Minimal contraction of preference relations

Denis Mindolin and **Jan Chomicki**
Dept. of Computer Science and Engineering
University at Buffalo
Buffalo, NY 14260-2000
{mindolin,chomicki}@cse.buffalo.edu

Abstract

Changing preferences is very common in real life. The expressive power of the operations of preference change introduced so far in the literature is limited to *adding* new information about preference and equivalence. We discuss the operation of *discarding* preferences - preference contraction. We argue that the property of *minimality* and the preservation of *strict partial orders* are crucial for contractions. Contractions can be further constrained by specifying which preferences *should not* be contracted. We provide algorithms for computing minimal and minimal preference-protecting contraction. We also show some preference query optimization techniques which can be used in the presence of contraction.

Content areas: Knowledge Representation, Knowledge Engineering, Databases

1 Introduction

A number of preference representation and reasoning frameworks have been developed. Among the most popular ones are *CP-nets* (Boutilier *et al.* 2004) and the *binary relation* framework (Chomicki 2003; Kießling 2002). In the CP-net framework, preferences are represented as graphs. This framework is simple and intuitive, but the expressive power of the framework is limited. A number of extensions to that model have been introduced (Brafman, Domshlak, & Shimony 2002; Wilson 2004).

In the *binary relation* framework, preferences are represented as binary relations over objects. Preferences in this framework are *strict partial orders (SPO)*: transitive and irreflexive binary relations. The SPO properties are known to capture the rationality of preferences (Fishburn 1970). This framework can deal with finite as well as infinite preference relations, the latter represented using finite *preference formulas*. Connections between these two frameworks have been recently established in (Endres & Kießling 2006; Mindolin & Chomicki 2007), where it was shown how some variants of CP-nets can be represented as preference formulas. The binary relation framework is the focus of our paper.

Working with preferences in any framework, it is naive to expect that they never change. Preferences can change over time: if one likes something now, it does not mean one will still like it in the future. It was shown in (Doyle 2004) that along with the discovery of sources of preference change and elicitation of the change itself, it is important to preserve the correctness of preference model in the presence of change. In the binary relation framework, a natural correctness criterion is the preservation of SPO properties of preference relations.

Two SPO-preserving operations of preference change in the binary relation framework have been proposed in the literature: *preference revision* (Chomicki 2007a) and *equivalence adding* (Balke, Guntzer, & Siberski 2006). Informally, preference revision is defined as follows. Let $\succ_0$ be the initial preference relation which generally represents the user preferences learned so far. Let $\succ_1$ be a *revising relation* consisting of new preferences generally corresponding to the information learned from the user or provided by her directly. Then the revised preference relation is the least SPO preference relation which contains a *composition* of $\succ_0$ and $\succ_1$. The composition operators used in (Chomicki 2007a) are: union composition, prioritized composition, and Pareto composition.

The *equivalence adding* operation (Balke, Guntzer, & Siberski 2006) is defined as follows. Let $\succ_0$ be the user preferences learned so far. Let eq be an equivalence relation over objects. Then the preference relation $\succ_0$ with added equivalence eq is the least preference relation which contains $\succ_0$ and for which the pairs of objects eq are *equivalent*. (Balke, Guntzer, & Siberski 2006) discusses several definitions of equivalence.

The two operations above assume that changing preferences can be done only by adding new preference or equivalence information. However, these are not the only ways people change their preferences in real life. For instance, it is common to *discard* some preferences one used to hold if the reason for holding those preferences is no longer valid. That is, given the initial preference relation $\succ$ and a subset CON of the initial preference relation, we want the new preference relation *not to contain* the relation CON. None of the operations above allow this kind of change.

Example 1 *Assume that Mary wants to buy a car and her preference over cars is that a good car should be as new as possible. Such preference can be represented as the following preference relation*

$$o_1 \succ_1 o_2 \equiv o_1.y > o_2.y$$

The information about all cars which are in stock now is *shown in the table below:*

id	*make (m)*	*year (y)*	*price (p)*
t_1	*vw*	*2007*	*15000*
t_2	*bmw*	*2007*	*20000*
t_3	*kia*	*2006*	*15000*
t_4	*kia*	*2007*	*12000*

Then the set of the most preferred cars according to $\succ_1$ *is* $S_1 = \{t_1, t_2, t_4\}$.

Assume that having examined the set S_1*, Mary decides to revise her preferences: among the cars made in the same year, she prefers cheaper ones. So the new preference is represented as* $\succ_2$*:*

$$o_1 \succ_2 o_2 \equiv o_1.y > o_2.y \vee o_1.y = o_2.y \wedge o_1.p < o_2.p$$

and the set of the best cars according to $\succ_2$ *is* $S_2 = \{t_4\}$.

Assume that having observed the set S_2*, Mary understands that it is too narrow. She decides that the car* t_1 *is not really worse than* t_4*. She generalizes that by stating that the cars made in* 2007 *which cost* 12000 *are not better than the cars made in* 2007 *costing* 15000*. So* t_4 *is not preferred to* t_1 *any more, and thus the set of the best cars according to the new preference relation should be* $S_3 = \{t_1, t_4\}$.

The problem which we face here is how to represent the preference relation $\succ_2$ *with that change? Namely, we want to find a preference relation obtained from* $\succ_2$ *in which certain preferences do not hold. A naive solution is to represent the new preference as* $\succ_3 \equiv (\succ_2 - CON)$*, where* $CON(o_1, o_2) \equiv o_1.y = o_2.y = 2007 \wedge o_1.p = 12000 \wedge o_2.p = 15000$*, i.e.* CON *is the preference we want to discard. So*

$$o_1 \succ_3 o_2 \equiv (o_1.y > o_2.y \vee o_1.y = o_2.y \wedge o_1.p < o_2.p) \wedge \neg(o_1.y = o_2.y = 2007 \wedge o_1.p = 12000 \wedge o_2.p = 15000).$$

However, $\succ_3$ *is not transitive since if we take* $t_5 = (bmw, 2007, 12000)$*,* $t_6 = (bmw, 2007, 14000)$*, and* $t_7 = (bmw, 2007, 15000)$*, then* $t_5 \succ_3 t_6$ *and* $t_6 \succ_3 t_7$ *but* $t_5 \not\succ_3 t_7$*. So this change does not preserve SPO. Thus, to make the changed preference relation transitive, some other preferences have to be discarded in addition to* CON*. At the same time, discarding too many preferences is not a good solution since they may be important. So we need to discard a minimal part of* $\succ_2$ *which contains* CON *and preserves SPO of the modified preference relation.*

An SPO preference relation which is minimally different from $\succ_2$ *and does not contain* CON *is shown below:*

$$o_1 \succ'_3 o_2 \equiv (o_1.y > o_2.y \vee o_1.y = o_2.y \wedge o_1.p < o_2.p) \wedge \neg(o_1.y = o_2.y = 2007 \wedge o_1.p = 12000 \wedge o_2.p > 12000 \wedge o_2.p \leq 15000)$$

The set of the best cars according to $\succ'_3$ *is* $S'_3 = \{t_1, t_4\}$*. As we can see, the relation* $\succ'_3$ *is different from the naive solution* $\succ_3$ *in the sense that* $\succ'_3$ *implies that a car made in* 2007 *costing* 12000 *is not better than a car made in* 2007 *costing* from 12000 to 1500.

The operation of discarding preferences, *preference contraction*, is the topic of this paper. As we showed in Example 1, when discarding preferences, it is important not to discard more preferences than it is necessary to preserve SPO.

However, the preference relation $\succ'_3$ shown in Example 1 is not the only possible SPO minimally different from $\succ_2$ which is disjoint with CON, and there exists an infinite number of such preference relations. Each of them discards different sets of preferences in addition to CON. At the same time, some preferences discarded in addition to CON may be important for the user, so he or she may want to keep them in the contracted preference relation. This observation motivates the operation of *preference-protecting minimal contraction* which we introduce in the paper. That is, in addition to providing the preferences to be discarded, one can also provide the preferences to be *protected from removal* in the modified preference relation.

The problem we tackle in the paper is *finding minimal contractions of preference relations which preserve SPO*. The main results of the paper are as follows. First, we present necessary and sufficient conditions of the minimal and the minimal preference-protecting contractions. Second, we provide algorithms to compute these contractions. Finally, we show how to optimize preference query evaluation with the presence of contraction.

2 Basic Notions

The preference relation framework we use in the paper is based on (Chomicki 2003).

Let $\mathcal{U}$ be a universe of *objects* each of each having a fixed set of *attributes* $\mathcal{A} = \{A_1, ...A_m\}$. Let each attribute A_i be associated with a *domain* $\mathcal{D}_i$. We consider here two kinds of infinite domains: *C* (uninterpreted constants) and *Q* (rational numbers).

Binary relations $R \subseteq \mathcal{U} \times \mathcal{U}$ considered in the paper are *finite* or *infinite*. Finite binary relations are represented as sets of pairs of objects. The infinite binary relations we consider here are *finitely representable* as *formulas*. Given a binary relation C, its formula representation is denoted as F_C.

We consider two kinds of atomic formulas here:

- *equality constraints*: $o_1.A_i = o_2.A_i$, $o_1.A_i \neq o_2.A_i$, $o_1.A_i = c$, or $o_1.A_i \neq c$, where o_1, o_2 are object variables, A_i is a *C* -attribute, and c is an uninterpreted constant;
- *rational-order constraints*: $o_1.A_i \theta o_2.A_i$ or $o_1.A_i \theta c$, where $\theta \in \{=, \neq, <, >, \leq, \geq\}$, o_1, o_2 are object variables, A_i is a *Q* -attribute, and c is a rational number.

An example of a relation represented using rational-order constraints is $\succ_2$ from Example 1.

Another way to represent binary relations is by using *graph* notation, as we show in the next definition.

Definition 1 *Given a binary relation* $R \subseteq \mathcal{U} \times \mathcal{U}$ *and two objects* x *and* y *such that* xRy *(*$xy \in R$*),* xy is an R-edge from x to y*. Similarly, we can define a finite* R-path from x to y *and an infinite* R-path from x.

Preference relations in our framework are defined as follows.

Definition 2 *A binary relation $\succ \subset \mathcal{U} \times \mathcal{U}$ is a* preference relation, *if it is a strict partial order (SPO) relation, i.e. transitive and irreflexive. The formula representation $F_\succ$ of $\succ$ is called a* preference formula.

An element of a preference relation is called *a preference*. We use the symbol $\succ$ to refer to preference relations. The following expression $o_1 \succeq o_2$ is a shortening for $(o_1 \succ o_2 \lor o_1 = o_2)$.

3 Preference contraction

The key notion of preference contraction is the *contracting relation* which defines the set of pairs of objects such that the first object in each pair should not be preferred to the second object. We require the contracting relation to be a subset of the preference relation to be contracted. Apart from that, we do not impose any other restrictions on contracting relations (i.e. they can be finite of infinite) unless stated otherwise. Throughout the paper, all contracting relations are denoted by CON.

Definition 3 *A binary relation P^- is* a contraction of a preference relation $\succ$ by CON *if $CON \subseteq P^- \subseteq \succ$, and $(\succ - P^-)$ is a preference relation (i.e. an SPO). The relation $(\succ - P^-)$ is called the* contracted relation.

A relation P^* is a minimal contraction of $\succ$ by CON *if P^* is a contraction of $\succ$ by CON, and there is no other contraction P' of $\succ$ by CON s.t. $P' \subset P^*$.*

The notion of minimal contraction narrows the set of preference contractions. However, as we illustrate in Example 2, minimal preference contraction is generally not unique for given preference and contracting relations. In fact, the number of minimal contractions for infinite preference relations can be infinite. This differs from minimal preference revision (Chomicki 2007a) which is uniquely defined for given preference and revising relations.

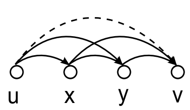


Figure 1: Preference $\succ$.

Example 2 *Take the preference relation $\succ$ as shown in Figure 1 as the set of all edges, and the contracting relation $CON = \{uv\}$. Then there are three possible minimal contractions of $\succ$ by CON: $P_1^- = \{ux, uy, uv\}$, $P_2^- = \{yv, xv, uv\}$, and $P_3 = \{ux, yv, uv\}$.*

3.1 Contraction conditions

Definition 4 *Given a contracting relation CON of a preference relation $\succ$, a $\succ$-path from x to y is a CON*-detour *if $xy \in CON$.*

First, let us consider the problem of finding any preference contraction, not necessary minimal. As we showed in Example 1, the naive solution of computing the set difference of $\succ$ and CON does not preserve SPO. We formulate below a necessary and sufficient condition for a subset of a preference relation to be its contraction.

Lemma 1 *Given a preference relation (i.e. an SPO) $\succ$ and a relation $P^- \subseteq \succ$, $(\succ - P^-)$ is a preference relation (i.e. an SPO) iff for every $xy \in P^-$, $(\succ - P^-)$ contains no paths from x to y.*

Now let us consider minimal preference contractions. For instance, take the minimal contraction from Example 2. Note that adding any edge from a minimal contraction to the contracted relation creates a CON-detour in the contracted relation. However, having CON-detours in the contracted relation violates its transitivity by Lemma 1. This property of minimal contractions is formally stated in Theorem 1.

Theorem 1 *Let P^- be a contraction of $\succ$ by CON. Then P^- is a* minimal *contraction of $\succ$ by CON iff for every $xy \in P^-$, there is a CON-detour T in $\succ$ which contains the edge xy and no other edge in T is in P^-.*

Or, in other words, for any edge in P^-, there exists at least one CON-detour which is disconnected only by that edge.

In fact, the condition from Theorem 1 can be stated in terms of paths of length 3 due to the transitivity of $\succ$.

Corollary 1 *A contraction P^- of $\succ$ by CON is* minimal *iff the formula*

$$\begin{aligned}\forall x, y\, \exists u, v (\ & F_{P^-}(x, y) \land F_{CON}(u, v) \land F_\succ(x, y) \land \\ & \neg F_{P^-}(u, x) \land \neg F_{P^-}(y, v) \land \\ & (F_\succ(u, x) \lor u = x) \land (F_\succ(y, v) \lor y = v))\end{aligned}$$

is valid.

Therefore, checking minimality of a contraction can be done by performing quantifier elimination on the above formula.

3.2 Construction of minimal contraction

In the algorithm computing a minimal preference contraction introduced in this section, we use the following idea. Take Example 2 and the set P_1^-. That set was constructed as follows: we took the CON-edge uv and put in P_1^- all the edges which start some path from u to v. For the preference relation $\succ$ from Example 2, P_1^- turned out to be a minimal contraction.

Generally, if CON contains more than one edge, the set consisting of all edges starting CON-detours is a contraction by CON.

Lemma 2 *Let $\succ$ be a preference relation and CON be a contracting relation of $\succ$. Then*

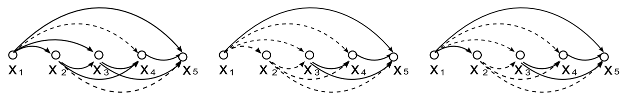
$$P^- := \{\, xy \mid \exists xv \in CON\ .\ x \succ y \land (y \succ v \lor y = v)\}$$

is a contraction of $\succ$ by CON.

However, in the next example we show that such contraction is not always minimal.

Example 3 *Take the preference relation $\succ$ as shown in Figure 2(a) as the set of all edges, and the contracting relation CON shown as the dashed edges.*

Let P^- be defined as in Lemma 2. Then $(\succ - P^-)$ is shown in Figure 2(b) by the solid edges. P^- is not minimal because $P^- - \{x_1x_2\}$ is also a contraction of $\succ$ by CON. In fact, $P^- - \{x_1x_2\}$ is a minimal *contraction of $\succ$ by CON.*



(a) Preference $\succ$ (b) $\succ - P^-$. (c) Minimally contracted $\succ$.

Figure 2: Preference contraction

As we can see, having the edge x_1x_2 in P^- was not necessary. First, it is not a CON-edge. Second, the CON-detour $x_1 \succ x_2 \succ x_4$ is already disconnected by $x_2x_4 \in P^-$.

As we show in Example 3, a minimal contraction can be constructed by adding to it only the edges which start some CON-detour if the detour is not already disconnected. We follow this idea in Algorithm 1. The algorithm returns a minimal preference contraction by a contracting relation CON under the condition that CON is a *k-layer* relation defined as follows.

Definition 5 *A* layer index *of an edge $xy \in CON$ is* the maximum length of a $\succ$-path started by y and consisting of the end nodes of CON-edges. *A* layer *is the set of all CON-edges with the same* layer index.

Then CON is a k-layer *relation if*

$$max_{xy \in CON}(\textit{layer index of } xy) \leq k$$

We need the *k-layer* property in the algorithm to be able to partition CON into layers and then process the layers one by one.

Example 4 *Let a preference relation $\succ$ be defined be defined as $o_1 \succ o_2 \equiv o_1.p < o_2.p.$, where p is a Q -attribute.*

Let also the contracting relations CON_1 and CON_2 be defined as

$$CON_1(o_1, o_2) \equiv o_1.p < 1 \wedge (o_2.p = 2 \vee o_2.p = 3).$$
$$CON_2(o_1, o_2) \equiv o_1.p < 1 \wedge o_2.p \geq 2$$

Then CON_1 is a k-layer *relation since there exists only one chain $o_1 \succ o_2$ of the end nodes of CON_1, where $o_1.p = 2$ and $o_2.p = 3$. The length of this chain is 2.*

The relation CON_2 is not k-layer *since all $\succ$-paths started by objects with the value of p equal to 2 are infinite.*

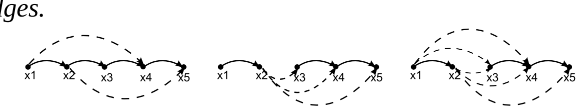
Algorithm 1 constructs a contraction P^- of CON by picking the layers of CON in the ascending order of their *layer index*. For each layer, we add to P^- a minimal set of $\succ$-edges which contract $\succ$ by the CON-edges of that layer.

Theorem 2 *Algorithm 1 returns a minimal contraction of $\succ$ by CON and halts in k iterations for a* k-layer *relation CON.*

Algorithm 1 minContr($\succ$, CON)

$i = 0$, $P_0^- = \emptyset$, $C_0 = CON$
repeat
  $i := i + 1$;
  {Find the dest. nodes of the i-th layer CON-edges}
  $L_i := \{\, y \mid \exists x(xy \in C_{i-1} \wedge \neg \exists uv \in C_{i-1}(y \succ v))\}$
  {Find the edges contracting the i-th layer of CON}
  $E_i := \{xy \mid \exists v \in L_i(xv \in CON \wedge x \succ y \wedge (y \succ v \vee y = v) \wedge yv \notin P_{i-1}^- \wedge yv \notin CON)\}$
  $P_i^- := P_{i-1}^- \cup E_i$ {Add these edges to P_{i-1}^-}
  $C_i := C_{i-1} - E_i$
until $C_i = \emptyset$
return P_i^-

Example 5 *Let a preference relation $\succ$ be defined by the solid edges in Figure 3(a). The transitive edges are skipped. Let a contracting relation CON be defined by the dashed edges.*



(a) Preference $\succ$ (b) $\succ$ after Step 1 (c) $\succ$ after Step 2

Figure 3: Preference contraction

Then the result of applying the first step of the algorithm is shown in Figure 3(b). Namely, $L_1 = \{x_5\}$, $P_1^- = \{x_2x_3, x_2x_4, x_2x_5\}$. At the second iteration (Figure 3(c)), $L_2 = \{x_4\}$ and $P_2^- = P_1^- \cup \{x_1x_3, x_1x_4\}$. At the third (and the last) iteration, $L_3 = \emptyset$, i.e. all CON-edges are already processed.

We believe that the k-layer restriction is not too severe because in many cases CON is provided as a finite set of object pairs. Such relations are k-layer by definition.

The *k-layer* property of CON is crucial for the algorithm since it guarantees its termination. If CON is not a *k-layer* relation, then the algorithm is incomplete: it misses some infinite descending paths, i.e. returns a minimal contraction by a *subset* of CON, or fails to terminate.

An important property of Algorithm 1 is that it works for finite as well as finitely representable infinite preference relations. Our implementation for finite relations (Appendix 10) requires time $O(|CON|^2 \cdot |\succ| \cdot log(|\succ|))$. In the case of finitely representable preference relations, the sets L_i, E_i, P_i^-, and C_i have to be replaced with the corresponding formulas F_{L_i}, F_{E_i}, $F_{P_i^-}$, and F_{C_i}; all the set operations have to be replaced with the corresponding boolean connectives; and quantifier elimination should be used to compute F_{L_i} and F_{E_i}.

We also note that any contraction P^- generated by Algorithm 1 has the property that any edge in P^- starts a CON-detour in $\succ$. We call such contractions *prefix contractions*.

4 Preference-protecting contraction

Generally, it is not always the case that all minimal contractions are equivalent from the point of view of users. For instance, a contraction may discard some preferences (in addition to CON) which the user does not want to discard. Thus, in addition to specifying a contracting relation CON, a subset P^+ of the original preference relation to be protected in the contracted preference relation may also be specified. Such a relation is complementary w.r.t. the contracting relation: the relation CON defines the preferences to discard whereas the relation P^+ defines the preferences to protect.

Such a situation often arises in real life. For instance, some preferences P^+ may be more important than others, so P^+ should hold after contraction. Moreover, in many iterative preference modification frameworks, P^+ is the set of the recently introduced preferences meaning that the old preferences are less relevant and thus may be dropped.

Definition 6 *Let $P^+ \subseteq \succ$. Then P^* is* a minimal contraction of $\succ$ by CON that protects P^+ *if 1) P^* is a minimal contraction of $\succ$ by CON, and 2) $P^* \cap P^+ = \emptyset$.*

4.1 Contraction conditions

Given any contraction P^- of $\succ$ by CON, by Lemma 1, P^- must contain at least one edge from every CON-detour. Thus, if P^+ contains a whole CON-detour, protecting P^+ in a contraction of $\succ$ by CON is not possible. The same holds for minimal contractions, too.

Theorem 3 *Let CON be a k-layer contracting relation, and $P^+ \subset \succ$. There exists a minimal contraction of $\succ$ by CON that protects P^+ iff $P^+_{TC} \cap CON = \emptyset$, where P^+_{TC} is the transitive closure of P^+.*

4.2 Construction of minimal preference-protecting contraction

A naive way of computing a minimal preference-protecting contraction is to find a minimal contraction P^- of $(\succ - P^+)$ and then add P^+ to P^-. However, $(\succ - P^+)$ is not an SPO in general, thus preserving SPO in $(P^- \cup P^+)$ becomes problematic.

The algorithm we propose here is a reduction to the minimal contraction algorithm shown in the previous section. First, we find a contracting relation CON' such that contracting $\succ$ by CON' is equivalent to contracting $\succ$ by CON with protected P^+. After that, we use Algorithm 1 to contract $\succ$ by CON'.

The intuition beyond the algorithm is as follows. Take any minimal *prefix* contraction P^- of $\succ$ by CON. The prefix property implies that if P^+-edges do not start CON-detours in $\succ$, then $P^- \cap P^+ = \emptyset$ and thus P^- is a minimal contraction which protects P^+. However, if P^+ contains edges starting CON-detours, then any P^+-protecting contraction has to contain the set Q defined in the next proposition.

Proposition 1 *Take any $P^+ \subset \succ$. Then any contraction of $\succ$ by CON protecting P^+ contains the set Q*

$$Q = \{xy \mid \exists u : u \succ x \succ y \wedge uy \in CON \wedge ux \in P^+\}$$

We show further that if P^+ is transitive and P^- a minimal prefix contraction of $\succ$ by $CON \cup Q$, then P^- protects P^+. Finally, we show that such P^- is also minimal w.r.t. not only $CON \cup Q$ but also CON.

Algorithm 2 minContrProt($\succ, CON, P^+$)

Require: P^+ is transitive
1: $Q = \{xy \mid \exists u : u \succ x \succ y \wedge uy \in CON \wedge ux \in P^+\}$
2: $CON' = CON \cup Q$
3: $P^- = minContr(\succ, CON')$
4: **return** P^-

Theorem 4 *If CON is a* k-layer *contracting relation, and P^+ is transitive, then Algorithm 2 terminates and returns a contraction of $\succ$ by CON which 1) is minimal, and 2) protects P^+.*

Note that we use the function `minContr` in Algorithm 2 because CON' is a k-layer relation. It is explained by the fact that CON is a k-layer relation and the set of the end nodes of CON' edges coincides with the corresponding set for CON by the construction of Q.

As in the case of Algorithm 1, Algorithm 2 can be used to find contractions of finite and finitely representable preference relations.

5 Query evaluation in database framework

Dealing with preferences, the two common tasks are 1) given two objects, find the more preferred one, and 2) find the most preferred objects in a set. The former problem is solved easily given the preference relation. To solve the later problem, the *winnow operator* is proposed in (Chomicki 2003). It picks from a given set of objects the most preferred objects according to a given preference relation. A number of optimization methods to evaluate queries involving winnow have been introduced (Chomicki 2007b; Hafenrichter & Kießling 2005).

Definition 7 *Let $\mathcal{U}$ be a universe of objects each of each having the set of attributes $\mathcal{A}$. Let $\succ$ be a preference relation over $\mathcal{U}$. Then the* winnow operator *is written as $w_\succ(\mathcal{U})$, and for every finite subset r of $\mathcal{U}$:*

$$w_\succ(r) = \{t \in r \mid \neg\exists t' \in r.t' \succ t\}$$

In this section, we show some new techniques which can be used to optimize evaluation of the winnow operator under contracted preferences. The results below are represented in terms of the standard *relational algebra* operator *selection* denoted as $\sigma_F(r)$. It picks from the object set r all the objects for which the condition F holds. The condition F is a boolean expression involving comparisons between attribute names and constants.

In user-guided preference modification frameworks (Chomicki 2007a; Balke, Guntzer, & Siberski 2006), it is assumed that users alter their preferences after examining sets of the most preferred objects returned by winnow. Thus, if preference contraction is incorporated into such frameworks, there is a need to compute winnow under contracted preference relations. Here we show how the evaluation of winnow can be optimized in such cases.

Let $\succ$ be a preference relation, CON be a contracting relation of $\succ$, and $\succ'$ be a contraction of $\succ$ by CON. Denote the set of the starting and the ending objects of CON-edges as $S(CON)$ and $E(CON)$ correspondingly.

$$S(CON) = \{x \mid \exists xy \in CON\}$$
$$E(CON) = \{y \mid \exists xy \in CON\}$$

Similarly, define the sets $S(P^-)$ and $E(P^-)$. Let us also define the set $M(CON)$ of the objects which participate in CON-detours in $\succ$

$$M(CON) = \{y \mid \exists x, y, z \,.\, x \succ y \wedge xz \in CON \wedge (y \succ z \vee y = z)\}.$$

Assume we also know quantifier-free formulas $F_{S(P^-)}$, $F_{E(P^-)}$, $F_{M(CON)}$, and $F_{S(CON)}$ representing these sets. Then the following holds.

Proposition 2

1. $w_{\succ}(r) \subseteq w_{\succ'}(r)$
2. *If* $\sigma_{F_{S(P^-)}}(w_{\succ}(r)) = \emptyset$*, then* $w_{\succ}(r) = w_{\succ'}(r)$.
3. $w_{\succ'}(r) = w_{\succ'}(w_{\succ}(r) \cup \sigma_{F_{E(P^-)}}(r))$
4. *If* P^- *is a minimal contraction, then* $w_{\succ'}(r) = w_{\succ'}(w_{\succ}(r) \cup \sigma_{F_{M(CON)}}(r))$
5. *If* P^- *is a prefix contraction, then* $\sigma_{F_{S(P^-)}}(r) = \sigma_{F_{S(CON)}}(r)$

According to Proposition 2, the result of winnow under a contracted preference is always a superset of the result of winnow under the original preference. This is caused by the fact that if we reduce the set of preference edges, the set of undominated objects can only grow.

In the second case, the contraction does not change the result of winnow. Running the winnow query is generally expensive, thus one can first evaluate the specified selection query over the computed result of the original winnow. If the result is empty, then computing the winnow under the contracted preference relation is not needed. The reasoning here is as follows. Take the preference relation $\succ$. Then for any dominated object $o \in r$ there is an object $o' \in w_{\succ}(r)$ dominating o. However, if o is in $w_{\succ'}(r)$ then o' does not dominate o in $\succ'$. Thus some $\succ$-edges going from $w_{\succ}(r)$ are lost in $\succ'$.

The third statement of the proposition is useful when the set r is large and thus running $w_{\succ'}$ over the whole set r is expensive. Instead, one can compute $\sigma_{F_{E(P^-)}}(r)$ and then evaluate $w_{\succ'}$ over $(\sigma_{F_{E(P^-)}}(r) \cup w_{\succ}(r))$ (assuming that $w_{\succ}(r)$ is already known). However, if the size of the formula $F_{E(P^-)}$ is too large, then running $\sigma_{F_{E(P^-)}}(r)$ may be also expensive. In this case, one can use a superset of $\sigma_{F_{E(P^-)}}(r)$, for example $\sigma_{F_{M(CON)}}(r)$.

It may be the case that the size of $F_{S(P^-)}$ is large and thus evaluation of $\sigma_{F_{S(P^-)}}(r)$ is expensive. Then, if P^- is a *prefix contraction*, one can use $F_{S(CON)}$ instead of $F_{S(P^-)}$.

6 Related and future work

A general framework of preference change is proposed in (Hansson 1995). Preference change there is considered from the point of view of belief change theory. In addition to contraction, it introduces the operators of revision, domain expansion and reduction. Preference contraction is defined via preference revision. Similarly to our definition, the preference contraction from (Hansson 1995) preserves rationality postulates (e.g. transitivity) and performs minimal change of preferences. However, due to the generality of the framework, the postulate set and the measure of minimality are not fixed. (Hansson 1995) defines contraction only for finite domains and does not provide any methods of computing contractions. There is also no notion of preference-protecting contraction.

(Dong *et al.* 1999) proposes algorithms of incremental maintenance of the transitive closure of graphs using relational algebra. The graph modification operations are edge insertion and deletion. Transitive graphs in (Dong *et al.* 1999) consist of two kinds of edges: the edges of the original graph and the edges induced by its transitive closure. When an edge xy of the original graph is contracted, the algorithm also deletes all the transitive edges uv such that all the paths from u to v in the original graph go through xy. As a result, such contraction is not minimal according to our definition of minimality. Moreover, (Dong *et al.* 1999) considers only finite graphs, whereas our algorithms can work with infinite relations.

Other preference modification operations are proposed in (Chomicki 2007a) and (Balke, Guntzer, & Siberski 2006). However, they do not address preference contraction.

In this paper, we consider only one kind of contraction constraints - preference protection. However, other constraints are also feasible. For instance, one could require that if a contraction protects a preference relation P_1^+ then it should protect P_2^+. Another direction is to design contraction algorithms which are not limited to k-layer contracting relations. Since other preference models (e.g. CP-nets) can be represented in the binary relation framework, an interesting direction is to apply our results in those frameworks.

Appendix 1. Proof of Lemma 1

Lemma 1. *Given a preference relation (i.e. an SPO)* $\succ$ *and a relation* $P^- \subseteq \succ$, $(\succ - P^-)$ *is a preference relation (i.e. an SPO) iff for every* $xy \in P^-$, $(\succ - P^-)$ *contains no paths from* x *to* y.

Proof

$\Leftarrow$ Prove that if for all $xy \in P^-$, $(\succ - P^-)$ contains no paths of from x to y, then $(\succ - P^-)$ is an SPO. Clearly, since $\succ$ is irreflexive, $(\succ - P^-)$ is irreflexive, too. Prove that $(\succ - P^-)$ is transitive. If $(\succ - P^-)$ is not transitive, then there are exist such objects x, y, z that $xz \notin (\succ - P^-)$ but $xy, yz \in (\succ - P^-)$. Thus $xz \in P^-$ and there is a path in $(\succ - P^-)$ from x to z consisting of two edges xy and yz. However, this contradicts to the assumption that there is no path in $(\succ - P^-)$ from x to z for every $xz \in P^-$.

$\Rightarrow$ Prove that if $(\succ - P^-)$ is an SPO, then $(\succ - P^-)$ contains no paths from x to y for every $xy \in P^-$. Clearly, if $xy \notin (\succ - P^-)$ but there is a path from x to y in $(\succ - P^-)$, then $(\succ - P^-)$ is not transitive.

Appendix 2. Proof of Theorem 1

Before going to the proof of Theorem 1, let us define the set Φ which we use in the proof of the theorem.

Definition 8 *Let CON be a contracting relation of a preference relation $\succ$, and P^- be a contraction of $\succ$ by CON. Fix any edge $xy \in P^- - CON$. Let*

- $\Phi_0(xy) = \{xy\}$*;*
- $\Phi_i(xy) = \{u_i v_i \in P^- | \exists u_{i-1}v_{i-1} \in \Phi_{i-1}(xy)$
  $(u_i = u_{i-1} \wedge v_{i-1} \succ v_i \wedge v_{i-1}v_i \notin P^- \bigvee$
  $u_i \succ u_{i-1} \wedge v_{i-1} = v_i \wedge u_i u_{i-1} \notin P^-)\}$*;*

Then $\Phi(xy)$ is defined as

$$\Phi(xy) = \bigcup_{i=0}^{\infty} \Phi_i(xy).$$

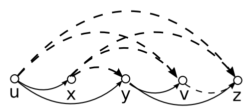


Figure 4: $\Phi(xy)$ for Example 6.

Example 6 *Let a preference relation $\succ$ be the set of all edges in Figure 4 and P^- be defined by the dashed edges. Let us construct $\Phi(xy)$ (assuming that xy is not an edge of the contracting relation).*

- $\Phi_0(xy) = \{xy\}$*;*
- $\Phi_1(xy) = \{xv, xz\}$*;*
- $\Phi_2(xy) = \{uv, uz\}$*;*

So $\Phi(xy) = \{xy, xv, xz, uv, uz\}$.

Some properties of the set Φ are shown in Lemma 3 .

Lemma 3 *Let P^- be a contraction of a preference relation $\succ$ by a contracting relation CON. Then for every $xy \in (P^- - CON)$, $\Phi(xy)$ has the following properties:*

1. *for all $uv \in \Phi(xy)$, $u \succeq x$ and $y \succeq v$;*
2. *for all $uv \in \Phi(xy)$, $ux, yv \notin P^-$;*
3. *if $(\Phi(xy) \cap CON) = \emptyset$, then $(P^- - \Phi(xy))$ is a contraction of $\succ$ by CON.*

Proof

Properties 1 and 2. We prove the first two properties by induction on i.

Base case. For every $uv \in \Phi_0(xy)$ the properties hold by the construction of Φ_0.

Inductive case. Let the properties hold for $\Phi_i(xy)$, i.e.

$$\forall u_i v_i \in \Phi_i(xy) \rightarrow u_i \succeq x \wedge y \succeq v_i \wedge u_i x, y v_i \notin P^- \quad (1)$$

Prove that these properties hold for every $u_{i+1}v_{i+1} \in \Phi_{i+1}(xy)$. Since $u_{i+1}v_{i+1} \in \Phi_{i+1}(xy)$, the following is true

$$\begin{aligned} \exists u_i v_i \in & \Phi_i(xy)(\\ & u_{i+1} = u_i \wedge v_i \succ v_{i+1} \wedge v_i v_{i+1} \notin P^- \vee \\ & u_{i+1} \succ u_i \wedge v_i = v_{i+1} \wedge u_{i+1} u_i \notin P^-) \end{aligned} \quad (2)$$

Prove that $u_{i+1} \succeq x$ and $ux \notin P^-$ (the case of yv_{i+1} is similar). Show that these statements hold for each disjunct from (2).

Case 1. Take the first disjunct from (2), i.e.

$$u_{i+1} = u_i \wedge v_i \succ v_{i+1} \wedge v_i v_{i+1} \notin P^- \quad (3)$$

Then (1) and (3) imply $u_{i+1} \succeq x$ and $u_{i+1}x \notin P^-$.

Case 2. Take the second disjunct from (2). So

$$u_{i+1} \succ u_i \wedge v_i = v_{i+1} \wedge u_{i+1} u_i \notin P^- \quad (4)$$

First, (1) and (4) imply $u_{i+1} \succ u_i$ and $u_i \succeq x$. Thus $u_{i+1} \succ x$ by transitivity of $\succ$. Second, (1) and (4) imply $u_i x \notin P^-$ and $u_{i+1}u_i \notin P^-$. Thus, $u_{i+1}x \notin P^-$ by transitivity of $(\succ - P^-)$.

Property 3. Show that if $\Phi(xy) \cap CON = \emptyset$, then $(P^- - \Phi(xy))$ is a contraction of $\succ$ by CON.

First, $(\Phi(xy) \cap CON) = \emptyset$ and $CON \subseteq P^-$ imply $CON \in (P^- - \Phi(xy))$. Hence, we only need to prove that $(\succ - (P^- - \Phi(xy)))$ is transitive (its irreflexivity follows from the irreflexivity of $\succ$).

Intransitivity of $(\succ - (P^- - \Phi(xy)))$ means

$$\begin{aligned} \exists u, v, z(uz \notin (\succ - (P^- - \Phi(xy))) \wedge \\ uv, vz \in (\succ - (P^- - \Phi(xy)))) \end{aligned} \quad (5)$$

(5) implies $uz \notin (\succ - P^-)$. Thus, by transitivity of $(\succ - P^-)$, we get

$$uv \notin (\succ - P^-) \vee vz \notin (\succ - P^-). \quad (6)$$

(6) and (5) imply

$$uv \in \Phi(xy) \vee vz \in \Phi(xy) \quad (7)$$

By definition of $\Phi(xy)$, it is not possible that both uv and vz are in $\Phi(xy)$. W.l.o.g. assume that

$$uv \in \Phi(xy) \wedge vz \notin \Phi(xy) \quad (8)$$

From (8) and (5), it follows that

$$uv \in \Phi(xy) \wedge vz \notin P^- \quad (9)$$

However, (9) implies $uz \in \Phi(xy)$ by definition of $\Phi(xy)$. Thus $uz \in (\succ - (P^- - \Phi(xy)))$ which contradicts to (5).

Theorem 1. *Let P^- be a contraction of $\succ$ by CON. Then P^- is a* minimal *contraction of $\succ$ by CON iff for every $xy \in P^-$, there is a CON-detour T in $\succ$ which contains the edge xy and no other edge in T is in P^-.*

Or, in other words, for any edge in P^-, there exists at least one CON-detour which is disconnected only by that edge.

Proof

$\Rightarrow$ First, prove that if P^- is a minimal contraction, then for every $xy \in P^-$, there exists a CON-detour T disconnected only by xy, i.e.

$$\begin{aligned} & \forall xy \in P^-(\exists CON\text{-detour } T(\\ & \quad xy \in T \wedge \forall uv(uv \neq xy \wedge uv \in T \rightarrow uv \notin P^-))) \end{aligned}$$

Assume it is not the case. Then

$$\begin{aligned} & \exists xy \in P^-(\forall CON\text{-detour } T(\\ & \quad xy \notin T \vee \exists uv(uv \neq xy \wedge uv \in T \wedge uv \in P^-))) \end{aligned} \quad (1)$$

Consider the first disjunct of (1). That is, prove that for any $xy \in P^-$, there exists a CON-detour which xy belongs to. If for some $xy \in P^-$, no such detour exists, then $\Phi(xy) \cap CON = \emptyset$ by the construction of $\Phi(xy)$. Hence by Lemma 3, $(P^- - \Phi(xy))$ is a contraction of $\succ$ by CON. This contradicts to the assumption that P^- is a *minimal* contraction.

Consider the second disjunct of (1). Similarly to what we did above, let us show that $\Phi(xy) \cap CON = \emptyset$. If $\exists uv \in \Phi(xy) \cap CON$, then by Lemma 3, $u \succeq x \wedge y \succeq v \wedge ux, yv \notin P^-$, i.e. there's a CON-detour from u to v where only xy is in P^-. However, we assumed that such a detour does not exists. Thus $CON \cap \Phi(xy) = 0$. Then by Lemma 3, $(P^- - \Phi(xy))$ is a contraction of $\succ$ by CON. This contradicts to the assumption that P^- is a *minimal* contraction.

$\Leftarrow$ Let for every edge in P^-, there exists at least one CON-detour disconnected only by that edge. In this case, if we remove some edge xy from the contraction P^-, then there will be a CON-detour which is not disconnected and thus by Lemma 1, $(\succ -P^- \cup \{xy\})$ is not a contraction of $\succ$ by CON. Hence, P^- is a minimal contraction.

Appendix 3. Proof of Corollary 1

Corollary 1. *A contraction P^- of $\succ$ by CON is* minimal *iff the formula*

$$\begin{aligned}\forall x, y\, \exists u, v (F_{P^-}(x,y) \wedge F_{CON}(u,v) \wedge F_{\succ}(x,y) \wedge \\ \neg F_{P^-}(u,x) \wedge \neg F_{P^-}(y,v) \wedge \\ (F_{\succ}(u,x) \vee u = x) \wedge (F_{\succ}(y,v) \vee y = v)) \qquad (1)\end{aligned}$$

is valid.

Proof

Prove that the validity of (1) is equivalent to the necessary and sufficient condition from Theorem 1. Namely, prove that (1) is valid iff *for every edge $xy \in P^-$, there is a CON-detour disconnected only by xy.*

$\Leftarrow$ (1) implies that for all xy, there is a CON-detour consisting of one (if $u = x$ and $y = v$) up to three (if $u \neq x$ and $y \neq v$) edges going from u to v which is disconnected only by xy.

$\Rightarrow$ Assume that for some edge $xy \in P^-$, there is a CON-detour from u to v

$$u \succ \ldots \succ x \succ y \succ \ldots \succ v$$

disconnected only by xy. The detour from u to x is not disconnected, and thus, by transitivity of $(\succ -P^-)$, $ux \in (\succ -P^-)$ unless $u = x$. Similarly, either $yv \in (\succ -P^-)$ or $y = v$. Hence, there is a CON-detour of at most three edges disconnected only by xy.

Appendix 4. Proof of Lemma 2

Lemma 2. *Let $\succ$ be a preference relation and CON be a contracting relation of $\succ$. Then*

$$P^- := \{\, xy \mid \exists xv \in CON\ .\ x \succ y \wedge (y \succ v \vee y = v)\}$$

is a contraction of $\succ$ by CON.

Proof

To prove that P^- is a contraction of $\succ$ by CON, it suffices to show that $CON \subseteq P^-$ and $(\succ -P^-)$ is transitive. First, $(\succ -P^-)$ is transitive by Lemma 1, since for every edge $xy \in P^-$, the starting of each detour from x to y is in P^-. Second, P^- contains CON by construction.

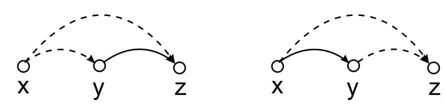
Appendix 5. Proof of Theorem 2

Before we go into the details of the proof, let us define the relations *above* and *below* over edges.

Definition 9 *Given a preference relation $\succ$ and two edges $xy, x'y'$ of $\succ$, the edge xy is* above *(or* below*) the edge $x'y'$ if $y \succ y'$ (or $y' \succ y$, correspondingly).*

The next notion, *forks*, is used to simplify the description of the theorem proof.

Definition 10 *Let $\succ$ be a preference relation and P^- be a subset of $\succ$. Then a triple xyz is a* fork *in $\succ -P^-$ if 1) $x \succ y \succ z$, and 2) $xz, xy \in P^- \wedge yz \notin P^-$ (or, $xz, yz \in P^- \wedge xy \notin P^-$).*



(a) Fork xyz, the shorter edge xy, the longer edge xz.

(b) Fork xyz, the shorter edge yz, the longer edge xz.

Figure 5: Forks

The edge xy (or, yz respectively) is called the shorter edge of the fork *and xz is called* the longer edge of the fork.

To keep the notation simple, let us denote the relation returned by Algorithm 1 as P^-.

Lemma 4 *Algorithm 1 returns a contraction of $\succ$ by CON and terminates in k iterations for a* k-layer *relation CON.*

Proof

1) *Termination*. Prove that the algorithm stops in k iterations. Note that the initial value of C (i.e. C_0) is equal to CON. In each iteration, E_i is constructed as a superset of the set of the bottom most edges in C_{i-1}. After that, the set C_{i-1} is reduced by E_i. Since C_0 has k *layers*, the function will terminate in k iterations when all the layers of CON are exhausted.

2) *Contraction*. Prove that P^- is a contraction of $\succ$ by CON, i.e. a) $CON \subseteq P^-_k$ and b) $(\succ -P^-_k)$ is an SPO.

a) E_i calculated at every iteration of the algorithm is a superset of the CON-edges of layer i. Hence P^-_i is a superset of the CON-edges of the layers 1 through i. Therefore, P^- is a superset of CON.

b) Prove that $(\succ -P^-_k)$ is an SPO. Since $\succ$ is irreflexive, $(\succ -P^-_k)$ is irreflexive, too. So it suffices to show that $(\succ -P^-_k)$ is transitive. Prove that the relation $(\succ -P^-_i)$ is transitive for every i from 0 to k. We do it by induction on i.

1) Base step. $(\succ -P^-_0) = \succ$ is transitive since $\succ$ is an SPO relation.

2) Inductive step. Let $(\succ - P_i^-)$ be transitive. Prove that $(\succ - P_{i+1}^-)$ is transitive, too. For the sake of contradiction, assume that $(\succ - P_{i+1}^-)$ is not transitive. So there exist x, y, z such that

$$xy \in P_{i+1}^- = P_i^- \cup E_{i+1} \quad (1)$$

but

$$xz, zy \notin P_{i+1}^- = P_i^- \cup E_{i+1} \quad (2)$$

(1) implies that either $xy \in P_i^-$ or $xy \in E_{i+1}$. However, $xy \in P_i^-$ along with (1) implies that $(\succ -P_i^-)$ is not transitive which constradicts to the assumption. Thus

$$xy \in E_{i+1} \quad (3)$$

Therefore, by the definition of E_{i+1}, the following is true

$$\exists v \in L_{i+1}(xv \in CON \wedge x \succ y \succeq v \wedge yv \notin P_i^- \wedge yv \notin CON)\} \quad (4)$$

So we have the CON-edge xv with $v \in L_{i+1}$ and $x \succ z \succ v$ by transitivity of $\succ$. From (2), we know that $xz \notin E_{i+1}$. It implies (by the definition of E_{i+1}) that either 1) $zv \in P_i^-$ or 2) $zv \in CON$.

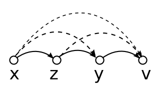


Figure 6: Transitivity. Inductive case. Dashed edges are in $(E_{i+1} \cup P_i^-)$. Solid edges are in $(\succ -(E_{i+1} \cup P_i^-))$.

In case 1) we have $zv \in P_i^-$, $zy \notin P_i^-$, $yv \notin P_i^-$. We also know from (4) that $y = v$ or $y \succ v$. However, y can not be equal to v since then $zy \in P_i^-$ and $zy \notin P_i^-$. At the same time, $y \succ v$ implies that $(\succ -P_i^-)$ is not transitive. Contradiction.

Consider case 2), i.e. $zv \in CON$. Since $v \in L_{i+1}$, by the definition of E_{i+1}, we have three choices: (i) $zy \in E_{i+1}$, (ii) $yv \in CON$, and (iii) $yv \in P_i^-$ by the construction of E_{i+1}. The choice (i) contradicts to (2). The choices (ii) and (iii) contradict to (4). Contradiction.

Lemma 5 *Take the i-th iteration of Algorithm 1 for any i. Then for any non-CON edge xy of E_i, there exist $v \in L_i$ such that xyv is a fork in $(\succ -P^-)$ with the shorter edge xy.*

Proof
Take any $xy \in E_i - CON$. Then by the definition of E_i,

$$\exists v \in L_i \wedge xv \in CON \wedge x \succ y \succeq v \wedge yv \notin P_{i-1}^- \wedge yv \notin CON \quad (1)$$

By construction of P^-, $E_i \subseteq P^-$. Thus

$$xy \in P^-. \quad (2)$$

By Lemma 4, $CON \subseteq P^-$. Thus

$$xv \in P^-. \quad (3)$$

Moreover, $v = y$ would mean that xy is a CON-edge which contradicts to the assumption. From that and (1), we get

$$y \succ v. \quad (4)$$

Assume xyv is not a fork in $(\succ -P^-)$. Then (2), (3), and (4) imply that yv must be in P^- (otherwise xyv is a fork). Thus there exists j such that $i \leq j$ and $yv \in E_j$. The next expression shows what it means for yv to be in E_j

$$\exists u \in L_j(yu \in CON \wedge y \succ v \succeq u \wedge vu \notin P_{j-1}^- \wedge vu \notin CON) \quad (5)$$

As a result, $yv \notin CON$ (see (1)) and $yu \in CON$ (see (5)) imply that $v \neq u$ and thus $v \succ u$. However, since $u \in L_j$, $v \in L_i$, and $j \geq i$, $v \succ u$ is not possible by the construction of L_i, L_j.

Theorem 2. *Algorithm 1 returns a minimal contraction of $\succ$ by CON and halts in k iterations for a* k-layer *relation CON.*

Proof
1) *Termination*. See Lemma 4.
2) *Contraction*. See Lemma 4.
3) *Minimality.* Prove that P_k^- is a *minimal* contraction of $\succ$ by CON. By Lemma 5, every non-CON edge xy in P^- is the shorter edge in a fork xyv in $(\succ -P^-)$ where xv is a CON-edge. Thus, the two-edge CON-detour consisting of the edges xy and yv is disconnected only by xy. Hence, by Theorem 1, P^- is a minimal contraction.

Appendix 6. Proof of Theorem 3

Theorem 3. *Let CON be a k-layer contracting relation, and $P^+ \subset \succ$. There exists a minimal contraction of $\succ$ by CON that protects P^+ iff $P_{TC}^+ \cap CON = \emptyset$, where P_{TC}^+ is the transitive closure of P^+.*

Proof
$\Rightarrow$ Prove that if P^- is a minimal contraction of $\succ$ by CON protecting P^+, then $P_{TC}^+ \cap CON = \emptyset$. If $\exists xy \in P_{TC}^+ \cap CON$, then there is a CON-detour from x to y which is entirely in P^+, and no edge from this detour is in P^-. However, the edge xy must be in P^-, since P^- is a contraction by CON. Thus by Lemma 1, $(\succ -P^-)$ is not transitive, and P^- is not a contraction of $\succ$ by CON.
$\Leftarrow$ If $P_{TC}^+ \cap CON = \emptyset$, then Algorithm 2 can be used to compute a minimal contraction of $\succ$ by CON protecting P^+.

Appendix 7. Proof of Proposition 1

Proposition 1. *Take any $P^+ \subset \succ$. Then any contraction of $\succ$ by CON protecting P^+ contains the set Q*

$$Q = \{xy \mid \exists u : u \succ x \succ y \wedge uy \in CON \wedge ux \in P^+\}$$

Proof
Take any contraction P^- of $\succ$ by CON protecting P^+. Let $xy \in Q$, i.e.

$$\exists u : u \succ x \succ y \wedge uy \in CON \wedge ux \in P^+.$$

Then $uy \in CON$ implies $uy \in P^-$. Since P^- protects P^+, $ux \notin P^-$. Thus if $xy \notin P^-$, then $(\succ -P^-)$ is not transitive and therefore not a contraction of $\succ$. Hence, xy must be a member of P^-.

Appendix 8. Proof of Theorem 4

In Theorem 3 we showed that

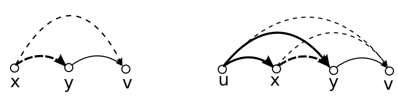
$$P_{TC}^{+} \cap CON = \emptyset \tag{A}$$

is required to be able to construct a minimal contraction of $\succ$ by CON with protected P^{+}. So in the following theorem, we assume that the above condition holds.

Theorem 4. *If CON is a* k-layer *contracting relation, and P^{+} is transitive, then Algorithm 2 terminates and returns a contraction of $\succ$ by CON which 1) is minimal, and 2) protects P^{+}.*

Proof

1. *Termination.* Prove that the function `minContrProt` terminates. Note that by the construction of CON', if we take any edge $xy \in CON'$, there will be an edge $x'y \in CON$. Thus, if CON is a k-layer relation, CON' is k-layer, too, and by Theorem 2, `minContr(`$\succ$`,` CON'`)` terminates. Hence, the function `minContrProt` terminates, too.
2. *Contraction & P^{+} protection.* By Theorem 2, `minContr(`$\succ$`,` CON'`)` returns a contraction P^{-} of $\succ$ by CON'. Clearly, since $CON \subseteq CON'$, P^{-} is also a contraction of $\succ$ by CON.
Now prove that P^{-} protects P^{+}, i.e. that $P^{+} \cap P^{-} = \emptyset$. Assume
$$\exists xy \in P^{-} \cap P^{+}. \tag{1}$$
Since P^{-} is returned by `minContr`, there exists i such that $xy \in E_i$, i.e.
$$\exists v \in L_i (xv \in CON' \wedge x \succ y \wedge (y \succ v \vee y = v) \wedge yv \notin P_{i-1}^{-} \wedge yv \notin CON')\} \tag{2}$$



(a) Case 1. (b) Case 2.

Figure 7: Proof of P^{+} preservation.

(2) implies these two cases: 1) $xv \in CON$, and 2) $xv \in Q$.
Case 1.
$$xv \in CON. \tag{3}$$
Then $y = v$ along with (3) and (1) violate (A). Therefore, (2) implies
$$y \succ v. \tag{4}$$
Next, $xv \in CON$ and $xy \in P^{+}$ imply $yv \in Q$ by the definition of Q. Thus $yv \in CON' = CON \cup Q$. However, this contradicts to (1).
Case 2.
$$xv \in Q. \tag{5}$$
Then, according to the expression for Q, we get
$$\exists u : u \succ x \succ v \wedge uv \in CON \wedge ux \in P^{+} \tag{6}$$
From $ux \in P^{+}$ and $xy \in P^{+}$ we get that $uy \in P^{+}$ by transitivity of P^{+}.
From $uv \in CON$ and $uy \in P^{+}$ it follows that $yv \in Q$ by definition of Q. So we get the same contradiction as in case 1.
3. *Minimality.* Use Theorem 1 to prove that P^{-} is a minimal contraction of $\succ$ by CON. We need to show that for every $xy \in P^{-}$ there's a CON-detour which is disconnected only by xy.
By Theorem 2, P^{-} is a minimal contraction of $\succ$ by CON'. Thus, by Theorem 1, there is a CON'-edge uv such that a path T from u to v is disconnected only by xy. We have two choices: 1) $uv \in CON$, and 2) $uv \in Q$.
In the first case, the same path T will satisfy the minimality condition of Theorem 1.
In the second case, $uv \in Q$ implies
$$\exists z : z \succ u \succ v \wedge zv \in CON \wedge zu \in P^{+}$$
Take the path T' which consists of the edge zu and the path T appended to it. This path T' is a CON-detour going from z to v and is disconnected by only xy. Hence, T' satisfies the minimality condition of Theorem 1.

Appendix 9. Proof of Proposition 2

Proposition 2

1. $w_{\succ}(r) \subseteq w_{\succ'}(r)$

2. If $\sigma_{F_{S(P^{-})}}(w_{\succ}(r)) = \emptyset$*, then* $w_{\succ}(r) = w_{\succ'}(r)$*.*

3. $w_{\succ'}(r) = w_{\succ'}(w_{\succ}(r) \cup \sigma_{F_{E(P^{-})}}(r))$

4. If P^{-} is a minimal contraction, then $w_{\succ'}(r) = w_{\succ'}(w_{\succ}(r) \cup \sigma_{F_{M(CON)}}(r))$

5. If P^{-} is a prefix contraction, then $\sigma_{F_{S(P^{-})}}(r) = \sigma_{F_{S(CON)}}(r)$

Proof

1. By definition, $w_{\succ}(r)$ contains the set of the undominated objects w.r.t the preference relation $\succ$. Thus, $\succ' \subset \succ$ implies that if an object o was undominated w.r.t $\succ$, it will be undominated w.r.t $\succ'$, too. However, if o was dominated w.r.t. $\succ$, it will become undominated w.r.t. $\succ'$ if all the $\succ$-edges going to o were contracted. Thus $w_{\succ}(r) \subseteq w_{\succ'}(r)$.
2. Assume that there is an object o such that
$$o \in w_{\succ'}(r) - w_{\succ}(r). \tag{1}$$
Since $o \notin w_{\succ}(r)$, there is an object o' such that
$$o' \in w_{\succ}(r) \wedge o \succ o. \tag{2}$$
However, since $o \in w_{\succ'}(r)$, the object o' does not dominate o w.r.t $\succ'$. Thus,
$$o'o \in P^{-} \tag{3}$$
and $o' \in S(P^{-})$. Since, $o' \in w_{\succ}(r)$, we get $o' \in \sigma_{F_{S(P^{-})}}(w_{\succ}(r))$, i.e. $\sigma_{F_{S(P^{-})}}(w_{\succ}(r)) \neq \emptyset$.
3. It is clear that for any subset r' of r which contains $w_{\succ'}(r)$, we have $w_{\succ'}(r) = w_{\succ'}(r')$. That is,
$$\forall r' (w_{\succ'}(r) \subseteq r' \subseteq r \rightarrow w_{\succ'}(r) = w_{\succ}(r')) \tag{4}$$
Therefore, to prove (5)
$$w_{\succ'}(r) = w_{\succ'}(w_{\succ}(r) \cup \sigma_{F_{E(P^{-})}}(r)) \tag{5}$$

we need to show that

$$w_{\succ'}(r) \subseteq w_{\succ}(r) \cup \sigma_{F_{E(P^-)}}(r)$$

or

$$w_{\succ'}(r) - w_{\succ}(r) \subseteq \sigma_{F_{E(P^-)}}(r). \tag{6}$$

Take the objects o and o' as shown in (1) and (2). Then (3) implies $o \in E(P^-)$. Moreover, (1) implies $o \in r$. Thus, $o \in \sigma_{F_{E(P^-)}}(r)$.

4. From Theorem 1, it follows that the statement (7) is true.

$$\sigma_{F_{E(P^-)}}(r) \subseteq \sigma_{F_{M(CON)}}(r). \tag{7}$$

Moreover, (7) implies (8)

$$w_{\succ}(r) \cup \sigma_{F_{E(P^-)}}(r) \subseteq w_{\succ}(r) \cup \sigma_{F_{M(P^-)}}(r). \tag{8}$$

(8), (5), and (4) imply

$$w_{\succ'}(r) = w_{\succ'}(w_{\succ}(r) \cup \sigma_{F_{M(CON)}}(r))$$

5. Follows from the definition of *prefix* contractions.

Appendix 10. Finite case implementation of Algorithm 1

Here we present an implementation of Algorithm 1 for the finite case of $\succ$`(X,Y)` and `CON(X,Y)`. It consists of three functions: `init`, `getLayerOrder`, and `minContrFinite`. It constructs a table `R(X,Y,F)` which is a copy of $\succ$ `(X,Y)` with the flag `F` set to `1` if the corresponding tuple is a member of the minimal contraction returned by the algorithm.

The function `init` initializes the algorithm by creating the table `R`, setting the flag `F` to `1` for all the `CON`-edges of `R`, and sorting `R` and `CON`.

Algorithm 3 init($\succ$, `CON`)

Create a table `R(X,Y,F)` and copy $\succ$`(X,Y)` to it. For each row of `R`, set `F` to `0`.
Sort `R` by the pair `(X,Y)`.
Sort `CON` by the pair `(X,Y)`.
for all `t` in `CON` **do**
  `t'` := find a tuple in `R` with `X` = `t.X` and `Y` = `t.Y`
  if `t'` exists **then**
    `t'.F` := 1
  end if
end for
return `R`, `CON`

The function `getLayerOrder` orders `CON`-edges by layer index. Namely, it creates a list `L` of the destination-nodes of `CON` ordered by the layer index of the corresponding `CON`-edge. It is done by copying $\succ$(X,Y) to in the table `T(X,Y,C)` and setting the flag `C` of each `T`-tuple to `1` if the corresponding tuple represents an edge going from one CON-edge destination to another. After that, we pick all the CON-edge destinations in the order of their layer index, and store them in the list `L`.

Algorithm 4 `getLayerOrder(R, CON)`

$\mathtt{Y}_{CON}$:= the list of all `Y`-values of `CON`
Sort `E` and eliminate duplicates
Create a table `T(X,Y,C)` and copy `R(X,Y)` to it. Set the value of `C` of each row to `0`.
for all `t` in `T` **do**
  if `t.X` in `E` and `t.Y` in $\mathtt{Y}_{CON}$ **then**
    `t.C` := 1
  end if
end for
`L` := empty list
repeat
  for all `b` in `L` **do**
    N_b := # of tuples in `T` with `X` = `b`
    if N_b = 0 **then**
      Push `b` to the end of `L`
      Delete all the tuples from `T` with `Y` = `b`
      Delete `b` from $\mathtt{Y}_{CON}$
    end if
  end for
until $|$`E`$| = \emptyset$
return `L`

Algorithm 5 `minContrFinite(`$\succ$`, CON)`

Require: $\succ$ is transitive, `CON` $\in \succ$
1: `R`, `CON` := `init(`$\succ$`, CON)`
2: `L` := `getLayerOrder(R, CON)`
3: **for all** `e` in L **do**
4:   **for all** `c` in `CON` **do**
5:     **if** `c.Y` = `e` **then**
6:       **for all** `t` in `R` **do**
7:         **if** `t.X` = `c.X` and `t.F` = 0 **then**
8:           **if** exists a tuple `o` in `R` with `o.X` = `t.Y`, `o.Y` = `e`, and `o.F` = 0 **then**
9:             `t.F` := 1
10:           **end if**
11:         **end if**
12:       **end for**
13:     **end if**
14:   **end for**
15: **end for**
16: **return** all tuples `t` in `R` with `t.F = 1`

The function `minContrFinite` is the main function of the algorithm. First, it performs some preparations by calling `init` and `orderNodesByLayerIndex`. Then it picks every element `e` of `L`, and for every `CON`-edge which ends in `e`, it checks if there is a two-edge `CON`-detour which is not disconnected yet. If it exists, the starting edge of the detour is added to the contraction (i.e. the flag `F` of the corresponding `R`-edge is set to `1`).

The algorithm runtime analysis gives the following results. 1) the function `init` requires time $O(|\succ| \cdot log|\succ| + |CON| \cdot (log|\succ| + log|CON|))$. 2) the function `getLayerOrder` requires time $O(|CON|^2 \cdot |\succ|)$. Finally, the loop in lines 3-15 of `minContrFinite` requires time $O(|CON|^2 \cdot |\succ| \cdot log|\succ|)$. Thus, the total run time is $O(|CON|^2 \cdot |\succ| \cdot log|\succ|)$.

References

Balke, W.-T.; Guntzer, U.; and Siberski, W. 2006. Exploiting indifference for customization of partial order skylines. In *Proc. IDEAS '06*, 80–88.

Boutilier, C.; Brafman, R.; Domshlak, C.; Hoos, H.; and Poole, D. 2004. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Research (JAIR)* 21:135–191.

Brafman, R. I.; Domshlak, C.; and Shimony, E. 2002. Introducing variable importance tradeoffs into CP-nets. *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)* 69–76.

Chomicki, J. 2003. Preference formulas in relational queries. *ACM Trans. Database Syst.* 28(4):427–466.

Chomicki, J. 2007a. Database Querying under Changing Preferences. *Annals of Mathematics and Artificial Intelligence* 50(1-2):79–109.

Chomicki, J. 2007b. Semantic optimization techniques for preference queries. *Inf.Syst.(IS)* 32(5):670–684.

Dong, G.; Libkin, L.; Su, J.; and L.Wong. 1999. Maintaining the transitive closure of graphs in SQL. *Int. Journal of Information Technology* 5:46–78.

Doyle, J. 2004. Prospects for preferences. *Computational Intelligence* 20(2):111–136.

Endres, M., and Kießling, W. 2006. Transformation of TCP-net queries into preference database queries. *Proceedings of the ECAI 2006 Multidisciplinary Workshop on Advances in Preference Handling*.

Fishburn, P. C. 1970. Utility theory for decision making. *Wiley and Sons*.

Hafenrichter, B., and Kießling, W. 2005. Optimization of relational preference queries. In *ADC '05: Proceedings of the 16th Australasian database conference*, 175–184.

Hansson, S. O. 1995. Changes in preference. *Theory and Decision* 38(1):1–28.

Kießling, W. 2002. Foundations of Preferences in Database Systems. *International Conference on Very Large Data Bases (VLDB)* 311–322.

Mindolin, D., and Chomicki, J. 2007. Hierarchical CP-networks. *3rd Multidisciplinary Workshop on Advances in Preference Handling (M-PREF)* Vienna, Austria.

Wilson, N. 2004. Extending CP-nets with stronger conditional preference statements. *Proc. AAAI 2004* 735–741.