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. Here, 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.

Introduction

A large number of preference handling frameworks have been developed (Fishburn 1970; Boutilier *et al.* 2004; Hansson & Grüne-Yanoff 2006). In this paper, we work with the *binary relation* preference framework (Chomicki 2003). Preferences are represented here as binary relations over objects. They are required to be *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*.

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. Preference change is an active topic of current research (Chomicki 2007a; Freund 2004). 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, prioritized, 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 she prefers newer cars. Such a preference can be represented as the relation $\succ_1$ defined by the following formula*

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

where $>$ denotes the standard ordering of rational numbers, and the attribute y defines the year when cars are made.

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 (this is prioritized composition). 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 small. 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 change to the preference relation $\succ_2$. 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 15000.

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 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 *contracting preference relations minimally while preserving the SPO properties*. The main results of the paper are as follows. First, we present necessary and sufficient conditions for the minimal and the minimal preference-protecting contractions. Second, we provide algorithms to compute such contractions. Finally, we show how to optimize preference query evaluation in the presence of contraction.

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* $\mathcal{A} = \{A_1, \ldots, A_m\}$. 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 R, its formula representation is denoted F_R.

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.

Binary relations are commonly represented as *directed graphs*.

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 define a finite* R-path from x to y *and an infinite* R-path from x. *The length of a finite R-path is the number of R-edges in the path.*

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 $\succ$ with subscripts to refer to preference relations. We write $x \succeq y$ as a shorthand for $(x \succ y \vee x = y)$.

Preference contraction

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

Definition 3 *A binary relation P^- is* a full contractor 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 full contractor of $\succ$ by CON *if P^* is a full contractor of $\succ$ by CON, and there is no other full contractor P' of $\succ$ by CON s.t. $P' \subset P^*$.*

A preference relation is minimally contracted *if it is contracted by a* minimal full contractor.

The notion of minimal full contractor narrows the set of full contractors. However, as we illustrate in Example 2, minimal full contractor is generally not unique for given preference and base contractor relations. In fact, the number of minimal full contractors 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.

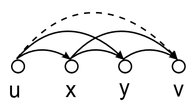


Figure 1: Preference $\succ$.

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

Contraction conditions

Definition 4 *Given a base contractor 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 full contractor, 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 full contractor.

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 full contractors. For instance, take the minimal full contractor from Example 2. Note that adding any edge from a minimal full contractor 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 full contractors is formally stated in Theorem 1.

Theorem 1 *Let P^- be a full contractor of $\succ$ by CON. Then P^- is a* minimal *full contractor 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 three due to the transitivity of $\succ$.

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

$$\forall x, y(F_{P^-}(x, y) \rightarrow F_{\succ}(x, y) \wedge \exists u, v(F_{CON}(u, v) \wedge (F_{\succ}(u, x) \vee u = x) \wedge (F_{\succ}(y, v) \vee y = v) \wedge \neg F_{P^-}(u, x) \wedge \neg F_{P^-}(y, v)))$$

holds.

Thus, in the infinite case of $\succ$, checking minimality of a full contractor can be done by performing quantifier elimination on the above formula. To check minimality in the finite case, one has to construct a relational algebra expression which corresponds to the negation of the formula above; evaluating it over the finite relations $\succ$, CON, and P^-; and checking if the corresponding result set is empty.

Construction of minimal full contractor

In the algorithm computing a minimal full contractor 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 full contractor.

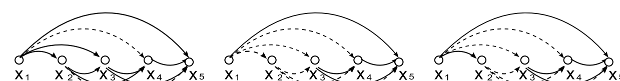
Generally, if CON contains more than one edge, the set consisting of all edges starting CON-detours is a full contractor by CON.

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

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

is a full contractor of $\succ$ by CON.

However, in the next example we show that such a full contractor is not always minimal.



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

Figure 2: Preference contraction

Example 3 *Take the preference relation $\succ$ as shown in Figure 2(a) as the set of all edges, and the base contractor 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 full contractor of $\succ$ by CON. In fact, $P^- - \{x_1x_2\}$ is a minimal *full contractor of $\succ$ by CON.*

As we can see, having the edge x_1x_2 in P^- is 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 full contractor 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 full contractor by a base contractor CON under the condition that CON is a *k-layer* relation defined as follows.

Definition 5 *The* 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 partition CON into layers and then process the layers one by one.

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

Let also the base contractor 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* 1.

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 full contractor 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.

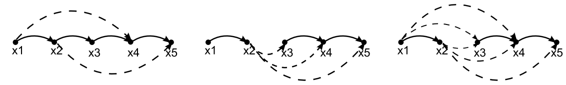
Theorem 2 *For a* k-layer *base contractor relation* CON*, Algorithm 1 returns a minimal full contractor of* $\succ$ *by* CON *and halts in* $k + 1$ *iterations.*

Algorithm 1 minContr($\succ$, CON)

$i = -1$, $P^-_{-1} = \emptyset$, $C_{-1} = CON$
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}
  $E_i := \{xy \mid \exists v \in L_i(xv \in CON \wedge x \succ y \wedge y \succeq v \wedge yv \notin P^-_{i-1} \wedge yv \notin CON)\}$
  $P^-_i := P^-_{i-1} \cup E_i$ {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 omitted for clarity. Let a base contractor* CON *be defined by the dashed edges.*

Then the value of C_{-1} *is set to* $\{x_1x_4, x_2x_5\}$*. The result of applying the first step of the algorithm is shown in Figure 3(b). Namely,* $L_0 = \{x_5\}$*,* $P^-_0 = \{x_2x_3, x_2x_4, x_2x_5\}$*, and* $C_0 = \{x_1x_4\}$*. At the second iteration (Figure 3(c)),* $L_1 = \{x_4\}$*,* $P^-_1 = P^-_0 \cup \{x_1x_3, x_1x_4\}$*, and* $C_1 = \emptyset$*. After that, the algoritm returns* P^-_1 *since* $C_1 = \emptyset$*.*



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

Figure 3: Preference contraction

We note that the *k-layer* base contractor relations are not necessary finite. For instance, the relations CON in Example 1 and CON_1 in Example 4 are infinite. 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 full contractor 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 (Mindolin & Chomicki 2008) 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 full contractor P^- generated by Algorithm 1 has the property that every edge in P^- starts a CON-detour in $\succ$. We call such full contractors *prefix full contractors*.

Preference-protecting contraction

Generally, it is not always the case that all minimal full contractors are equivalent from the point of view of users. For instance, a full contractor may discard some preferences (in addition to CON) which the user does not want to discard. Thus, in addition to specifying a base contractor 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 base contractor: 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 full contractor of $\succ$ by CON that protects P^+ *if 1)* P^* *is a minimal full contractor of* $\succ$ *by* CON*, and 2)* $P^* \cap P^+ = \emptyset$*.*

Contraction conditions

Given any full contractor 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^+ while contracting $\succ$ by CON is not possible.

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

Minimal preference-protecting full contractor

A naive way of computing a minimal preference-protecting full contractor is to find a minimal full contractor 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 full contractor algorithm shown in the previous section. First, we find a base contractor 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* full contractor 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 full contractor which protects P^+. If P^+ contains edges starting CON-detours, then any P^+-protecting full contractor has to contain the set Q defined in the next proposition.

Proposition 1 *Given $P^+ \subset \succ$, every full contractor of $\succ$ by CON protecting P^+ contains the set Q*

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

The reasoning beyond Proposition 1 is that if we want to discard a preference uy but protect an edge ux starting its two-edge detour, then we have no choice but discarding the edge xy ending that detour. Otherwise, the transitivity of the contracted preference relation is violated.

We show further that if P^+ is transitive and P^- is a minimal prefix full contractor 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
$Q = \{xy \mid \exists u : u \succ x \succ y \wedge uy \in CON \wedge ux \in P^+\}$
$CON' = CON \cup Q$
$P^- = minContr(\succ, CON')$
return P^-

Theorem 4 *If CON is a* k-layer *base contractor, P^+ is transitive, and $P^+ \cap CON = \emptyset$, then Algorithm 2 terminates and returns a full contractor of $\succ$ by CON which 1) is minimal w.r.t. CON, and 2) protects P^+.*

Note that we use the function `minContr` in Algorithm 2 because CON' is a k-layer relation. It is justified 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 full contractors of finite and finitely representable preference relations.

Preference query evaluation

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 latter problem, the *winnow operator* is proposed in (Chomicki 2003). The winnow 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 base contractor of $\succ$, P^- be a full contractor of $\succ$ by CON, and $\succ' = (\succ - P^-)$. Denote the set of the starting and the ending objects of R-edges for a binary relation R as $S(R)$ and $E(R)$ correspondingly.

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

Let us also define the set $M(CON)$ of the objects which participate in CON-detours in $\succ$

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

Assume we also know quantifier-free formulas $F_{S(P^-)}$, $F_{E(P^-)}$, $F_{M(CON)}$, and $F_{S(CON)}$ representing these sets for P^- and CON. 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 full contractor, then* $w_{\succ'}(r) = w_{\succ'}(w_\succ(r) \cup \sigma_{F_{M(CON)}}(r))$
5. *If P^- is a prefix full contractor, 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 $F_{S(P^-)}$.

Related and future work

A general framework of preference change is proposed in (Hansson 1995). Preference change is considered there 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 base contractor. 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.

Acknowledgments

The comments of the anonymous reviewers are greatly acknowledged. Research supported by NSF grant IIS-0307434.

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.

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.

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

Freund, M. 2004. On the revision of preferences and rational inference processes. *Artif. Intell.* 152(1):105–137.

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., and Grüne-Yanoff, T. 2006. Preferences. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. http://plato.stanford.edu/archives/win2006/entries/preferences/.

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

Mindolin, D., and Chomicki, J. 2008. Minimal contraction of preference relations. Technical report 2008-05, CSE Dept., University at Buffalo, SUNY.