^3$ will be an optimal hybrid revision (since $d_H(\mathcal{K}, \mathcal{K}_B^1) > 10$, $d_H(\mathcal{K}, \mathcal{K}_B^2) = 0.9$, and $d_H(\mathcal{K}, \mathcal{K}_B^3) = 0.2$). In this case there are no other weighted hybrid revisions with distance of 0.2 or less from $\mathcal{K}$, so $\mathcal{K} \dot{+}_h loc(worker, R_4, 15)[0.5, 0.5] = \mathcal{K}_B^3$.

Weighted hybrid revision subsumes object, spatial, temporal or probabilistic revision: in fact, to obtain any combination, a user need only set all weights not associated with that type of revision in $d_H$ to $\infty$. Therefore, as a corollary to the fact that spatial revision is not possible in the general case (Section 3.5), we can state the following result.

**Proposition 5.** *Weighted hybrid revision cannot be guaranteed to satisfy all our AGM-style axioms.*

The following result derives the computational complexity of checking whether a given knowledge-base $\mathcal{K}'$ is an optimal weighted hybrid revision of $\mathcal{K}$.

**Theorem 11.** *Given a PST atom ra, and PST KBs $\mathcal{K}$, $\mathcal{K}'$ where $T$ is finite, deciding whether $\mathcal{K}' \cup \{ra\} = \mathcal{K} \dot{+}_h ra$ is coNP-complete.*

The reduction uses object change (though could use any of the coNP-hard types of change introduced in this paper).

Next we show how to accomplish hybrid revision by an appropriate *mixed* integer linear program (MILP) in a provably correct way. In order to achieve this, we extend and modify the definition of *PRLP* (Definition 19) by adding spatial, temporal, and object id integer valued variables that will co-exist with the real valued variables in the definition of *PRLP* as follows.

For each $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}$, let $region(a_i)$ be a set of regions that may replace $r_i$. In general, $region(a_i)$ consists of all possible regions belonging to $\mathcal{S}$ but one could consider an appropriate subset (we assume that $r_i \in region(a_i)$). For each region $r_j \in region(a_i)$, we use an *integer valued* variable $\hat{r}^i_j$ which is intended to be set to 1 if region $r_j$ is the revised region for $a_i$ and 0 otherwise. We thus limit the values of all $\hat{r}^i_j$ to $\{0, 1\}$ and add the constraint $\sum_j \hat{r}^i_j = 1$. This constraint indicates that exactly one such region can have a value set to 1. Temporal variables $\hat{t}^i_j$ and object id variables $\hat{id}^i_j$ are added in a similar way.

As in PRLP, we have the real-valued variables $low_i$ and $up_i$ limited to the $[0, 1]$ interval representing the revised lower and upper bounds for atom $a_i$. We continue the use of $v_{id,t,p,q}$ to represent the probability of object $id$ being at location $p$ at time $t$ and at location $q$ at time $t+1$.

We now formally introduce the following set of integer linear constraints.

**Definition 22** *(Hybrid Revision Linear Program (HRLP)).* For knowledge base $\mathcal{K} = \{a_i = loc(id_i, r_i, t_i)[\ell_i, u_i]\}$, and revision atom $ra = loc(id_r, r_r, t_r)[\ell_r, u_r]$, let $HRLP(\mathcal{K}, ra)$ be the following set of constraints:

(1) $\ell_r \leqslant \sum_{p \in r_r} \sum_{q \in \mathcal{S}} v_{id_r,t_r,p,q} \leqslant u_r$ (enforce proper revision).
(2) $\forall a_i = loc(id_i, r_i, t_i)[\ell, u] \in \mathcal{K}$, $\forall id_j \in ID$, $\forall t_k \in T$, and $\forall r_m \in regions(a_i)$,

$$low_i - \big(1 - \hat{id}^i_j\big) - \big(1 - \hat{t}^i_k\big) - \big(1 - \hat{r}^i_m\big) \leqslant \sum_{p \in r_m} \sum_{q \in \mathcal{S}} v_{id_j,t_k,p,q} \leqslant up_i + \big(1 - \hat{id}^i_j\big) + \big(1 - \hat{t}^i_k\big) + \big(1 - \hat{r}^i_m\big).$$

(3) For all $a_i \in \mathcal{K}$: $\sum_{id_j \in ID} \hat{id}^i_j = 1$, $\sum_{t_j \in T} \hat{t}^i_j = 1$, $\sum_{r_j \in regions(a_i)} \hat{r}^i_j = 1$.
(4) For all $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}$: $low_i \leqslant \ell_i$, $up_i \geqslant u_i$.
(5) For all $id \in ID$, $t, t+1 \in T$, and $p \in \mathcal{S}$: $\sum_{q \in \mathcal{S}} v_{id,t,q,p} = \sum_{q \in \mathcal{S}} v_{id,t+1,p,q}$.
(6) For all $id \in ID$, $t \in T$, and $p, q \in \mathcal{S}$: $v_{id,t,p,q} = 0$ if $reachable_{id}(p, q) \notin RD$.
(7) For all $id \in ID$, and $t \in T$: $\sum_{p \in \mathcal{S}} \sum_{q \in \mathcal{S}} v_{id,t,p,q} = 1$.

Given a solution to $HRLP(\mathcal{K}, ra)$, we can construct a bijection $\beta$ such that $\beta(\mathcal{K})$ is a hybrid revision of $\mathcal{K}$ by adding $ra$ as follows. Let $\theta$ be a solution to $HRLP(\mathcal{K}, ra)$. Define

$$\beta_\theta(\mathcal{K}) = \big\{loc(id_j, r_m, t_k)[low_i\theta, up_i\theta] \mid a_i \in \mathcal{K} \wedge \hat{id}^i_j\theta = 1 \wedge \hat{r}^i_m\theta = 1 \wedge \hat{t}^i_k\theta = 1\big\}.$$

Note that because $\hat{id}^i_j$, $\hat{r}^i_m$, and $\hat{t}^i_k$ are all either zero or one, constraint (3) of *HRLP* ensures that $\beta_\theta(\mathcal{K})$ is well-defined.

We now show that the problem of finding an optimal weighted hybrid revision corresponds to solving a mixed integer linear program that has $HRLP(\mathcal{K}, ra)$ as its set of constraints.

**Theorem 12.** *Let $\mathcal{K} = \{a_i \mid a_i = loc(id_i, r_i, t_i)[\ell_i, u_i]\}$, and $ra = loc(id_r, r_r, t_r)[\ell_r, u_r]$, the revision atom. $\theta$ is an optimal solution of the optimization problem*

***minimize the objective function***

$$w_O\Bigg(\sum_{a_i \in \mathcal{K}} \sum_{id_j \in ID} \hat{id}^i_j \cdot d_O(id_i, id_j)\Bigg) + w_S\Bigg(\sum_{a_i \in \mathcal{K}} \sum_{r_j \in regions(a_i)} \hat{r}^i_j \cdot d_S(r_i, r_j)\Bigg)$$
$$+ w_T\Bigg(\sum_{a_i \in \mathcal{K}} \sum_{t_j \in T} \hat{t}^i_j \cdot d_T(t_i, t_j)\Bigg) + w_P\Bigg(\sum_{a_i \in \mathcal{K}} (\ell_i - low_i) + (up_i - u_i)\Bigg)$$

***subject to***

$HRLP(\mathcal{K}, ra)$

*iff $\beta_\theta(\mathcal{K})$ is an optimal hybrid revision of $\mathcal{K}$ by adding ra.*

The number of constraints in $HRLP(\mathcal{K}, ra)$ is equal to the number of constraints in $PRLP(\mathcal{K}, ra)$ plus $O(|\mathcal{K}| \cdot |ID| \cdot |T| \cdot |2^{\mathcal{S}}|)$ new constraints due to items (2) and (3) in Definition 22. As the number of constraints in $PRLP(\mathcal{K}, ra)$ is $O(|\mathcal{K}| \cdot |T| \cdot |\mathcal{S}|^2)$, the total number of constraints in $HRLP(\mathcal{K}, ra)$ is $O(|\mathcal{K}| \cdot |ID| \cdot |T| \cdot |2^{\mathcal{S}}|)$. Since $|\mathcal{S}|$ is constant, this gives an upper bound of $O(|\mathcal{K}| \cdot |T| \cdot |ID|)$ constraints for $HRLP(\mathcal{K}, ra)$. It is worth noting that, although in the worst case the number of possible ways to replace each region $r_i$ of atom $a_i \in \mathcal{K}$ is bounded by the constant $|2^{\mathcal{S}}|$, this number could be appropriately bounded by a smaller constant in specific real-life applications by, for instance, only allowing changes to connected subsets of $\mathcal{S}$. The number of variables in *HRLP*($\mathcal{K}, ra$) is equal to the number of variables in *PRLP*($\mathcal{K}, ra$) (that is, $O(|\mathcal{K}| \cdot |T| \cdot |\mathcal{S}|^2)$) plus $O(|ID| \cdot |\mathcal{K}| + |T| \cdot |\mathcal{K}| + |2^{\mathcal{S}}| \cdot |\mathcal{K}|)$ new variables due to items (2) and (3) in Definition 22. As $|\mathcal{S}|$ is constant, this gives an upper bound of $O(|\mathcal{K}| \cdot (|ID| + |T|))$ variables for *HRLP*($\mathcal{K}, ra$). The size of *HRLP*($\mathcal{K}, ra$) is $O(|\mathcal{K}|^2 \cdot |ID|^2 \cdot |T|^2)$ (number of variables times number of constraints). Thus the size of *HRLP*($\mathcal{K}, ra$) is equivalent to the size of *PRLP*($\mathcal{K}, ra$) times $|ID|^2$. However, *HRLP*($\mathcal{K}, ra$) is a set of linear constraints over integer as well as real variables and therefore all known solution procedures run in exponential time, making it substantially more expensive to solve than an equivalently sized *PRLP*($\mathcal{K}, ra$), which has only real variables and is therefore solvable in polynomial time.

### *4.2. Prob-MAXCARD (PMC) hybrid revision*

In the previous section, we considered weighted hybrid revision which uses a linear combination of distances along each of four dimensions (object change, spatial change, temporal change, and probability interval change) to define an optimal revision. However, the max-subset and max-cardinality revision mechanisms are not considered there.

In this section we address a hybrid belief revision strategy that combines maximal cardinality subset and probability revision. According to this revision strategy, an atom may be either deleted from the knowledge base or have its probability interval changed. We call this Prob-MAXCARD (PMC) revision.

We will define a mixed integer linear program problem for knowledge base $\mathcal{K}$ and revision atom $ra$. Before doing this, we first need to introduce a set of mixed integer linear constraints.

**Definition 23** *(Max-Card and Probability Revision Program).* Let *MCPRP*($\mathcal{K}, ra$) be equal to *PRLP*($\mathcal{K}, ra$) except that inequalities 1(a) and 1(b) are replaced by the following:

1. For each $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}$:
   (a) $low_i - \delta_i \leqslant (\sum_{p \in r_i} \sum_{q \in \mathcal{S}} v_{id_i,t_i,p,q})$.
   (b) $(\sum_{p \in r_i} \sum_{q \in \mathcal{S}} v_{id_i,t_i,p,q}) \leqslant up_i + \delta_i$.
   (c) $\delta_i \in \{0, 1\}$.

Observe that, in *MCPRP*($\mathcal{K}, ra$), for each $a_i \in \mathcal{K}$, if $\delta_i = 0$ (1)(a) is equivalent to that of *PRLP*($\mathcal{K}, ra$). Otherwise, $\delta_i = 1$ and equation (1)(a) is trivially satisfied by any solution of *MCPRP*($\mathcal{K}, ra$), as if $a_i$ did not belong to $\mathcal{K}$.

Before formally definition PMC revision, we assume the existence of two vectors $w^m$ and $w^p$ of weights. The $i$'th component of vector $w^m$, namely $w_i^m$, indicates the importance of retaining $a_i$ in $\mathcal{K}$, while the $i$'th component of $w^p$, namely $w_i^p$, indicates the cost of modifying the probability interval of $a_i$.

**Definition 24** *(PROB-MAXCARD (PMC) Revision).* Suppose $\mathcal{K}$ is a PST KB and $ra$ is a PST atom. Let $\theta$ be a solution of the linear program

$$\textbf{minimize} \sum_{a_i \in \mathcal{K}} w_i^m \cdot \delta_i + \sum_{a_i \in \mathcal{K}} w_i^p \cdot \big((\ell_i - low_i) + (up_i - u_i)\big) \quad \textbf{subject to}\ MCPRP(\mathcal{K}, ra),$$

where $w^m$ and $w^p$ are vectors of weights, each of them having $|\mathcal{K}|$ strictly positive elements. Return the PST KB

$$\big\{loc(id_i, r_i, t_i)[low_i\theta, up_i\theta] \mid loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K} \wedge \delta_i\theta = 0\big\} \cup \{ra\}.$$

It is worth noting that, for each solution $\theta$ of the above mixed integer linear program and for each atom $a_i \in \mathcal{K}$, either $\delta_i\theta = 0$ or $low_i\theta = \ell_i$ and $up_i\theta = u_i$.

The vectors of weight $w^m$ and $w^p$ can be used to specify, for each atom, if either maximal-cardinality revision or probability revision is preferred. We can use these vectors to say that for a given subset of atoms of $\mathcal{K}$ we trust the probability values but a maximal-cardinality revision for this portion of the knowledge base is allowed. The uniqueness of the solution can be obtained as before by using the total ordering $O_T$.

## 5. Improvements and approximations of probabilistic revision

The decision problems associated with all revision strategies proposed thus far are intractable with the exception of the probabilistic revision proposed in Definition 20. As this is the only AGM-compliant revision strategy for PST KBs that has any chance of being practically useful, this section focuses on methods to make it more efficient.

### *5.1. Reducing the size of the linear program PRLP*

Probabilistic revision is computed by solving a linear program associated with the set of constraints *PRLP*($\mathcal{K}, ra$). It is well known that smaller linear programs are usually (but not always) more efficiently solvable than larger ones. In this subsection we eliminate both constraints and variables to generate a smaller linear program $PRLP^T$ by using the intuition that only time points explicitly mentioned in the knowledge base $\mathcal{K}$ need to be considered when constructing $PRLP(\mathcal{K}, ra)$. However, we need to be careful to make sure that the reachability constraints are still satisfied.

Let $T_\mathcal{K}$ be the set $\{t \mid loc(-,-,t)[\ell,u] \in \mathcal{K}\}$ of time points explicitly mentioned in $\mathcal{K}$. $T_\mathcal{K}$ is at most the size of $T$, and often smaller due to the fact that there is not always information for every possible time point.

There are several places in the definition of $PRLP(\mathcal{K}, ra)$ where the set $T$ of time points explicitly causes the incorporation of variables and constraints. Fortunately, we can simplify parts 3(c) and 3(d) of Definition 19 as follows.

1. We change the reachability constraints defined in the original *PRLP* as:

   $$\forall p, q \in \mathcal{S},\ t \in T,\ id \in ID \text{ if } reachable_{id}(p,q) \notin RD\colon\ v_{id,t,p,q} = 0.$$

   When $T_\mathcal{K} = \{t_0, \ldots, t_n\}$ with $t_j < t_{j+1}$, we are concerned with the object's ability to move from $p$ to $q$ in $t_{j+1} - t_j$ time points. An object can reach $q$ from $p$ in $k$ time points if there is a path from $p$ to $q$: $p_1, \ldots, p_k$ where $p_1 = p$, $p_k = q$, and for each $p_i, p_{i+1}$, the object can reach $p_{i+1}$ from $p_i$ ($reachable_{id}(p_i, p_{i+1})$). We say $reachable_{id}(p,q,t)$ iff such a path of length $t$ exists. We write *RDT* for the set of all $reachable_{id}(p,q,t)$ that can be generated from *RD*. Now, when using $T_\mathcal{K}$, we can replace the reachability constraints of the form given above with the simpler set of constraints

   $$\forall p, q \in \mathcal{S},\ t_i \in T_\mathcal{K},\ id \in ID \text{ if } reachable_{id}(p,q,t_{i+1} - t_i) \notin RDT\colon\ v_{id,t_i,p,q} = 0.$$

2. The second change involves the constraints ensuring that the probability of entering point $p$ equals the probability of exiting point $p$

   $$\sum_{q \in \mathcal{S}} v_{id,t,q,p} = \sum_{q \in \mathcal{S}} v_{id,t+1,p,q}.$$

   If $t + 1 \notin T_\mathcal{K}$ then we can replace such constraints in *PRLP* with constraints:

   $$\sum_{q \in \mathcal{S}} v_{id,t_i,q,p} = \sum_{q \in \mathcal{S}} v_{id,t_{i+1},p,q}.$$

Formally, this results in the following set of linear constraints.

**Definition 25** *($PRLP^T$).* The set of linear constraints $PRLP^T(\mathcal{K}, ra)$ contains all constraints from $PRLP(\mathcal{K}, ra)$ (generated with $T = T_\mathcal{K}$) except that constraints of the form

- $\forall p, q \in \mathcal{S}$, $t \in T$, $id \in ID$ if $reachable_{id}(p,q) \notin RD$: $v_{id,t,p,q} = 0$,
- $\sum_{q \in \mathcal{S}} v_{id,t,q,p} = \sum_{q \in \mathcal{S}} v_{id,t+1,p,q}$

are replaced by

- $\forall p, q \in \mathcal{S}$, $t_i \in T$, $id \in ID$ if $reachable_{id}(p,q,t_{i+1} - t_i) \notin RDT$: $v_{id,t_i,p,q} = 0$,
- $\sum_{q \in \mathcal{S}} v_{id,t_i,q,p} = \sum_{q \in \mathcal{S}} v_{id,t_{i+1},p,q}$.

The following theorem says that probabilistic revisions can be correctly computed from this reduced linear program $PRLP^T$.

**Theorem 13.** *(See Theorem 5 from [22].) For any given $\mathcal{K}$ and ra, there is a solution to $PRLP^T(\mathcal{K}, ra)$ iff there is a solution to $PRLP(\mathcal{K}, ra)$.*

### 5.2. Approximate probabilistic revision via space partitioning

While polynomial, the running time of the linear programming-based probabilistic revision algorithm can be quite large. With the optimizations mentioned above, there are at most $|\mathcal{S}|^2 \cdot |T_\mathcal{K}|$ variables (where $|T_\mathcal{K}|$ will be bounded by the min of $|\mathcal{K}|$ and $|T|$) and at most $|\mathcal{S}| \cdot |T_\mathcal{K}| + |\mathcal{K}|$ constraints. The size of the linear program that must then be solved is therefore $O((|\mathcal{S}| \cdot |T_\mathcal{K}| + |\mathcal{K}|) \cdot (|\mathcal{S}|^2 \cdot |T_\mathcal{K}|))$ — each constraint contains at most one entry per variable. Linear programming is a worst-case cubic operation, putting the entire process in:

$$O\big(\big((|\mathcal{S}| \cdot |T_\mathcal{K}| + |\mathcal{K}|) \cdot \big(|\mathcal{S}|^2 \cdot |T_\mathcal{K}|\big)\big)^3\big) = O\big(|\mathcal{S}|^9 \cdot |T_\mathcal{K}|^6 + |\mathcal{S}|^6 \cdot |T_\mathcal{K}|^3 \cdot |\mathcal{K}|^3\big).$$

The term with the largest exponent is $|\mathcal{S}|$, which suggests that focusing on decreasing the size of $\mathcal{S}$ will most drastically improve the running time of probabilistic revision.

Our strategy here will be to reduce the size of $\mathcal{S}$ by partitioning it into related sets of points based on the PST atoms in the KB.

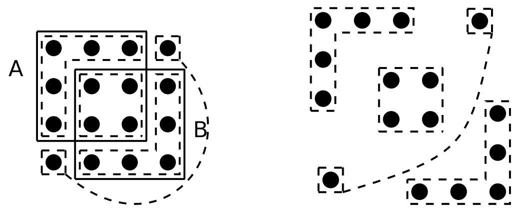


**Fig. 3.** An example partitioning according to point equivalence. Each dot is a point, each solid box on the left is a PST atom's region, and each dotted box is a partition (with the upper right and lower left dots being part of the same partition). On the right the partitions are displayed without the PST atom regions.

*5.2.1. Definition of partition*

We first define a concept of equivalence that leads to the partitioning. Given a PST KB $\mathcal{K}$, we use the notation $\mathcal{K}^{id,t}$ to denote the set of all PST atoms of the form $loc(id,-,t)[-,-]$ in $\mathcal{K}$.

**Definition 26** *(Point-Equivalence).* Suppose $\mathcal{K}$ is a PST KB, $ra = loc(id,r,t)[\ell,u]$ a revision atom, $t$ a time point, and $p_1, p_2$ are two points in $\mathcal{S}$. We say that $p_1$ and $p_2$ are *t-equivalent*, denoted $p_1 \sim_t p_2$, if and only if for all $loc(id,r,t)[\ell',u'] \in \mathcal{K}^{id,t}$, $p_1 \in r \Leftrightarrow p_2 \in r$.

Intuitively, when two points $p_1$, $p_2$ are $t$-equivalent, the variables associated with points $p_1$, $p_2$ in the linear program $PRLP(\mathcal{K}, ra)$ occur in exactly the same constraints.

**Example 17.** Fig. 3 shows a partitioning of space where each dot is a point in space and each solid box is a PST atom's region. The dotted regions represent the partitions. The knowledge base being used is

$$\big\{loc(id, A, 0)[\ell_1, u_1], loc(id, B, 0)[\ell_2, u_2]\big\}.$$

It is easy to see that $\sim_t$ is an equivalence relation and induces a set of equivalence classes. Let $\mathcal{P}^t$ be the set of such equivalence classes for a given time point $t$. Note that $|\mathcal{P}^t| \leqslant |\mathcal{S}|$. Our intuition is that the variables in $PRLP(\mathcal{K}, ra)$ associated with all points occurring in a given equivalence class can be collapsed into a single variable (representing the sum of the variables being collapsed).

*5.3. Reachability between partitions*

Suppose we have partitioned $\mathcal{S}$ for each time point $t_1, \ldots, t_n \in T_{\mathcal{K}}$, obtaining $\mathcal{P}^{t_1}, \ldots, \mathcal{P}^{t_n}$. We create two new versions of the reachable predicate to address an object's potential to move from partition $P \in \mathcal{P}^{t_i}$ to $P' \in \mathcal{P}^{t_{i+1}}$: cautious reachability and optimistic reachability.

**Definition 27.** For times $t$ and $t'$, partition $P \in P^t$ and partition $P' \in P^{t'}$,

- **Cautious Partition Reachability:**
  We say $reachable^{\forall}_{id}(P, P', t'-t)$ iff for all $p \in P$ and $q \in P'$, $reachable_{id}(p, q, t'-t)$.
- **Optimistic Partition Reachability:**
  We say $reachable^{\exists}_{id}(P, P', t'-t)$ iff there exists $p \in P$ and $q \in P'$, such that $reachable_{id}(p, q, t'-t)$.

Intuitively, cautious reachability requires that all points in $P'$ be reachable from all points in $P$ within a given time frame. Optimistic reachability, on the other hand, only requires that some point in $P'$ be reachable from some point in $P$ within a given time period.

**Example 18.** With a reachability predicate that returns true if the two points are at most Manhattan distance 2 apart, in Fig. 3 the cautious reachability predicate is never true, while the optimistic reachability predicate is always true. Fig. 4 shows another partitioning where the cautious reachability predicate is sometimes true (shown with arrows in the figure). Notice how the square center partition $p9$ is not reachable via the cautious reachability predicate in Fig. 4: this is due to the fact that every other partition contains at least one point more than distance 2 from at least one of the points in the square center partition (i.e. the lower right point of $p9$ does not reach the point in $p1$, the lower right point in $p9$ does not reach the upper point in $p2$, and so forth).

*5.4. Partition granularity*

The partitions and reachability conditions create a new space where partitions function as points, and objects can move between partitions just as they had previously moved between points. Under cautious partition reachability, however, the

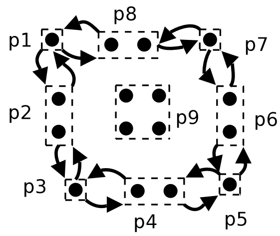


**Fig. 4.** This is a granularized version of the partitions in Fig. 3. With a reachability predicate that returns true for points Manhattan distance at most two, the arrows represent cautious reachability.

analogy is incomplete. Cautious reachability is so cautious that in many cases, there will be partitions that cannot be reached or cannot reach anywhere else. For instance, it may be that for all partitions $P \in \mathcal{P}^{t_i}$, no $P' \in \mathcal{P}^{t_{i-1}}$ can reach $P$ under cautious partition reachability.

In a process called "granularizing", we split partitions in each $\mathcal{P}^{t_i}$ until each partition can reach and be reached by at least one other partition under cautious partition reachability. The following example illustrates granularization.

**Example 19.** Fig. 4 shows a granularization of the partitioning in Fig. 3. Also displayed in Fig. 4 are arrows representing cautious partition reachability when the original reachability predicate allows the object to move at most Manhattan distance two. Notice that before granularization, there was no cautious partition reachability, while after granularization, we have a much more connected (though still not fully connected) graph.

Algorithm 3 accomplishes this granularization by iteratively dividing the partitions at time points $t_0$ and $t_1$ until every partition at time point $t_0$ can cautiously reach a partition at time point $t_1$ and every partition at time point $t_1$ is cautiously reachable from a partition at time point $t_0$.

---

**Algorithm 3. Granularize**$(\mathcal{P}^{t_0}, \mathcal{P}^{t_1})$: For time points $t_0$ and $t_1$, take partitions $\mathcal{P}^{t_0}$ and $\mathcal{P}^{t_1}$ and return $(\mathcal{P}^{t_0}_\star, \mathcal{P}^{t_1}_\star)$ where every member of $\mathcal{P}^{t_0}_\star$ can reach some member of $\mathcal{P}^{t_1}_\star$ under cautious partition reachability and every member of $\mathcal{P}^{t_1}_\star$ can be reached by some member of $\mathcal{P}^{t_0}_\star$ under cautious partition reachability.

---

1: Let $\mathcal{P}^{t_0}_\star = \mathcal{P}^{t_0}$ and $\mathcal{P}^{t_1}_\star = \mathcal{P}^{t_1}$.
2: Mark every partition in $\mathcal{P}^{t_0}$ and $\mathcal{P}^{t_1}$ as "not done".
3: **while** Any partition in $\mathcal{P}^{t_0}$ or $\mathcal{P}^{t_1}$ is marked "not done" **do**
4:   **for** Each partition $P \in \mathcal{P}^{t_0}_\star$ **do**
5:     **for** Each partition $P' \in \mathcal{P}^{t_1}_\star$ **do**
6:       If $P'$ is cautiously reachable from $P$ in time $t_1 - t_0$ then mark both $P$ and $P'$ as "done".
7:     **end for**
8:   **end for**
9:   **for** Each partition $P \in \mathcal{P}^{t_0}_\star$ not marked "done" **do**
10:     **for** Each partition $P' \in \mathcal{P}^{t_1}_\star$ **do**
11:       If there is any subset of $P'$ that $P$ cautiously reaches in time $t_1 - t_0$, let $P''$ be the largest such subset with minimal $x$ (resp. $y$) coordinates (for uniqueness). Add $P''$ and $P' \setminus P''$ to $\mathcal{P}^{t_1}$ and remove $P'$ (the "done" marks for $P''$ and $P' \setminus P''$ are inherited from $P'$).
12:       Mark $P$ and $P''$ "done".
13:       Goto line 3
14:     **end for**
15:   **end for**
16:   **for** Each partition $P \in \mathcal{P}^{t_1}_\star$ not marked "done" **do**
17:     **for** Each partition $P' \in \mathcal{P}^{t_0}_\star$ **do**
18:       If there is any subset of $P'$ that cautiously reaches $P$ in time $t_1 - t_0$, let $P''$ be the largest such subset with minimal $x$ (resp. $y$) coordinates (for uniqueness). Add $P''$ and $P' \setminus P''$ to $\mathcal{P}^{t_0}$ and remove $P'$ (the "done" marks for $P''$ and $P' \setminus P''$ are inherited from $P'$).
19:       Goto line 3
20:     **end for**
21:   **end for**
22:   If there is a partition not marked "done", find the largest partition in either $\mathcal{P}^{t_0}$ or $\mathcal{P}^{t_1}$, with minimal $x$ (resp. $y$) coordinates. Remove it from the partition set, split it in half, and add both halves as partitions to the partition set, inheriting any marks.
23: **end while**

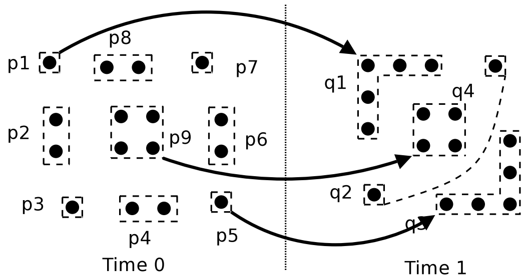


**Fig. 5.** Initial partitions of the same space at two time points, with arrows signifying cautious reachability.

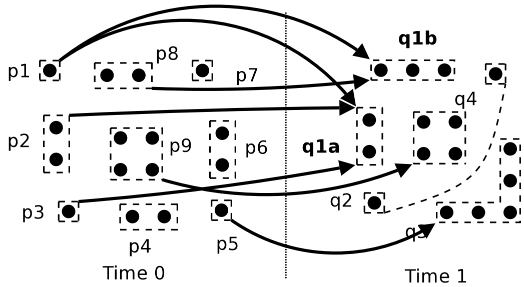


**Fig. 6.** First iteration splits $q1$ into $q1a$ and $q1b$.

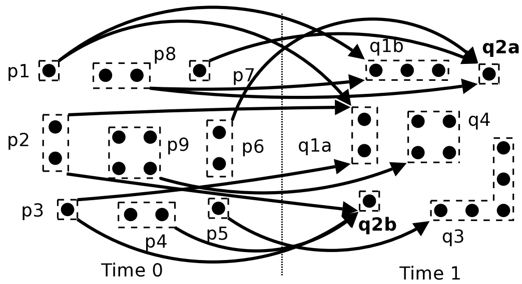


**Fig. 7.** Second iteration splits $q2$ into $q2a$ and $q2b$, finishing the granularization.

The following example shows how granularization works.

**Example 20.** Suppose the underlying $3 \times 3$ space is partitioned according to Fig. 4 at time point 0, and according to Fig. 3 at time point 1. For this example, we suppose a reachability predicate allowing the object to travel a Manhattan distance of at most 2 at each time point. The partitions are shown in Fig. 5 with the arrows signifying cautious reachability between the associated partitions. The algorithm divides the partitions so that every resulting partition at time 0 can cautiously reach a partition at time 1 (i.e. has an arrow originating from it), and every resulting partition at time 1 is cautiously reached by a partition at time 0 (i.e. has an arrow terminating at it).

On the first iteration of the while loop in Algorithm 3, the first unmarked partition encountered is $p2$, since $p2$ has no arrows originating from it. However, there is a subset of $q1$ that $p2$ reaches. The algorithm then removes $q1$ from the second partition, and replaces it with the partitions $q1a$ and $q1b$ shown in Fig. 6. Notice how with the division of $q1$, many more partitions are cautiously reachable.

On the second iteration of the while loop, the first unmarked partition encountered is $p4$. $p4$ can reach a subset of $q2$, so $q2$ is divided into $q2a$ and $q2b$ as shown in Fig. 7. After this step, enough cautious reachability predicates have become true that all partitions at time 0 can reach a partition at time 1, and all partitions at time 1 can be reached by a partition at time 0. Thus every partition is marked and the algorithm finishes.

The following result shows the correctness of the granularization procedure.

**Proposition 6.** *If the associated reachability definition is fully connected, then for $(\mathcal{P}^{t_0}, \mathcal{P}^{t_1})$ returned by Algorithm 3, for all $P \in \mathcal{P}^{t_0}$ there is $P' \in \mathcal{P}^{t_1}$ such that $P$ cautiously reaches $P'$ in time $t_1 - t_0$ and for all $P \in \mathcal{P}^{t_1}$ there is $P' \in \mathcal{P}^{t_0}$ such that $P$ cautiously reaches $P'$ in time $t_1 - t_0$.*

By partitioning space and using these new reachability predications, we can build new linear programs for computing approximate probabilistic revisions of a PST KB. In this section we assume such a consistent $\mathcal{K}$ and a revision atom $ra = loc(id, r, t)[\ell, u]$. We further assume that all atoms in $\mathcal{K}$ reference the same *id* as in the revision atom (we can do this without loss of generality because no atoms except those referencing the same *id* can be changed in probabilistic revision).

For technical purposes relating to the construction of the partitions, we add a null atom, $loc(id, r, t)[0, 1]$ (that does not change the meaning of $\mathcal{K}$) to $\mathcal{K}$. The inclusion of this null atom ensures that the time point $t$ will be in $T_{\mathcal{K}}$ and will include $r$ when computing the partitioning of $\mathcal{S}$ for time $t$. The linear program given in the next definition will then be guaranteed to have the variables relating to the revision atom.

As before, we create $T_{\mathcal{K}}$ as the set of time points mentioned in $\mathcal{K}$, and create a partitioning $\mathcal{P}^{t_i}$ for each $t_i \in T_{\mathcal{K}}$.

The variables for this set of linear constraints specify the probability of moving from partition to partition. They are of the form $v_{t_i,P,P'}$ where $P \in \mathcal{P}^{t_i}$ and $P' \in \mathcal{P}^{t_{i+1}}$. $v_{t_i,P,P'}$ represents the probability of *id* being in partition $P$ at time $t_i$ and moving to partition $P'$ at time $t_{i+1}$. For PST atoms $loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}$, the variables $low_i$ and $up_i$ are included as the revised atom's lower and upper bounds. We include only the more useful cautious case below; a similar definition can be given for the optimistic case.

**Definition 28** *(Cautious Partition Revision Linear Program).* For PST KB $\mathcal{K}$, revision atom *ra*, and granularizations $\mathcal{P}_\star^{t_i}$ of the partitions $\mathcal{P}^{t_i}$ for each $t_i \in T_{\mathcal{K}}$, the cautious partition revision linear program, $CPRLP(\mathcal{K}, ra, \{\mathcal{P}_\star^{t_i}\})$, contains the following constraints:

1. For each $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}$:
   (a) $0 \leqslant (\sum_{P\in\mathcal{P}_\star^{t_i},\, P\cap r_i\neq\emptyset} \sum_{P'\in\mathcal{P}_\star^{t_{i+1}}} v_{t_i,P,P'}) - low_i$.
   (b) $0 \geqslant (\sum_{P\in\mathcal{P}_\star^{t_i},\, P\cap r_i\neq\emptyset} \sum_{P'\in\mathcal{P}_\star^{t_{i+1}}} v_{t_i,P,P'}) - up_i$.
   (c) $\ell_i \geqslant low_i$, $low_i \geqslant 0$, $u_i \leqslant up_i$, and $up_i \leqslant 1$.
2. For $ra = loc(id, r, t)[\ell, u]$:
   (a) $\ell \leqslant (\sum_{P\in\mathcal{P}_\star^{t_i},\, P\cap r_i\neq 0} \sum_{P'\in\mathcal{P}_\star^{t_{i+1}}} v_{t_i,P,P'})$.
   (b) $u \geqslant (\sum_{P\in\mathcal{P}_\star^{t_i},\, P\cap r_i\neq 0} \sum_{P'\in\mathcal{P}_\star^{t_{i+1}}} v_{t_i,P,P'})$.
3. For each $t_i$ in $T_{\mathcal{K}}$ s.t. $t_{i+1} \in T_{\mathcal{K}}$:
   (a) For all $v_{t_i,P,P'}$, $v_{t_i,P,P'} \geqslant 0$.
   (b) $\sum_{P\in\mathcal{P}_\star^{t_i}} \sum_{P'\in\mathcal{P}_\star^{t_{i+1}}} v_{t_i,P,P'} = 1$.
   (c) For all $P \in \mathcal{P}_\star^{t_i}$, $P' \in \mathcal{P}_\star^{t_{i+1}}$ if $reachable^{\forall}_{id}(P, P', t_{i+1} - t_i) \notin RDT$: $v_{t_i,P,P'} = 0$.
   (d) For all $P \in \mathcal{P}_\star^{t_i}$: $\sum_{P'\in\mathcal{P}_\star^{t_i}} v_{t_i,P',P} = \sum_{P'\in\mathcal{P}_\star^{t_{i+1}}} v_{t_{i+1},P,P'}$.

Just as for the probabilistic revision algorithm (see Definition 20), we can minimize the objective function $\sum_i(\ell - low_i) + (up_i - u)$ subject to $CPRLP(\mathcal{K}, ra)$, as shown below. Unlike the probabilistic revision algorithm, the results will only be an approximation.

---

**Algorithm 4.** Computes an approximate probabilistic revision using partitioning of space for given $\mathcal{K}$ and $ra = loc(id, r, t)[\ell, u]$.

---

Let $\bar{\mathcal{K}} = \{loc(id', r', t')[\ell', u'] \in \mathcal{K} | id' = id\}$.
Add $loc(id, r, t)[0, 1]$ to $\bar{\mathcal{K}}$.
For each $t_i \in T_{\mathcal{K}}$, create $\mathcal{P}^{t_i}$, the partition of space according to point equivalence.
Set $\mathcal{P}_\star^{t_0} = \mathcal{P}^{t_0}$.
**for** $i = 0$ to $|T_{\mathcal{K}}| - 1$ **do**
  Let $(\mathcal{P}_\star^{t_i}, \mathcal{P}_\star^{t_{i+1}}) =$ **Granularize**$(\mathcal{P}^{t_i}, \mathcal{P}^{t_{i+1}})$.
**end for**
Construct $CPRLP(\bar{\mathcal{K}}, ra, \{\mathcal{P}_\star^{t_i}\})$.
Let $\theta$ be a solution minimizing $\sum_i(\ell - low_i) + (up_i - u)$ subject to $CPRLP(\mathcal{K}, ra, \{\mathcal{P}_\star^{t_i}\})$.
**return** $\{loc(id, r_i, t_i)[low_i\theta, up_i\theta] \mid loc(id, r_i, t_i)[\ell_i, u_i] \in \bar{\mathcal{K}}\} \cup \{ra\}$.

---

The following example shows how this algorithm works.

**Example 21.** Consider the knowledge base

$$\{a_1 = loc(id, A, 0)[0.5, 1], a_2 = loc(id, B, 0)[0.75, 1]\}$$

where $A$ and $B$ refer to the regions in Fig. 3. Further suppose the revision atom $ra = \{loc(id, A, 0)[0, 0]\}$, which states that the object is not in region $A$ at time 0. First, the algorithm sets $\bar{\mathcal{K}}$ to $\mathcal{K} \cup \{loc(id, A, 0)[0, 1]\}$, a dummy atom to ensure the appropriate partitions are created. Initially, Algorithm 4 partitions space according to point equivalence (resulting in the partitioning shown in Fig. 3). Then, after running the granularization (Algorithm 3) procedure before line 8, we let the resulting $\mathcal{P}_\star^0$ and $\mathcal{P}_\star^1$ be the partitionings shown in Fig. 7. The resulting linear program, $CPRLP(\mathcal{K}, ra, \{\mathcal{P}_\star^0, \mathcal{P}_\star^1\})$ is the following simplified set of linear constraints (all variables known to be zero have been removed):

1. Inequalities for each atom:
   (a) For $a_1 = loc(id, A, 0)[0.5, 1]$:
   $0 \leqslant v_{0,p1,q1a} + v_{0,p1,q1b} + v_{0,p2,q1a} + v_{0,p2,q2b} + v_{0,p9,q4} + v_{0,p8,q1b} + v_{0,p8,q2a} - low_1$,
   $0 \geqslant v_{0,p1,q1a} + v_{0,p1,q1b} + v_{0,p2,q1a} + v_{0,p2,q2b} + v_{0,p9,q4} + v_{0,p8,q1b} + v_{0,p8,q2a} - up_1$,
   $low_1 \leqslant 0.5$, $up_1 \geqslant 1$.
   (b) For $a_2 = loc(id, B, 0)[0.75, 1]$:
   $0 \leqslant v_{0,p9,q4} + v_{0,p4,q2b} + v_{0,p5,q3} + v_{0,p6,q2a} - low_2$,
   $0 \geqslant v_{0,p9,q4} + v_{0,p4,q2b} + v_{0,p5,q3} + v_{0,p6,q2a} - up_2$,
   $low_2 \leqslant 0.75$, $up_2 \geqslant 1$.
2. For the revision atom $loc(id, A, 0)[0, 0]$:
   $0 \leqslant v_{0,p1,q1a} + v_{0,p1,q1b} + v_{0,p2,q1a} + v_{0,p2,q2b} + v_{0,p9,q4} + v_{0,p8,q1b} + v_{0,p8,q2a}$,
   $1 \geqslant v_{0,p1,q1a} + v_{0,p1,q1b} + v_{0,p2,q1a} + v_{0,p2,q2b} + v_{0,p9,q4} + v_{0,p8,q1b} + v_{0,p8,q2a}$.
3. Extra constraints for correctness:
   (a) $v_{0,p1,q1a} \geqslant 0$, $v_{0,p1,q1b} \geqslant 0$, $v_{0,p2,q1a} \geqslant 0$, $v_{0,p2,q2b} \geqslant 0$, $v_{0,p3,q1a} \geqslant 0$, $v_{0,p3,q2b} \geqslant 0$, $v_{0,p4,q2b} \geqslant 0$, $v_{0,p5,q3} \geqslant 0$, $v_{0,p6,q2a} \geqslant 0$, $v_{0,p7,q2a} \geqslant 0$, $v_{0,p9,q4} \geqslant 0$.
   (b) $v_{0,p1,q1a} + v_{0,p1,q1b} + v_{0,p2,q1a} + v_{0,p2,q2b} + v_{0,p3,q1a} + v_{0,p3,q2b} + v_{0,p4,q2b} + v_{0,p5,q3} + v_{0,p6,q2a} + v_{0,p7,q2a} + v_{0,p9,q4} = 1$.
   (c) All variables known to be zero are removed.
   (d) Since we have only one time point in the example, no movement constraints are necessary.

This set of constraints is substantially smaller than $PRLP^T(\mathcal{K}, ra)$, which even after eliminating variables known to be zero would use at least 45 variables (here we have only 11). By minimizing $up_1 + up_2 - low_1 - low_2$ subject to those constraints, we will come up with a solution where $up_1 = 1$, $up_2 = 1$, $low_1 = 0$, and $low_2 = 0.75$. The resulting knowledge base is therefore: $\{loc(id, A, 0)[0, 1], loc(id, B, 0)[0.75, 1]\} \cup \{ra = loc(id, A, 0)[0, 0]\}$.

## 6. Related work

There is much work on *spatio-temporal logics* [10,16] in the literature. These logics extend temporal logics to handle space. There is also much work on qualitative spatio-temporal theories (for a survey see [5,18,28] which discusses the frame problem when constructing a logic-based calculus for reasoning about the movement of objects in a real-valued co-ordinate system). [25] focuses on relative position and orientation of objects with existing methods for qualitative reasoning in a Newtonian framework. Other efforts combine a spatial logic, such as $RCC-8$ [26], $BRCC-8$ [31] and $S4_u$ [4], with propositional temporal logics (*PTL*). The work on spatio-temporal reasoning is mostly qualitative [5,9,16,32], and focuses on relations between spatio-temporal entities while dealing with discrete time.

In contrast to all the important works mentioned above, our work brings two important new elements together.

- First, our work blends *probabilities* into the mix, and is intended for reasoning about moving objects whose location at a given point in time (past, present or future) is not known with certainty. The preceding work above does not.
- Second, this specific paper focuses on the problem of belief revision (not studied in past work) in spatio-temporal logics with uncertainty. We have not seen a treatment of belief revision in spatio-temporal logics thus far, and certainly not for probabilistic spatio-temporal logics.

In addition to the above works on spatio-temporal logics, there are works on logics integrating time and probabilities. Much of this work was performed in the model checking community. The PRISM system [14] supports a mix of time and probabilities for model checking by model checking w.r.t. specifications written in the temporal probabilistic logics PCTL [11] and CSL [2]. However, none of these works has any spatial element in them, and they focus on model checking, not on handling knowledge base updates.

Another related work by our group on reasoning about "go" theories [6–8] focused on spatio-temporal logical theories that were sets of "go" atoms. Such atoms intuitively described plans (known) of moving objects. A go-atom states that an object will go from location A to location B, leaving A at a time point in some time interval, arriving at B at a time point in some interval, and traveling in the interim at a velocity within some stated interval. [6] developed a basic theory of "go" theories, while [8] gave a closed world assumption for such theories. Later, [24] extended this logic to include some probabilistic information about such plans. We extended this work to uncertainty about where objects might be at a given time [20,21]. This paper builds upon the framework of [20,21], but makes two major changes. First, it addresses the problem of belief revision in probabilistic spatio-temporal theories — something we have not seen addressed before in the literature. Second (as a minor contribution), it adds the realistic requirement that vehicles can only reach certain places within certain time frames.

## 7. Conclusions

Though there has been extensive interest in AI over the last few years on reasoning about moving objects, to date we have seen no work (except the short version of this paper [23]) on belief revision in probabilistic spatio-temporal theories. This is a particularly important and thorny problem because information about moving objects may come in at a very high rate. For instance, moving objects with RFID tags on them will submit reports whenever they are within reach of a RFID scanner. The location of moving objects such as cell phones are continuously monitored by a satellite network (this is what allows police to track locations from where emergency cell phone calls have been made, or to track criminals engaged in cell phone communications). Additional applications include satellite tracking of birds and animals for wildlife and biological studies where the animals in question are constantly on the move. *What is common about all of these types of applications is that there is continuous change about when observations are made and where they are made.* Concurrently with such data collection methods are a wide suite of methods to predict where moving objects will be in the future. Recent work [29] focuses on predicting the destinations of moving objects using probabilistic HMMs. Mittu and Ross [17] and Hammel [30] developed methods to predict locations of enemy submarines in the future. Such prediction methods produce atoms closely resembling the PST atoms in this paper.

In this paper, we first build on the framework of our own past work [20,21] in order to include the realistic requirement that vehicles have velocity constraints and we show how to handle consistency checking in this setting. We then develop analogs of the AGM axioms to handle insertions into PST KBs and evaluate different ways of accomplishing these revisions.

We present three types of revision methods.

- The maximal consistent subset, and the maximal cardinality subset strategies revise a PST knowledge base in the presence of new information by completely *dropping* PST atoms (either by dropping a minimal subset, or a cardinality minimal subset). While both these methods satisfy our AGM-style axioms, they lead to computationally intractable problems.
- The second class of revision methods each *modify* one component of PST atoms in the PST knowledge base.
  - Spatial revisions only modify the spatial component of PST atoms and do not satisfy AGM-style axioms.
  - Temporal revisions only modify the temporal component of PST atoms, but lead to computational intractability.
  - Object revisions modify only the object id part of a PST atom and satisfy AGM-style axioms, but computing them is computationally intractable.
  - Probabilistic revisions have the wonderful feature that they both satisfy AGM-style axioms and are polynomially solvable.
- Our third class of revision methods are *hybrid* in nature, combining elements of the previous two methods. *Weighted hybrid revision* allows spatial, temporal, object and probabilistic revisions to co-occur — but again, computing an optimal revision of this kind is computationally intractable. Another hybrid mechanism — *Probabilistic Maximal Subset* revision combines maximal consistent subsets and probabilistic revision, and is also intractable.

Fig. 1 introduced in Section 1 presents a summary of our results — both in terms of AGM compliance and in terms of complexity results.

Last but not least, we study probabilistic revision further and develop methods to reduce the size of the problem so that the performance of probabilistic revision can be improved.

For example, consider the paper of [17] that describes a system to predict where (enemy) submarines will be in the future, given the presence of a sensor field that produces observations at irregular time periods. Whenever a new observation about a particular submarine is made, a prediction algorithm may produce new predictions about where that submarine will be in the future. When we attempt to add these new predictions to a PST knowledge base the result may well be inconsistent (because the new observation may conflict with past predictions — we all know predictions are often wrong) with the previous predictions and as a consequence, the PST knowledge base needs to be updated. The speed at which we need to update the PST knowledge base depends upon the speed with which new observations are being made. In the case of submarine observations via an underwater sonar field, the observations may not be coming in too frequently.

However, consider a different application where military air traffic control need to predict where enemy planes will be in the future. Such predictions may be made using algorithms that look at satellite surveillance data of the air space in question. However, the observations here are likely to be made much more frequently than in the submarine's case and hence, new predictions of the planned flight path of the enemy plane will be continuously generated. When inserting these into a knowledge base of predictions, we need to revise the PST knowledge base. Such revisions are likely to be needed quite frequently because the frequency of observations in this application is higher.

In general, much work remains to be done. A detailed implementation and experimental analysis of the heuristic algorithms in this paper needs to be done. Randomized algorithms to solve massive linear programs such as those in [13] need to be developed to further scale such belief revision algorithms. For instance, we need to scale them along several dimensions. First, we need to study what happens as more and more time points are considered, i.e. $|T|$ is increased. We also need to study what happens when $|\mathcal{S}|$ increases. A third thing is to study how these algorithms (and improvements upon them) perform when the frequency of updates is increased. Fast approximation algorithms, together with good approximation guarantees would also be helpful. In this paper, we have shown a number of NP-hardness results, but no *strong*

NP-hardness results (which generally show additionally that the problems are not polynomially approximable either). Such results would also be very useful. In addition, there is the possibility that priorities exist among the various atoms in a PST knowledge base. If we extend the PST-logic syntax to include the usual logic connectives (which we have not done here as the paper is already quite complex), and define logic consequence in the obvious way, then we can leverage the base revision paradigm [12] for this purpose. These are important directions for future work.

**Acknowledgements**

Researchers funded in part by ARO grant W911NF0910206, ONR grant N000140910685, and AFOSR grant FA95500510298. We thank the referees for many helpful comments and suggestions.

## Appendix A. The maximum cardinality subset sum problem

The maximum cardinality subset sum (MCSS) problem is defined as follows.

**Definition 29** *(MCSS Problem).* Given a set of positive integers $S = \{s_1, \ldots, s_n\}$, a positive integer constant $c$, and a subset $S' \subseteq S$, decide whether $S'$ is a maximum cardinality subset of $S$ which sum up exactly to $c$, that is, $\sum_{s_i \in S'} s_i = c$ and there is no subset $S'' \subseteq S$ such that $|S''| > |S'|$ and $\sum_{s_i \in S''} s_i = c$.

We show that MCSS is coNP-hard.

**Lemma 1.** *The Maximum Cardinality Subset Sum Problem is coNP-hard.*

**Proof.** We show a LOGSPACE reduction from the complement of the SUBSET SUM problem, which is coNP-hard [19], to our problem. The complement of the SUBSET SUM problem is defined as follows: given a set of positive integers $S = \{s_1, \ldots, s_n\}$ and a positive integer constant $c$, decide if there is no subset $S' \subseteq S$ such that $\sum_{s_i \in S'} s_i = c$.

Observe that the complement of SUBSET SUM is still coNP-hard even if we require that $S$ does not contain $\{c\}$. We consider this case in the following. We construct an instance of MCSS $(S_1, c_1, S'_1)$ starting from an instance $(S, c)$ of the complement of SUBSET SUM as follows. $S_1 = S \cup \{c\}$, $c_1 = c$, and $S'_1 = \{c_1\}$.

We prove that there is no subset $S' \subseteq S$ such that $\sum_{s_i \in S'} s_i = c$ iff $S'_1$ is a maximum cardinality subset of $S_1$ which sums up exactly to $c_1$ by contraposition.

($\Rightarrow$) Assume that $S'_1$ is *not* a maximum cardinality subset of $S_1$ which sums up exactly to $c_1$. Then there is $S''_1 \subseteq S_1$ such that $|S''_1| > |S'_1|$ and $\sum_{s_i \in S''_1} s_i = c_1$. As the elements of $S_1$ are positive integers, $S''_1$ does not contain $c_1$ (otherwise the sum would be greater than $c_1$). Thus, $S''_1 \subseteq S_1 \setminus \{c_1\}$, that is, $S'' \subseteq S$. As $c_1 = c$, there is also a subset of $S'' \subseteq S$ such that $\sum_{s_i \in S''} s_i = c$.

($\Leftarrow$) Assume that there is a subset $S' \subseteq S$ such that $\sum_{s_i \in S'} s_i = c$. As $S$ does not contain element $c$ and consists of positive integers only, $|S'| \geqslant 2$. Let $S''_1 \subseteq S_1$ contain the same elements of $S'$. Then there is $S''_1 \subseteq S_1$ such that $|S''_1| > |S'_1| = 1$ and $\sum_{s_i \in S''_1} s_i = c = c_1$. $\square$

## Appendix B. Proofs

**Proposition 1.** $\dot{+}_m$ *satisfies the AGM axioms.*

**Proof.** Axioms (A1) to (A4) are straightforward. Axiom (A5) also holds because $\mathcal{K} \dot{+}_m ra = \mathcal{K}' \cup ra$, where $\mathcal{K}'$ is the largest subset of $\mathcal{K}$ consistent with $ra$. The only way for $\mathcal{K} \dot{+}_m ra$ to be inconsistent is for $\mathcal{K}'$ to be empty and $ra$ inconsistent by itself (if $ra$ is consistent, in the worst case $\mathcal{K}' = \emptyset$). Axiom (A6) is verified because of the strict total ordering introduced for the maximal subsets. $\square$

**Theorem 2.** *Given PST KBs $\mathcal{K}$ and $\mathcal{K}'$, and revision atom $ra$, determining if $\mathcal{K}' \cup \{ra\} = \mathcal{K} \dot{+}_m ra$ is coNP-complete.*

**Proof.** (Membership) Given $\mathcal{K}$, $\mathcal{K}'$, and $ra$, a polynomial size witness for the complement of the problem of deciding whether $\mathcal{K}' \cup \{ra\} = \mathcal{K} \dot{+}_m ra$ is a PST KB $\mathcal{K}''$ such that $\mathcal{K}'' \cup \{ra\}$ accomplishes the revision of $\mathcal{K}$ by adding $ra$ via the subset strategy and either (i) $\mathcal{K}' \subsetneq \mathcal{K}''$ or (ii) $\mathcal{K}''$ precedes $\mathcal{K}'$ according the order $O_T$. As this witness can be verified in polynomial time, the problem is in coNP.

(Hardness) Take an instance of MCSS defined by $S = \{s_1, \ldots, s_n\}$, constant $c$ and $S' \subseteq S$ (Definition 29 in Appendix A). By Lemma 1, also in Appendix A, this problem is coNP-hard. We reduce from MCSS to show coNP-hardness. Let $tot = \sum_{s_i \in S} s_i$. Let $\mathcal{S} = \{p_1, \ldots, p_n, p_{n+1}\}$ ($n+1$ point space) and

$$\mathcal{K} = \{loc(id, \{p_i\}, 0)[s_i/tot, s_i/tot] \mid s_i \in S\}.$$

Let $ra = loc(id, \{p_{n+1}\}, 0)[1 - c/tot, 1 - c/tot]$, and let

$$\mathcal{K}' = \big\{loc\big(id, \{p_i\}, 0\big)[s_i/tot, s_i/tot] \mid s_i \in S'\big\}.$$

Consider orderings $O_T$ that prefer revision $\bar{\mathcal{K}}$ to $\bar{\mathcal{K}}'$ ($\bar{\mathcal{K}} < \bar{\mathcal{K}}'$) whenever $\sum_{loc(id,r,t)[\ell,u]\in\bar{\mathcal{K}}} \ell = c/tot$ and $\sum_{loc(id,r,t)[\ell,u]\in\bar{\mathcal{K}}'} \ell \neq c/tot$. If $\sum_{loc(id,r,t)[\ell,u]\in\bar{\mathcal{K}}} \ell = \sum_{loc(id,r,t)[\ell,u]\in\bar{\mathcal{K}}'} \ell = c/tot$ then $\bar{\mathcal{K}} < \bar{\mathcal{K}}'$ if $|\bar{\mathcal{K}}| > |\bar{\mathcal{K}}'|$. Finally, of all $\bar{\mathcal{K}}$ such that

$$\sum_{loc(id,r,t)[\ell,u]\in\bar{\mathcal{K}}} \ell = c/tot,$$

which also have size $|\mathcal{K}'|$, $\mathcal{K}'$ is minimal according to $O_T$. That is, $O_T$ prefers any *maximally sized* database whose atoms' lower bounds sum to $c/tot$, and further, of the databases of size $|\mathcal{K}'|$ whose atoms' lower bounds sum to $c/tot$, it prefers $\mathcal{K}'$.

Now we show by contraposition that $\mathcal{K}'$ is a maximal subset revision of $\mathcal{K}$ w.r.t. *ra* under $O_T$ iff $(S, c, S')$ is an instance of MCSS.

($\Rightarrow$): Suppose that $(S, c, S')$ is not an instance of MCSS. Then there is $S'' \subseteq S$ such that $\sum_{s\in S''} s = c$ and $|S''| > |S|$. Construct $\mathcal{K}''$:

$$\mathcal{K}'' = \big\{loc\big(id, \{p_i\}, 0\big)[s_i/tot, s_i/tot] \mid s_i \in S''\big\}.$$

$\mathcal{K}'' \cup \{ra\}$ is consistent, $\sum_{loc(id,r,t)[\ell,u]\in\mathcal{K}''} \ell = c/tot$ and $|\mathcal{K}''| > |\mathcal{K}'|$. Therefore $\mathcal{K}''$ is preferred by $O_T$ over $\mathcal{K}'$ and $\mathcal{K}'$ is not a revision of $\mathcal{K}$ by adding *ra* via the max-subset strategy.

($\Leftarrow$): Suppose $\mathcal{K}'$ is not a maximal subset revision of $\mathcal{K}$. Let $(S, c, S')$ be an instance of MCSS. Let $\mathcal{K}''$ be the optimal max-subset revision. Notice that $\mathcal{K}''$ cannot be a superset of $\mathcal{K}'$: the sum of the lower bounds of all atoms in $\mathcal{K}'$ is exactly $c/tot$, and the revision atom enforces a lower bound of $1 - c/tot$. Since all concerned atoms have neither intersecting regions nor zero lower bounds, the inclusion of any other atom would force inconsistency. Thus $\mathcal{K}'' \not\supset \mathcal{K}'$ and thus $\mathcal{K}''$ must be preferred over $\mathcal{K}'$ by $O_T$. $O_T$ prefers knowledge bases whose atoms' lower bounds sum to $c/tot$, thus the lower bounds of the atoms in $\mathcal{K}''$ sum to $c/tot$. Among knowledge bases whose atoms' lower bounds sum to $c/tot$ (including both $\mathcal{K}'$ and $\mathcal{K}''$), $O_T$ prefers the larger knowledge bases, and among such knowledge bases that are also the same size as $\mathcal{K}'$, $O_T$ prefers $\mathcal{K}'$. Therefore $|\mathcal{K}''| > |\mathcal{K}'|$ (otherwise $\mathcal{K}'$ would be preferred by $O_T$ and $\mathcal{K}''$ would not be optimal). Now construct $S'' = \{s_i \mid loc(id, \{p_i\}, 0)[s_i/tot, s_i/tot] \in \mathcal{K}''\}$. Note that $\sum_{s_i\in S''} s_i = c$ and since $|\mathcal{K}''| > |\mathcal{K}'|$, $|S''| > |S'|$. Because of $S''$, $(S, c, S')$ is not an instance of MCSS. $\square$

**Theorem 3.** *Given PST KBs $\mathcal{K}$ and $\mathcal{K}'$ and PST atom ra, determining whether $\mathcal{K}' \cup \{ra\}$ accomplishes the revision of $\mathcal{K}$ by adding ra via the max-subset strategy according only to Definition* 9 (irrespective of the order $O_T$) *can be accomplished in polynomial time w.r.t. the size of $\mathcal{K}$.*

**Proof.** We can use the following procedure to check if $\mathcal{K}'$ is a subset of $\mathcal{K}$ and $\mathcal{K}' \cup \{ra\}$ is consistent and there is no other $\mathcal{K}'' \cup \{ra\}$ that accomplishes the same revision such that $\mathcal{K}' \subsetneq \mathcal{K}''$. The idea is to take potential max-subset revision $\mathcal{K}'$, and check all supersets of $\mathcal{K}'$ that have exactly one more element and are still subsets of $\mathcal{K}$: if one of those subsets is consistent with *ra* then $\mathcal{K}'$ does not accomplish max-subset revision of $\mathcal{K}$. However, if all such supersets are inconsistent, we can verify that $\mathcal{K}'$ accomplishes max-subset revision by checking if $\mathcal{K}'$ is consistent with *ra*.

**CheckSubsets**($\mathcal{K}, \mathcal{K}', ra$)

1: If $\mathcal{K}'$ is not a subset of $\mathcal{K}$, return *false*.
2: If $\mathcal{K}' \cup \{ra\}$ is not consistent, return *false*.
3: Let $V = \{\mathcal{K}' \cup \{sa\} | sa \in \mathcal{K}, sa \notin \mathcal{K}'\}$ (notice $|V| \leqslant |\mathcal{K}|$).
4: Check the consistency of $\mathcal{K}'' \cup \{a\}$ for all $\mathcal{K}'' \in V$.
5: If all $\mathcal{K}'' \in V$ results in inconsistent $\mathcal{K}'' \cup \{ra\}$, then return *true*.
6: Return *false* otherwise.

Notice that every step of CheckSubsets is polynomially computable: the most expensive step is line 4, where we check at most $|\mathcal{K}|$ knowledge bases for consistency. However, since all the knowledge bases have size at most $|\mathcal{K}|$ and since consistency checking is polynomial, that line runs in polynomial time.

The existence of such a polynomial time algorithm allows us to show that determining if $\mathcal{K}'$ accomplishes max-subset revision is in both NP and coNP, and is therefore polynomially computable.

**In NP**: If $\mathcal{K}'$ accomplishes the revision of $\mathcal{K}$ by adding *ra* via the max-subset strategy, then this can be verified by checking that CheckSubsets($\mathcal{K}, \mathcal{K}', ra$) returns true.

**In coNP**: If $\mathcal{K}'$ does not accomplish the revision of $\mathcal{K}$ by adding *ra* via the max-subset strategy, then there is $\mathcal{K}'' \supset \mathcal{K}'$ such that $\mathcal{K}'' \cup \{ra\}$ is consistent. Take some $sa \in \mathcal{K}'' \setminus \mathcal{K}'$. $(\mathcal{K}' \setminus \{sa\}) \cup \{ra\}$ must be consistent (otherwise $\mathcal{K}''$ would be inconsistent). Thus the algorithm returns false in this case providing a polynomial time method for verifying that $\mathcal{K}'$ does not accomplish max-subset revision. $\square$

**Theorem 4.** *Determining if $\mathcal{K}'$ accomplishes max-cardinality revision of $\mathcal{K}$ by adding ra is coNP-complete.*

**Proof.** (Membership) Given $\mathcal{K}$, *ra*, and $\mathcal{K}'$, a polynomial size witness for the complement of the problem of deciding whether $\mathcal{K}'$ is a max-cardinality revision of $\mathcal{K}$ is a PST KB $\mathcal{K}''$ such that $\mathcal{K}'' \cup \{ra\}$ accomplishes the revision of $\mathcal{K}$ by adding *ra* via the subset strategy and $|\mathcal{K}''| > |\mathcal{K}'|$. As this witness can be verified in polynomial time, the problem is in coNP.

(Hardness) Take an instance of MCSS defined by $S = \{s_1, \ldots, s_n\}$, constant $c$ and $S' \subset S$ (Definition 29 in Appendix A). By Lemma 1, also in Appendix A, this problem is coNP-hard. We reduce from MCSS to show coNP-hardness. Let $tot = 1 + \sum_{s_i \in S} s_i$. Let $\mathcal{S} = \{p_1, \ldots, p_n, p_{n+1}\}$ ($n+1$ point space). We use the knowledge base

$$\mathcal{K}_{back} = \{loc(id, \{p_1, \ldots, p_n\}, 0)[c/tot, 1 - k/((n+1) \cdot tot)] \mid k = 1, \ldots, n+1\}$$

as a backdrop of $n+1$ atoms (distinguished by their upper bounds) that we do not expect to change. $\mathcal{K}_{back}$ ensures that unless we remove more than $n+1$ atoms the lower bound for the probability in the region $\{p_1, \ldots, p_n\}$ will be $c/tot$. Let

$$\mathcal{K} = \{loc(id, \{p_i\}, 0)[s_i/tot, s_i/tot] \mid s_i \in S\} \cup \mathcal{K}_{back}.$$

Let $ra = loc(id, \{p_{n+1}\}, 0)[1 - c/tot, 1 - c/tot]$, and let

$$\mathcal{K}' = \{loc(id, \{p_i\}, 0)[s_i/tot, s_i/tot] \mid s_i \in S'\} \cup \mathcal{K}_{back}.$$

We show by contraposition that $\mathcal{K}'$ accomplishes revision of $\mathcal{K}$ by *ra* via max-cardinality iff $S'$ is the maximal cardinality subset of $S$ such that $\sum_{s \in S'} s = c$.

($\Rightarrow$): Suppose that $S'$ is not a maximal cardinality subset of $S$ such that $\sum_{s \in S'} s = c$. Then there is $S'' \subset S$ with $|S''| > |S'|$ such that $\sum_{s \in S''} s = c$. Let $\mathcal{K}'' = \{loc(id, \{p_i\}, 0)[s_i/tot, s_i/tot] \mid s_i \in S''\} \cup \mathcal{K}_{back}$. Since $|S''| > |S'|$, $|\mathcal{K}''| > |\mathcal{K}'|$. Further, since $\sum_{s \in S''} s = c$, $\mathcal{K}''$ is consistent with *ra*. Thus $\mathcal{K}'$ does not accomplish the revision of $\mathcal{K}$ by adding *ra* via max-cardinality.

($\Leftarrow$): Suppose that $\mathcal{K}'$ is not the max-cardinality revision of $\mathcal{K}$. Let $\mathcal{K}''$ be a max-cardinality revision of $\mathcal{K}$. Notice that $ra \cup \mathcal{K}_{back}$ causes no problem: members of $\mathcal{K}_{back}$ ensure that the area $\{p_1, \ldots, p_n\}$ has probability at least $c/tot$, while *ra* ensures that the area $\{p_1, \ldots, p_n\}$ has probability at most $c/tot$. To eliminate the requirement that $\{p_1, \ldots, p_n\}$ has probability at least $c/tot$, all $n+1$ members of $\mathcal{K}_{back}$ would have to be removed from $\mathcal{K}$, and therefore we can be assured that in $\mathcal{K}''$, there is at least one atom $loc(id, \{p_1, \ldots, p_n\}, 0)[c/tot, \cdot]$. Thus the sum of atoms in $\mathcal{K}''$ that are not also in $\mathcal{K}_{back}$ must be at least $c/tot$:

$$\sum_{loc(id,\{p_i\},0)[s_i/tot, s_i/tot] \in \mathcal{K}'' \setminus \mathcal{K}_{back}} s_i/tot \geqslant c/tot.$$

Also, because $\mathcal{K}''$ is compatible with *ra*, those same atoms can add up to no more than $c/tot$:

$$\sum_{loc(id,\{p_i\},0)[s_i/tot, s_i/tot] \in \mathcal{K}'' \setminus \mathcal{K}_{back}} s_i/tot \leqslant c/tot.$$

Thus we have that

$$\sum_{loc(id,\{p_i\},0)[s_i/tot, s_i/tot] \in \mathcal{K}'' \setminus \mathcal{K}_{back}} s_i/tot = c/tot.$$

Thus we can create $S'' = \{s_i \mid loc(id, \{p_i\}, 0)[s_i/tot, s_i/tot] \in \mathcal{K}'' \setminus \mathcal{K}_{back}\}$, and we are guaranteed that $\sum_{s_i \in S''} s_i = c$. Since $\mathcal{K}''$ is bigger than $\mathcal{K}'$, it follows that $S''$ is bigger than $S'$. So $S'$ is not a maximal cardinality subset of $S$ such that $\sum_{s \in S'} s = c$. □

**Proposition 2.** *For knowledge base $\mathcal{K}$ and revision atom ra, $\mathcal{K} \dotplus_c ra$ is an AGM-compliant revision function.*

**Proof.** (A1)–(A5) are straightforward. (A6) follows from the use of the strict total ordering $O_T$ to determine $\mathcal{K} \dotplus_c ra$ from all knowledge bases accomplishing max-cardinality revision of $\mathcal{K}$ by *ra*. □

**Theorem 5.** *Suppose that $\mathcal{K}$ is a PST KB and ra a PST atom. Let $\theta$ be a solution of the optimization problem*

$$\textbf{\textit{minimize}} \sum_{a_i \in \mathcal{K}} \delta_i \quad \textbf{\textit{subject to}}\ MCSRP(\mathcal{K}, ra).$$

*Then $\mathcal{K}' = \{a_i \in \mathcal{K} \mid \delta_i\theta = 0\} \cup \{ra\}$ accomplishes the revision of $\mathcal{K}$ by adding ra via the max-cardinality strategy.*

**Proof.** For each $\hat{\mathcal{K}} \subset \mathcal{K}$, let $\Theta_{\hat{\mathcal{K}}}$ be the set $\{\theta \mid \theta$ is a solution of *MCSRP*$(\mathcal{K}, ra)$ such that $\forall a_i \in \hat{\mathcal{K}}, \delta_i\theta = 1\}$. We first show that $\Theta_{\hat{\mathcal{K}}}$ is equivalent to the set of solutions of $LP(\mathcal{K} \setminus \hat{\mathcal{K}})$ (see Definition 8), in the sense that for each solution $\theta \in \Theta_{\hat{\mathcal{K}}}$, there is a solution $\theta'$ of $LP(\mathcal{K} \setminus \hat{\mathcal{K}})$ such that $v_{id,t,p,q}\theta = v_{id,t,p,q}\theta'$, and vice versa.

Let $\hat{\mathcal{K}} \subset \mathcal{K}$ and $\theta \in \Theta_{\hat{\mathcal{K}}}$. Then, for each $a_i \in \hat{\mathcal{K}}$, constraint (1)(a) is equal to $\ell_i - 1 \leqslant (\sum_{p \in r_i} \sum_{q \in \mathcal{S}} v_{id_i,t_i,p,q}) \leqslant u_i + 1$. This is equivalent to using atom $a'_i = loc(id_i, r_i, t_i)[0, 1]$ instead of $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i]$ in $\hat{\mathcal{K}}$. Since $\mathcal{K} \setminus \hat{\mathcal{K}} \cup \{a' =$

$loc(id_i, r_i, t_i)[0,1] \mid a = loc(id_i, r_i, t_i)[\ell, u] \in \hat{\mathcal{K}}\} \equiv \mathcal{K} \setminus \hat{\mathcal{K}}$, for $\theta \in \Theta_{\hat{\mathcal{K}}}$, the constraints in $MCSRP(\mathcal{K}, ra)$ are equivalent to those in $LP(\mathcal{K} \setminus \hat{\mathcal{K}})$. Thus there is a solution $\theta'$ of $LP(\mathcal{K} \setminus \hat{\mathcal{K}})$ such that $v_{id,t,p,q}\theta' = v_{id,t,p,q}\theta$. Next assume that there is a solution $\theta'$ of $LP(\mathcal{K} \setminus \hat{\mathcal{K}})$. It is easy to see that constraints (1)(a) for atoms $a_i \in \hat{\mathcal{K}}$ are satisfied by every solution $\theta$ of $MCSRP(\mathcal{K}, ra)$ such that $\delta_i\theta = 1$, that is, they are satisfied by $\theta \in \Theta_{\hat{\mathcal{K}}}$. Moreover all the other constraints in $MCSRP(\mathcal{K}, ra)$ are satisfied by making $v_{id,t,p,q}\theta = v_{id,t,p,q}\theta'$. Thus for each $\hat{\mathcal{K}} \subset \mathcal{K}$, $\Theta_{\hat{\mathcal{K}}}$ is not empty iff $LP(\mathcal{K} \setminus \hat{\mathcal{K}})$ has a solution iff $\mathcal{K} \setminus \hat{\mathcal{K}} \cup \{ra\}$ is consistent.

We now show that every optimal solution of the optimization problem determines a knowledge base $\mathcal{K}'$ accomplishing the revision of $\mathcal{K}$ by adding $ra$ via the max-cardinality strategy.

For each solution $\theta$ of $MCSRP(\mathcal{K}, ra)$, define $\mathcal{K}(\theta) = \{a_i \in \mathcal{K} \mid \delta_i\theta = 0\}$. Clearly, for each solution $\theta$ of $MCSRP(\mathcal{K}, ra)$, it is the case that $\theta \in \Theta_{\mathcal{K} \setminus \mathcal{K}(\theta)}$. Moreover, as shown above, $\mathcal{K}(\theta) \cup \{ra\}$ is consistent.

Let $\bar{\theta}$ be an optimal solution of the optimization problem, and let $\mathcal{K}'$ be $\mathcal{K}(\bar{\theta})$. As $\bar{\theta}$ is a solution of $MCSRP(\mathcal{K}, ra)$ and $\bar{\theta} \in \Theta_{\mathcal{K} \setminus \mathcal{K}'}$, $\mathcal{K}' \cup \{ra\}$ is consistent. As the job of the objective function is to minimize the sum of the $\delta_i$ values, there is no solution $\theta'$ of $MCSRP(\mathcal{K}, ra)$ such that $\Sigma_{a_i \in \mathcal{K}}\delta_i\theta' < \Sigma_{a_i \in \mathcal{K}}\delta_i\bar{\theta}$. For every solution $\theta$ of $MCSRP(\mathcal{K}, ra)$, $|\mathcal{K}(\theta)| = |\mathcal{K}| - \Sigma_{a_i \in \mathcal{K}}\delta_i\theta$. Thus for every solution $\theta$ of $MCSRP(\mathcal{K}, ra)$, corresponding to a knowledge base $\mathcal{K}(\theta)$ such that $\mathcal{K}(\theta) \cup \{ra\}$ is consistent, $|\mathcal{K}(\theta)| \leqslant |\mathcal{K}(\bar{\theta})|$, proving that $\mathcal{K}'$ accomplishes the revision of $\mathcal{K}$ by adding $ra$ via the max-cardinality strategy. □

**Theorem 6.** *Let $\mathcal{K}$ be any knowledge base and $|\mathcal{S}| > 2$. A spatial revision satisfying the AGM axioms is possible for every atom $ra = loc(id, r, t)[\ell, u]$ where $r$ is a strict subset of $\mathcal{S}$ iff for all $a_i = loc(id, r_i, t)[\ell_i, u_i] \in \mathcal{K}$ either $l_i = 0$ or $u_i = 1$.*

**Proof.** ($\Leftarrow$) To show that a revision satisfying the AGM axioms is possible, it suffices show that a spatial revision consistent with the revision atom exists because then there will be an optimal spatial revision and Axioms (A1)–(A5) follow directly — (A6) follows from the use of $O_T$. We now construct such a revision. Suppose that for all $a_i \in \mathcal{K}$, $\ell_i = 0$ or $u_i = 1$, and $ra = loc(id, r, t)[\ell, u]$ is given. For every $a_i$ for which $u_i = 1$, change $r_i$ to *Space*. For every $a_i$ for which $\ell_i = 0$, there are two cases: (1) if $|r| > 1$, change all such $r_i$ to the same point in $r$, and (2) if $|r| = 1$, change all such $r_i$ to the same point in $\mathcal{S} - r$.

($\Rightarrow$) Suppose that for some $a_i = loc(id_i, r_i, t_i)[\ell_i, u_i] \in \mathcal{K}$, $\ell_i \neq 0$ and $u_i \neq 1$. Let $ra = loc(id_i, \{p\}, t_i)[1, 1]$ where $p \in r_i$. There is no spatial revision. □

**Proposition 3.** *If $T$ is infinite and RD is fully connected, then there exists a minimal temporal revision $\mathcal{K}' = \mathcal{K} \dotplus_t ra$ that satisfies the AGM axioms.*

**Proof.** It suffices to show that there exists a temporal revision of $\mathcal{K}$ consistent with $ra$. Let $ra = loc(id, r, t)[\ell, u]$ and $\mathcal{K} = \{loc(id_i, r_i, t_i)[\ell_i, u_i]\}$. Let $\Delta t_{id}$ be the maximal time it takes to move from any point in space to any other according to the reachability predicate. Define $\mathcal{K}'' = \{loc(id, r_i, t + (i+1) \cdot \Delta t_{id})[\ell_i, u_i] \mid id_i = id\} \cup \{loc(id_i, r_i, t_i)[\ell_i, u_i] \mid id_i \neq id\}$. Since all atoms referencing $id$ refer to time points $\Delta t_{id}$ apart, the object $id$ can have moved anywhere in space between atoms and no two atoms (including the revision atom) can contradict one another. Thus $\mathcal{K}''$ is a temporal revision consistent with $ra$.

That the knowledge base $\mathcal{K}' = \mathcal{K} \dotplus_t ra$ satisfies the AGM Axioms (A1)–(A4) follows directly from the fact that if $\mathcal{K}$ is consistent with $ra$, $\mathcal{K}' = \mathcal{K} \cup \{ra\}$. Since a consistent $\mathcal{K}'$ always exists, Axiom (A5) is verified. Finally, the total ordering $O_T$ ensures compliance with Axiom (A6). □

**Theorem 7.** *Given a PST atom ra, and PST KBs $\mathcal{K}$, and $\mathcal{K}'$, deciding whether $\mathcal{K}'$ is a temporally optimal revision of $\mathcal{K}$ by ra is coNP-hard. Further, if $T$ is finite then deciding whether $\mathcal{K}'$ is a temporally optimal revision of $\mathcal{K}$ by adding ra is coNP-complete.*

**Proof.** (Membership) Given $\mathcal{K}'$, a polynomial size witness for the complement of the problem of deciding whether $\mathcal{K}'$ is a temporally optimal revision of $\mathcal{K}$ is a $\mathcal{K}''$ such that (i) $\mathcal{K}''$ is a temporally optimal revision of $\mathcal{K}$; (ii) $\mathcal{K}'' \cup \{ra\}$ is consistent; and (iii) $d_T(\mathcal{K}, \mathcal{K}'') < d_T(\mathcal{K}, \mathcal{K}')$. Obviously, items (i) and (iii) can be verified in polynomial time. Moreover, if $T$ is finite, item ii) can also be verified in polynomial time by solving $LP(\mathcal{K}'' \cup \{ra\})$.

(Hardness) We formalize the temporally optimal revision problem as follows: $(\mathcal{K}, ra, \mathcal{K}')$ is a positive instance of the optimal temporal revision problem iff $\mathcal{K}'$ is an optimal temporal revision of $\mathcal{K}$ by adding $ra$ (here we do not consider the total order $O_T$).

We give a LOGSPACE reduction from the Maximum Cardinality Subset Sum (MCSS) (see Definition 29 in Appendix A) problem to our problem. Lemma 1 from Appendix A states that MCSS is coNP-hard.

Consider an instance of MCSS $(S = \{s_1, \ldots, s_n\}, c, S')$, and construct an instance of optimal temporal revision as follows. Let $\mathcal{S} = \{p_1, \ldots, p_n, p_{n+1}, \ldots, p_{2n+1}\}$. Create a reachability predicate where $reachable_{id}(p_i, p_j) \in RD$ if: $j = n+1$ and $i < n+1$, or $i > n$ and $j = i+1$, or $j < n+1$ and $i = 2n+1$. Let $Tot = \sum_{s_i \in S} s_i$. Let $\mathcal{K} = \{loc(id, \{p_i\}, 0)[s_i/Tot, s_i/Tot] \mid s_i \in S\}$, $ra = loc(id, \{p_{n+1}\}, 1)[c/Tot, c/Tot]$, and

$$\mathcal{K}' = \bigl\{loc\bigl(id, \{p_i\}, 0\bigr)[s_i/Tot, s_i/Tot] \mid s_i \in S'\bigr\} \cup \bigl\{loc\bigl(id, \{p_i\}, 1\bigr)[s_i/Tot, s_i/Tot] \mid s_i \notin S'\bigr\}.$$

Now we show by contraposition that $(S, c, S')$ is in MCSS iff $(\mathcal{K}, ra, \mathcal{K}')$ is an optimal temporal revision.

($\Rightarrow$): Notice that $d_T(\mathcal{K},\mathcal{K}') = |S| - |S'|$. Suppose that $(\mathcal{K}, ra, \mathcal{K}')$ is not an optimal temporal revision. Then there is a temporal revision $\mathcal{K}''$ s.t. $d_T(\mathcal{K},\mathcal{K}'') < d_T(\mathcal{K},\mathcal{K}')$. Let $\mathcal{K}''_{\neq 0} = \{loc(id, \{p_i\}, t_i)[s_i/Tot, s_i/Tot] \in \mathcal{K}'' \mid t_i \neq 0\}$ and let $\mathcal{K}''_0 = \mathcal{K}'' \setminus \mathcal{K}''_{\neq 0}$. Observe several things about $\mathcal{K}''_{\neq 0}$:

- For all $loc(id, \{p_i\}, t_i)[s_i/Tot, s_i/Tot]$, $t_i \leqslant n$. If not, then $d_T(\mathcal{K},\mathcal{K}'') > n$ and since $|S'| \leqslant n$, $d_T(\mathcal{K},\mathcal{K}')$ would be smaller and $\mathcal{K}''$ would not be the counter example.
- Because of the reachability predicate, all $t_i > 0$ are equal to 1. If $ra$ is consistent with $\mathcal{K}''$, then $ra$ is consistent with
$$\mathcal{K}''_0 \cup \big\{loc\big(id, \{p_i\}, 1\big)[s_i/Tot, s_i/Tot] \mid loc\big(id, \{p_i\}, t_i\big)[s_i/Tot, s_i/Tot] \in \mathcal{K}''_{\neq 0}\big\}.$$
Since $\mathcal{K}''$ has minimal $d_T(\mathcal{K},\mathcal{K}'')$ for all $loc(id, \{p_i\}, t_i)[s_i/Tot, s_i/Tot] \in \mathcal{K}''_{\neq 0}$, $t_i = 1$.
- Because $d_T(\mathcal{K},\mathcal{K}'') < d_T(\mathcal{K},\mathcal{K}')$, $|\mathcal{K}''_{\neq 0}| < |S| - |S'|$.
- Because of the reachability predicate, the probability left in locations $p_1, \ldots, p_n$ at time 0 all goes to the location $p_{n+1}$. Thus to be consistent with $ra$, it must be the case that:
$$\sum_{loc(id,\{p_i\},0)[s_i/Tot,s_i/Tot]\in\mathcal{K}''_0} s_i/Tot \leqslant c/Tot.$$
- Because all atoms in $\mathcal{K}''_{\neq 0}$ are at time point 1, to be consistent with $ra$, it must be the case that:
$$\sum_{loc(id,\{p_i\},t_i)[s_i/Tot,s_i/Tot]\in\mathcal{K}''_{\neq 0}} s_i/Tot \leqslant 1 - c/Tot.$$
As $\sum_{s_i} s_i/Tot = 1$, and since $\mathcal{K}''_0$ and $\mathcal{K}''_{\neq 0}$ are disjoint and cover $\mathcal{K}''$, this implies:
$$\sum_{loc(id,\{p_i\},t_i)[s_i/Tot,s_i/Tot]\in\mathcal{K}''_0} s_i/Tot \geqslant c/Tot.$$
- Taken together, the previous three points imply
$$\sum_{loc(id,\{p_i\},t_i)[s_i/Tot,s_i/Tot]\in\mathcal{K}''_0} s_i/Tot = c.$$

Construct $S''$ where
$$S'' = \big\{s_i \mid loc\big(id, \{p_i\}, t_i\big)[s_i/Tot, s_i/Tot]\big\} \in \mathcal{K}''_0.$$
Since $\sum_{loc(id,\{p_i\},t_i)[s_i/Tot,s_i/Tot]\in\mathcal{K}''_{\neq 0}} s_i/Tot = c$, we know that $\sum_{s_i\in S''} s_i = c$. Further, since $d_T(\mathcal{K},\mathcal{K}'') < d_T(\mathcal{K},\mathcal{K}')$ we know that $|S| - |S''| < |S| - |S'| \Rightarrow |S''| > |S'|$. Because of $S''$, $(S, c, S')$ is not in MCSS.

($\Leftarrow$): Suppose that $(S, c, S')$ is not in MCSS. Let $S''$ be a counter-example to $(S, c, S')$ being in MCSS. Construct $\mathcal{K}''$ as:
$$\mathcal{K}'' = \big\{loc\big(id, \{p_i\}, 0\big)[s_i/Tot, s_i/Tot] \mid s_i \in S''\big\} \cup \big\{loc\big(id, \{p_i\}, 1\big)[s_i/Tot, s_i/Tot] \mid s_i \notin S''\big\}.$$
Note that since $|S''| > |S'|$, $d_T(\mathcal{K},\mathcal{K}'') < d_T(\mathcal{K},\mathcal{K}')$. Further notice that since $\sum_{s\in S''} s = c$, the total assigned probability at time 1 for $id$ is 1 in $\mathcal{K}'' \cup \{ra\}$ and it is therefore consistent. Thus $\mathcal{K}''$ is a counterexample to $(\mathcal{K}, ra, \mathcal{K}')$ being a temporally optimal revision. □

**Corollary 1.** *Given a PST atom ra, and PST KBs $\mathcal{K}$, $\mathcal{K}'$ where $T$ is finite, checking whether $\mathcal{K}' = \mathcal{K} \dotplus_t ra$ is coNP-complete.*

**Proof.** (Membership) A polynomial size witness for the complement of the problem of deciding whether $\mathcal{K}' = \mathcal{K} \dotplus_t ra$ is a temporal revision $\mathcal{K}''$ such that $\mathcal{K}'' \cup \{ra\}$ is consistent and either $d_T(\mathcal{K},\mathcal{K}'') < d_T(\mathcal{K},\mathcal{K}')$ or $d_T(\mathcal{K},\mathcal{K}'') = d_T(\mathcal{K},\mathcal{K}')$ and $\mathcal{K}''$ precedes $\mathcal{K}$ according the strict total order $O_T$. As in the case of the membership proof of Theorem 9, this witness can be verified in polynomial time.

(Hardness) Consider the proof of Theorem 7 and let $O_T$ be such that, for $d_T(\mathcal{K},\mathcal{K}') = |S| - |S'|$, $\mathcal{K}'$ is its minimum element. Then, reasoning as in the proof of Theorem 7, it is easy to see that $(S, c, S')$ is in MCSS iff $\mathcal{K}' = \mathcal{K} \dotplus_t ra$. □

**Theorem 8.** *If there exists a minimal temporal revision for $\mathcal{K}$ with respect to ra, then Algorithm 2 is correct, i.e.* ***TemporalRevision***$(\mathcal{K}$, *ra*) *returns $\mathcal{K} \dotplus_t ra$. Moreover,* ***TemporalRevision***$(\mathcal{K}, a)$ *satisfies the AGM axioms.*

**Proof.** Suppose that $\mathcal{K}'$ is the minimal temporal revision. That Algorithm 2 returns $\mathcal{K}' \cup \{ra\}$ follows from the order in which the algorithm checks the consistency of potential temporal revisions $\mathcal{K}''$. In each iteration of the while loop, $d(\mathcal{K},\mathcal{K}'')$ for all $\mathcal{K}''$ in *TODO* increases by one. Since each element of *TODO* is checked at each iteration, the algorithm cannot return any $\mathcal{K}''$ such that $d(\mathcal{K},\mathcal{K}'') < d(\mathcal{K},\mathcal{K}')$. Further, the list *TODO* is sorted according to $O_T$, therefore when the temporal distance from any element of *TODO* to $\mathcal{K}$ is $d(\mathcal{K},\mathcal{K}')$, the first element from *TODO* checked will be $\mathcal{K}'$ (when *TODO* is sorted according to $O_T$, $\mathcal{K}'$ will be placed at the front). Thus $\mathcal{K}' = \mathcal{K} \dot{+}_t ra$ will be returned.

That $\mathcal{K} \dot{+}_t ra$ complies with the AGM axioms is straightforward. □

**Theorem 9.** *Given a PST atom ra, and PST KBs $\mathcal{K}$, $\mathcal{K}'$ where $T$ is finite, deciding whether $\mathcal{K}'$ is an optimal object id revision of $\mathcal{K}$ is coNP-complete.*

**Proof.** (Membership) Given $\mathcal{K}'$, a polynomial size witness for the complement of the problem of deciding if $\mathcal{K}'$ is an optimal object id revision of $\mathcal{K}$ is a $\mathcal{K}''$ such that (i) $\mathcal{K}''$ is an object id revision of $\mathcal{K}$; (ii) $\mathcal{K}'' \cup \{ra\}$ is consistent; and (iii) $d_O(\mathcal{K},\mathcal{K}'') < d_O(\mathcal{K},\mathcal{K}')$. Obviously, this witness can be verified in polynomial time for items (i) and (iii). Moreover, as $T$ is finite, item (ii) can also be verified in polynomial time by solving $LP(\mathcal{K}'' \cup \{a\})$.

(Hardness) We show a LOGSPACE reduction from the Maximum Cardinality Subset Sum (MCSS) (see Definition 29 in Appendix A) problem to our problem. Lemma 1 states that MCSS is coNP-hard.

We construct an instance $\langle \mathcal{K}, ra, \mathcal{K}'\rangle$ of our problem starting from instance $\langle S = \{s_1,\dots,s_n\}, c, S'\rangle$ of MCSS as follows.

Let $\mathcal{S} = \{p_1,\dots,p_n,p_{n+1}\}$, *ID* finite such that $\{id_1, id_2\} \subseteq ID$, $T$ a finite set of time points which includes $t_0$, $d_O(id_1,id_2) = d_O(id_2,id_1) = 1$ and for all $id_i, id_j \in ID$ such that $i$ and $j$ are different from 1 and 2, $d_O(id_i,id_j) = n+1$, and $Tot = \sum_{s_i\in S} s_i$.

Observe that MCSS is still coNP-hard if $c \neq Tot/2$. We consider this case in our proof.

Assume that $RD = \emptyset$. We define an instance $\langle \mathcal{K}, ra, \mathcal{K}'\rangle$ of the problem as follows:

- $\mathcal{K} = \{loc(id_1,\{p_i\},t_0)[\frac{s_i}{Tot},\frac{s_i}{Tot}] \mid s_i \in S\} \cup \{loc(id_2,\{p_1,\dots,p_n\},t_0)[1-\frac{c}{Tot},1-\frac{c}{Tot}]\}$,
- $a = loc(id_1,\{p_1,\dots,p_n\},t_0)[\frac{c}{Tot},\frac{c}{Tot}]$,
- $\mathcal{K}' = \{loc(id_1,\{p_i\},t_0)[\frac{s_i}{Tot},\frac{s_i}{Tot}] \mid s_i \in S'\} \cup \{loc(id_2,\{p_i\},t_0)[\frac{s_i}{Tot},\frac{s_i}{Tot}] \mid s_i \in S\setminus S'\} \cup \{loc(id_2,\{p_1,\dots,p_n\},t_0)[1-\frac{c}{Tot},1-\frac{c}{Tot}]\}$.

We prove by contraposition that $S'$ is a maximum cardinality subset of $S$ which sums up exactly to $c$ iff $\mathcal{K}'$ is an optimal object id revision of $\mathcal{K}$.

($\Rightarrow$) Assume that $\mathcal{K}'$ is *not* an optimal object id revision of $\mathcal{K}$. Then there is $\mathcal{K}''$ such that (i) $\mathcal{K}''$ is a object id revision of $\mathcal{K}$; (ii) $\mathcal{K}'' \cup \{ra\}$ is consistent; and (iii) $d_O(\mathcal{K},\mathcal{K}'') < d_O(\mathcal{K},\mathcal{K}')$.

Let $S''$ be the set of $s_i \in S$ such that $(id_1,\{p_i\},t_0,[\frac{s_i}{Tot},\frac{s_i}{Tot}]) \in \mathcal{K}''$. As $d_O(\mathcal{K},\mathcal{K}'') = \sum_{s_i\in S\setminus S''} d_O(id_1,id_2) = \sum_{s_i\in S\setminus S''} 1 = n - |S''|$ and $d_O(\mathcal{K},\mathcal{K}'') < d_O(\mathcal{K},\mathcal{K}') = n - |S'|$, $n - |S''| < n - |S'|$, thus $|S''| > |S'|$.

We now show that $\sum_{s_i\in S''} s_i = c$. As $\mathcal{K}'' \cup \{ra\}$ is consistent,

$$\sum_{loc(id_1,\{p_i\},t_0)[\frac{s_i}{Tot},\frac{s_i}{Tot}]\in\mathcal{K}''} \frac{s_i}{Tot} \leqslant \frac{c}{Tot}.$$

Thus, $\sum_{s_i\in S''} s_i \leqslant c$. Moreover, since $c \neq Tot/2$, for every object id revision of $\mathcal{K}$, the id of atom $\{loc(id_2,\{p_1,\dots,p_n\},t_0)[1-\frac{c}{Tot},1-\frac{c}{Tot}]\}$ cannot be changed into $id_1$, because it would be inconsistent with *ra*, which requires a probability mass equal to $\frac{c}{Tot}$ in the same locations and at the same time point. As $\mathcal{K}''$ contains $\{loc(id_2,\{p_1,\dots,p_n\},t_0)[1-\frac{c}{Tot},1-\frac{c}{Tot}]\}$,

$$\sum_{loc(id_2,\{p_i\},t_0)[\frac{s_i}{Tot},\frac{s_i}{Tot}]\in\mathcal{K}''} \frac{s_i}{Tot} \leqslant 1-\frac{c}{Tot}.$$

Thus, $\sum_{s_i\in S\setminus S''} s_i \leqslant Tot - c$. As $\sum_{s_i\in S''} s_i = Tot - \sum_{s_i\in S\setminus S''} s_i$, the latter implies that $\sum_{s_i\in S''} s_i \geqslant c$. Since we have already shown that $\sum_{s_i\in S''} s_i \leqslant c$, we obtain $\sum_{s_i\in S''} s_i = c$. Since we have further shown that $|S''| > |S'|$, $S'$ is not a maximum cardinality subset of $S$ which sums up exactly to $c$.

($\Leftarrow$) Assume now that $S'$ is *not* a maximum cardinality subset of $S$ which sums up exactly to $c$. Then, there is $S'' \subseteq S$ such that $|S|'' > |S'|$ and $\sum_{s_i\in S''} s_i = c$. Let $\mathcal{K}''$ be the following object id revision of $\mathcal{K}$. $\mathcal{K}'' = \{loc(id_1,\{p_i\},t_0)[\frac{s_i}{Tot},\frac{s_i}{Tot}] \mid s_i \in S''\} \cup \{loc(id_2,\{p_i\},t_0)[\frac{s_i}{Tot},\frac{s_i}{Tot}] \mid s_i \in S\setminus S''\} \cup \{loc(id_2,\{p_1,\dots,p_n\},t_0)[1-\frac{c}{Tot},1-\frac{c}{Tot}]\}$. As $\sum_{s_i\in S''} s_i = c$, it is easy to see that $\mathcal{K}'' \cup \{ra\}$ is consistent. Since $d_O(\mathcal{K},\mathcal{K}'') = n - |S''|$ and $d_O(\mathcal{K},\mathcal{K}') = n - |S'|$ and $|S|'' > |S'|$, we have $d_O(\mathcal{K},\mathcal{K}'') < d_O(\mathcal{K},\mathcal{K}')$. Thus $\mathcal{K}'$ is not an optimal object id revision of $\mathcal{K}$. □

**Corollary 2.** *Given PST atom ra, and PST KBs $\mathcal{K}$, $\mathcal{K}'$ where $T$ is finite, deciding whether $\mathcal{K}' = \mathcal{K} \dot{+}_o ra$ is coNP-complete.*

**Proof.** (Membership) A polynomial size witness for the complement of the problem of deciding if $\mathcal{K}' = \mathcal{K} \dot{+}_o ra$ is an object id revision $\mathcal{K}''$ such that $\mathcal{K}'' \cup \{ra\}$ is consistent and either $d_O(\mathcal{K},\mathcal{K}'') < d_O(\mathcal{K},\mathcal{K}')$ or $d_O(\mathcal{K},\mathcal{K}'') = d_O(\mathcal{K},\mathcal{K}')$ and $\mathcal{K}''$ precedes $\mathcal{K}$ according the strict total order $O_T$. As in the case of the membership proof of Theorem 9, this witness can be verified in polynomial time.

(Hardness) Consider the proof of Theorem 9 and let $O_T$ be such that, for $d_O(\mathcal{K},\mathcal{K}') = |S| - |S'|$, $\mathcal{K}'$ is its minimum element. Then, reasoning as in the proof of Theorem 9, it is easy to see that $(S, c, S')$ is in MCSS iff $\mathcal{K}' = \mathcal{K} \dot{+}_o ra$. □

**Theorem 10.** *Let $\mathcal{K}$ be any knowledge base. A lower bound revision of $\mathcal{K}$ satisfying the AGM axioms is possible for every atom ra iff for all $a_i \in \mathcal{K}$, $u_i = 1$.*

**Proof.** ($\Leftarrow$) Assume that all $u_i = 1$. Then lowering all $\ell_i$ to 0 yields a lower bound revision satisfying the AGM axioms.

($\Rightarrow$) Suppose that for some $a_i \in \mathcal{K}$, $a_i = loc(id, r_i, t)[\ell_i, u_i]$ and $u_i \neq 1$. Let $ra = loc(id, r_i, t)[1, 1]$. There is no lower bound revision. □

**Proposition 4.** *$\mathcal{K} \dot{+}_p ra$ satisfies* (A1)–(A6)*.*

**Proof.** Axiom (A1) follows from the fact that there is always a solution to *PRLP*($\mathcal{K}$) — consider the solution that sets all $low_i$ to 0 and all $up_i$ to 1. (A2) follows from the inequalities specified in 2(a) of PRLP. Axiom (A3) follows from the fact that upper bounds are increased and lower bounds are decreased (according to the inequalities in 1(c) of PRLP), loosening the knowledge base such that any interpretation satisfying $\mathcal{K} \cup \{ra\}$ must also satisfy $\mathcal{K} \dot{+}_p ra$. Axiom (A4) follows from the fact that the minimum value possible for the distance function occurs when there is no change to the knowledge base. Thus if at all possible, the algorithm returns the original values for the lower and upper bounds of the knowledge base making the updated knowledge base equal to the original knowledge base. Axiom (A5) follows from the fact that any solution to *PRLP* corresponds to a consistent knowledge base. Consider if $\theta$ is a solution to *PRLP*($\mathcal{K}, ra$). $\theta$ (minus any assignments to the variables $low_i$ and $up_i$) is also a solution to *LP*($\mathcal{K} \dot{+}_p ra$). Axiom (A6) follows from the use of the strict total order. If $ra \equiv ra'$, then the solutions to *PRLP*($\mathcal{K}, ra$) will be exactly the solutions to *PRLP*($\mathcal{K}, ra'$). Thus the minimal member of both sets of solutions will be the same according to the strict total order and the same revised knowledge base will be returned. □

**Theorem 11.** *Given a PST atom ra, and PST KBs $\mathcal{K}, \mathcal{K}'$ where $T$ is finite, deciding whether $\mathcal{K}' \cup \{ra\} = \mathcal{K} \dot{+}_h ra$ is coNP-complete.*

**Proof.** (Membership) A polynomial size witness for the complement of the problem of deciding whether $\mathcal{K}' = \mathcal{K} \dot{+}_h ra$ is a hybrid revision $\mathcal{K}''$ such that $\mathcal{K}'' \cup \{ra\}$ is consistent and either $d_H(\mathcal{K}, \mathcal{K}'') < d_H(\mathcal{K}, \mathcal{K}')$ or $d_H(\mathcal{K}, \mathcal{K}'') = d_H(\mathcal{K}, \mathcal{K}')$ and $\mathcal{K}''$ precedes $\mathcal{K}$ according the strict total order $O_T$. By the finiteness of $T$, the consistency of $\mathcal{K}'' \cup \{ra\}$ can be verified in polynomial time by solving *LP*($\mathcal{K}'' \cup \{ra\}$). Moreover, conditions regarding the minimality w.r.t. $d_H()$ and $O_T$ can also be verified in polynomial time.

(Hardness) It is easy to see that the coNP-hard problem of checking whether $\mathcal{K}' = \mathcal{K} \dot{+}_o ra$ can be reduced to the problem of deciding whether $\mathcal{K}' = \mathcal{K} \dot{+}_h ra$ by defining $d_H()$ so that, $w_S = w_T = w_P = \infty$ and $w_O \neq \infty$. □

**Theorem 12.** *Let $\mathcal{K} = \{a_i \mid a_i = loc(id_i, r_i, t_i)[\ell_i, u_i]\}$, and $ra = loc(id_r, r_r, t_r)[\ell_r, u_r]$, the revision atom. $\theta$ is an optimal solution of the optimization problem*

***minimize the objective function***

$$w_O\left(\sum_{a_i\in\mathcal{K}}\sum_{id_j\in ID}\hat{id}^i_j\cdot d_O(id_i, id_j)\right) + w_S\left(\sum_{a_i\in\mathcal{K}}\sum_{r_j\in regions(a_i)}\hat{r}^i_j\cdot d_S(r_i, r_j)\right)$$

$$+ w_T\left(\sum_{a_i\in\mathcal{K}}\sum_{t_j\in T}\hat{t}^i_j\cdot d_T(t_i, t_j)\right) + w_P\left(\sum_{a_i\in\mathcal{K}}(\ell_i - low_i) + (up_i - u_i)\right)$$

***subject to***

*HRLP*($\mathcal{K}, ra$)

*iff $\beta_\theta(\mathcal{K})$ is an optimal hybrid revision of $\mathcal{K}$ by adding ra.*

**Proof.** By contraposition.

($\Rightarrow$) Suppose that $\theta$ is a solution for *HRLP*($\mathcal{K}, ra$) and $\beta_\theta(\mathcal{K})$ is not an optimal hybrid revision of $\mathcal{K}$ by adding *ra*. Thus, there is $\mathcal{K}'$ such that $\mathcal{K}' \cup \{ra\}$ is consistent and $d_H(\mathcal{K}, \beta_\theta(\mathcal{K})) > d_H(\mathcal{K}, \mathcal{K}')$. Let $\mathcal{K}' = \{a'_i = loc(id'_i, r'_i, t'_i)[\ell'_i, u'_i]\}$. Construct solution $\theta'$ for *HRLP*($\mathcal{K}, ra$) such that, $\forall a_i \in \mathcal{K}$, $\forall id_j \in ID$, $\forall t_k \in T$, and $\forall r_m \in regions(a_i)$, $\hat{id}^i_j\theta' = 1$ iff $id'_i = id_j$, $\hat{t}^i_k\theta' = 1$ iff $t'_i = t_k$, and $\hat{r}^i_m\theta' = 1$ iff $r'_i = r_m$. The constraints in (3) in Definition 22 are clearly satisfied. Moreover, if $id'_i = id_j \wedge t'_i = t_k \wedge r'_i = r_m$ is false, then equations (2) in Definition 22 are trivially satisfied by every solution of *HRLP*($\mathcal{K}, ra$), since they become $L \leqslant \sum_{p\in r_m}\sum_{q\in\mathcal{S}} v_{id_j,t_k,p,q} \leqslant U$ with $[L, U] \supseteq [0, 1]$. The non-trivial constraints of type (2) are those such that $id'_i = id_j \wedge t'_i = t_k \wedge r'_i = r_m$, obtaining $low_i \leqslant \sum_{p\in r_m}\sum_{q\in\mathcal{S}} v_{id_j,t_k,p,q} \leqslant u_i$. Thus for the fixed solution $\theta'$, Definition 22 is equivalent to *PRLP*($\mathcal{K}', ra$). As $\mathcal{K}' \cup \{ra\}$ is consistent, there is a solution $\theta''$ for *PRLP*($\mathcal{K}', ra$) such that for each $a'_i \in \mathcal{K}'$, $low_i\theta'' = \ell'_i$, and $up_i\theta'' = u'_i$. Let $low_i\theta' = \ell'_i$ and $up_i\theta' = u'_i$. Thus, $\theta'$ is a solution of *HRLP*($\mathcal{K}, ra$) such that $\beta_{\theta'}(\mathcal{K}) = \mathcal{K}'$. As $d_H(\mathcal{K}, \mathcal{K}') = \sum_{a\in\mathcal{K}} d_H(a, a') = \sum_{a\in\mathcal{K}} w_O \cdot d_O(a, a') + w_S \cdot d_S(a, a') + w_T \cdot d_T(a, a') + w_P \cdot d_P(a, a')$ and $d_H(\mathcal{K}, \beta_\theta(\mathcal{K})) > d_H(\mathcal{K}, \mathcal{K}')$, $d_H(\mathcal{K}, \beta_\theta(\mathcal{K})) > d_H(\mathcal{K}, \beta_{\theta'}(\mathcal{K}))$, i.e., $\theta$ is not an optimal solution for *HRLP*($\mathcal{K}, ra$).

($\Leftarrow$) Suppose that $\beta_\theta(\mathcal{K})$ is a hybrid revision of $\mathcal{K}$ by adding *ra* and $\theta$ is not an optimal solution for *HRLP*($\mathcal{K}$, *ra*). Thus there is a solution $\theta'$ of *HRLP*($\mathcal{K}$, *ra*) such that $d_H(\mathcal{K}, \beta_\theta(\mathcal{K})) > d_H(\mathcal{K}, \beta_{\theta'}(\mathcal{K}))$. Reasoning similarly to the previous case, it is easy to see that $\beta_{\theta'}(\mathcal{K}) \cup \{ra\}$ is consistent and since $d_H(\mathcal{K}, \beta_\theta(\mathcal{K})) > d_H(\mathcal{K}, \beta_{\theta'}(\mathcal{K}))$ this implies that $\beta_\theta(\mathcal{K})$ is not an optimal hybrid revision. □

**Proposition 5.** *If the associated reachability definition is fully connected, then for $(\mathcal{P}^{t_0}, \mathcal{P}^{t_1})$ returned by Algorithm 3, for all $P \in \mathcal{P}^{t_0}$ there is $P' \in \mathcal{P}^{t_1}$ such that P cautiously reaches $P'$ in time $t_1 - t_0$ and for all $P \in \mathcal{P}^{t_1}$ there is $P' \in \mathcal{P}^{t_0}$ such that P cautiously reaches $P'$ in time $t_1 - t_0$.*

**Proof.** The proof follows from the fact that each line in Algorithm 3 maintains the invariant that every partition in $\mathcal{P}_\star^{t_0}$ marked "done" cautiously reaches at least one partition in $\mathcal{P}_\star^{t_1}$ and every partition in $\mathcal{P}_\star^{t_1}$ marked "done" is cautiously reached by at least one partition in $\mathcal{P}_\star^{t_0}$. We can be sure that eventually the algorithm will complete because eventually $\mathcal{P}_\star^{t_0}$ and $\mathcal{P}_\star^{t_1}$ will contain only singleton partitions, where each partition represents only one point. In this degenerate case, the cautious reachability predicate equals the underlying reachability predicate, which is fully connected by assumption. Therefore every point and the associated partition in $\mathcal{P}_\star^{t_0}$ can reach some other point and its associated partition in $\mathcal{P}_\star^{t_1}$, and every point and the associated partition in $\mathcal{P}_\star^{t_1}$ is reached by some other point and its associated partition in $\mathcal{P}_\star^{t_0}$. Line 22 guarantees that eventually, either the degenerate case will eventually occur or all partitions will be marked "done". □

## References


[1] C.E. Alchourrón, P. Gärdenfors, D. Makinson, On the logic of theory change: Partial meet contraction and revision functions, Journal of Symbolic Logic 50 (1985) 510–530.
[2] A. Aziz, K. Sanwal, V. Singhal, R.K. Brayton, Verifying continuous time Markov chains, in: Proceedings of the 8th International Conference on Computer Aided Verification, Springer-Verlag, London, UK, 1996, pp. 269–276.
[3] C. Baral, S. Kraus, J. Minker, V.S. Subrahmanian, Combining knowledge bases consisting of first-order theories, Computational Intelligence 8 (1) (1992) 45–71.
[4] B. Bennett, Modal logics for qualitative spatial reasoning, Journal of the Interest Group on Pure and Applied Logic 4 (1996) 23–45.
[5] A.G. Cohn, S.M. Hazarika, Qualitative spatial representation and reasoning: An overview, Fundam. Inf. 46 (1–2) (2001) 1–29.
[6] Dana Nau, Fusun Yaman, V.S. Subrahmanian, The logic of motion, in: Proc. 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR-2004), 2004, pp. 85–94.
[7] Dana Nau, Fusun Yaman, V.S. Subrahmanian, Going far, logically, in: Proc. IJCAI 2005, 2005, pp. 615–620.
[8] Dana Nau, Fusun Yaman, V.S. Subrahmanian, A motion closed world assumption, in: Proc. IJCAI 2005, 2005, pp. 621–626.
[9] A.G. Cohn, D. Magee, A. Galata, D. Hogg, S. Hazarika, Towards an architecture for cognitive vision using qualitative spatio-temporal representations and abduction, in: C. Freksa, C. Habel, K.F. Wender (Eds.), Spatial Cognition III, in: Lecture Notes in Computer Science, vol. 2685, Springer-Verlag, 2003.
[10] David Gabelaia, Roman Kontchakov, Agi Kurucz, Frank Wolter, Michael Zakharyaschev, On the computational complexity of spatio-temporal logics, in: FLAIRS Conference, 2003, pp. 460–464.
[11] H. Hansson, B. Jonsson, Logic for reasoning about time and reliability, Formal Asp. Comput. 6 (5) (1994) 512–535.
[12] S.O. Hansson, Knowledge-level analysis of belief base operations, Artificial Intelligence 82 (1) (1996) 215–235.
[13] Samir Khuller, Maria Vanina Martinez, Dana Nau, Gerardo Simari, Amy Sliva, V.S. Subrahmanian, Computing most probable worlds of action probabilistic logic programs: Scalable estimation for $10^{30,000}$ worlds, Annals of Mathematics and Artificial Intelligence 51 (2–4) (2007) 295–331.
[14] M. Kwiatkowska, G. Normal, D. Parker, Prism: Probabilistic symbolic model checker, in: Computer Performance Evaluation: Modelling Techniques and Tools, in: Lecture Notes in Computer Science, vol. 2324, Springer-Verlag, 2002, pp. 113–140.
[15] L.G. Khachiyan, A polynomial algorithm in linear programming, Soviet Mathematics Doklady 20 (1979) 191–194.
[16] Stephan Merz, Martin Wirsing, Júlia Zappe, A spatio-temporal logic for the specification and refinement of mobile systems, in: Mauro Pezzè (Ed.), FASE, in: Lecture Notes in Computer Science, vol. 2621, Springer, 2003, pp. 87–101.
[17] R. Mittu, R. Ross, Building upon the coalitions agent experiment (coax) — integration of multimedia information in gccs-m using impact, in: Intl. Conf. on Multimedia Information Systems, 2003, pp. 35–44.
[18] Philippe Muller, Space–time as a primitive for space and motion, in: FOIS, IOS Press, Amsterdam, June 1998, pp. 63–76.
[19] Christos M. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, MA, 1994.
[20] A.J. Parker, V.S. Subrahmanian, J. Grant, A logical formulation of probabilistic spatial databases, IEEE TKDE 19 (11) (2007) 1541–1556.
[21] A.J. Parker, V.S. Subrahmanian, J. Grant, Spot databases: Efficient consistency checking and optimistic selection in probabilistic spatial databases, IEEE TKDE 21 (1) (2009) 92–107.
[22] Austin Parker, Spatial probabilistic temporal databases, PhD thesis, University of Maryland, 2008.
[23] Austin Parker, Guillaume Infantes, John Grant, V.S. Subrahmanian, An agm-based belief revision mechanism for probabilistic spatio-temporal logics, in: AAAI, 2008.
[24] Austin Parker, Fusun Yaman, Dana Nau, V.S. Subrahmanian, Probabilistic go theories, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), January 2007, pp. 501–506.
[25] R. Rajagopalan, B. Kuipers, Qualitative spatial reasoning about objects in motion: Application to physics problem solving, in: IJCAI'94, San Antonio, TX, 1994, pp. 238–245.
[26] D. Cohn, A. Randell, Z. Cui, A spatial logic based on regions and connection, in: KR199: International Conference on Knowledge Representation and Reasoning, Morgan Kaufmann, 1992, pp. 165–176.
[27] H. Samet, Foundations of Multidimensional and Metric Data Structures, Morgan Kaufman, 2006.
[28] Murray Shanahan, Default reasoning about spatial occupancy, Artif. Intell. 74 (1) (1995) 147–163.
[29] Finnegan Southey, Wesley Loh, Dana F. Wilkinson, Inferring complex agent motions from partial trajectory observations, in: IJCAI, 2007, pp. 2631–2637.
[30] B. Yetso, T. Hammel, T.J. Rogers, Fusing live sensor data into situational multimedia views, in: Proc. 2003 Intl. Conf. on Multimedia Information Systems, 2003, pp. 145–146.
[31] Frank Wolter, Michael Zakharyaschev, Spatial reasoning in rcc-8 with boolean region terms, in: ECAI2000: Principles of Knowledge Representation and Reasoning, IOS Press, Berlin, 2000, pp. 244–248.
[32] Frank Wolter, Michael Zakharyaschev, Spatio-temporal representation and reasoning based on RCC-8, in: Anthony G. Cohn, Fausto Giunchiglia, Bart Selman (Eds.), KR2000: Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Francisco, 2000, pp. 3–14.