A polynomial time algorithm to detect PQI interval orders

An Ngo The(#,*), Alexis Tsoukiàs(*) and Philippe Vincke(+)
(#) Programme Doctorale ESSEC
(*) LAMSADE - CNRS, Université Paris Dauphine
{ngothe,tsoukias}@lamsade.dauphine.fr
(+) SMG - ISRO, Université Libre de Bruxelles
pvincke@smg.ulb.ac.be

Abstract

Let S be a PQI preference structure on a finite set A, where the relations P, Q, I stand for "strict preference", "weak preference" and "indifference" respectively. Two specific preference structures: PQI semi orders and PQI interval orders, have been considered and characterised recently in such a way that is possible to associate to each element of A an interval such that P holds when one interval is completely to the right of the other, I holds when one interval is included to the other and Q holds when one interval is to the right of the other, but they do have a non empty intersection (Q medelling the hesitation). While the detection of a PQI semiorder is straightforward, the case of the PQI interval order is more difficult as the theorem of existence consists in a second-order formula. The paper presents an algorithm for detecting a PQI interval order and demonstrates that it is backtracking free. This result leads to a matrix version of the algorithm which can be proved to be polynomial.

Keywords: Interval Orders, PQI Interval Orders, Detection Algorithm, Complexity

Introduction

In preference modelling and decision support we often have to compare intervals instead of discrete values (see Fishburn, 1985, Pirlot and Vincke, 1997). This is due to the unavoidable lack of precision and certainty in the evaluation of alternatives. The conventional model adopted in order to compare two intervals considers that "x is preferred to y" ($P(x, y)$) iff the interval associated to x is completely to the "right" (in the sense of the line representing the reals) of the interval associated to y. In all other cases "x is indifferent to y". Such a model (where indifference is not transitive) may conceal the fact that "x being to the right of y" (the intersection being not empty) is a situation intuitively different from the case where one interval (let's say the one of x) is included in the other (let's say y). The second case can be considered a "sure indifference" as much as can be considered as "sure preference" the case $P(x, y)$. Under such a perspective the first case is a situation of hesitation between preference and indifference which merits to be considered separately (see Tsoukiàs and Vincke, 1997). We may denote such a situation as "weak preference" and represented it as $Q(x, y)$.

The problem is to give the necessary and sufficient conditions for which a preference structure characterised by the presence of the relations P, Q and I may admit a representation by intervals as the one previously discussed. Such a problem was considered open for a long time (see Vincke, 1988) and has been solved by Tsoukiàs and Vincke, 1999, where an existential theorem is given. The operational problem is how to detect if a given PQI preference structure satisfies the conditions of the theorem. The problem is not an easy one because the theorem consists in a second order formula which could be undecidable. Actually, while trying to verify the conditions of the theorem there is space for some arbitrary decisions resulting in a tree defined by the branches created by each such arbitrary choice. Intuitively, if after such a choice an inconsistency occurs a backtracking should be done in order to try a new branch. This may result in a problem at least in NP.

The paper is dedicated to present an algorithm for detecting the satisfaction of the theorem by a given PQI preference structure which is polynomial. The paper is organised as follows. Section 1 presents the basic definitions and the problem to solve. Section 2 gives two theorems, the first giving the necessary and sufficient conditions for a PQI preference structure to have an interval representation and the second giving a different characterisation, which is less intuitive, but which will be used in order to build the detection algorithm. Section 3 gives the algorithm in a procedural way by which it is possible to demonstrate that it is "backtracking free". Section 4 presents a "matrix implementation" of the algorithm enabling to demonstrate that it is polynomial. Some conclusions are given at the end of the paper. Appendix A contains the demonstrations of the propositions used in the proof of Theorem 3.1 as well as the algorithm 3.1.

1 Problem setting

In the following we will use the following notation for any binary relation $(S, T, \cdots)$ on a finite set A:
$S^{-1}(x, y) =_{def} S(y, x)$;
$(S \cup T)(x, y) =_{def} S(x, y) \vee T(x, y)$
$(S \cap T)(x, y) =_{def} S(x, y) \wedge T(x, y)$
$(S \subset T)(x, y) =_{def} S(x, y) \rightarrow T(x, y)$
$(S.T)(x, y) =_{def} \exists z \; S(x, z) \wedge T(z, y)$

We first give the conventional definition and theorem concerning interval orders.

Definition 1.1 *(see Roubens and Vincke, 1985)*
Given P an asymmetric binary relation and I a reflexive and symmetric relation, $P \cup I$ being complete, the preference structure $\langle P, I \rangle$ is an interval order iff there exist two real valued functions l and r, such that $\forall\, x, y \in A$:
- i. $r(x) > l(x)$;
- ii. $P(x, y) \Leftrightarrow l(x) > r(y)$;
- iii. $I(x, y) \Leftrightarrow r(y) > l(x)$ and $r(x) > l(y)$;

In conventional interval orders when comparing two intervals two situations are considered:
- one interval is completely to the right of the other (strict preference);
- there is a non empty intersection of the intervals (indifference).

Theorem 1.1 *A $\langle P, I \rangle$ preference structure on a finite set A is an interval order iff $P.I.P \subset P$.*

Proof See Roubens and Vincke, 1985.

We now give the definitions of PQI preference structure and PQI interval order.

Definition 1.2 *(see Roubens and Vincke, 1985)*
A PQI preference structure on a finite set A is a triple of binary relations

$\langle P, Q, I \rangle$, such that:
- i. I is reflexive and symmetric;
- ii. P and Q are asymmetric;
- iii. $I \cup P \cup Q$ is complete;
- iv. P, Q, I are mutually exclusive.

Intuitively, in a PQI preference structure, P represents "sure preference", I represents "sure indifference" and Q (weak preference) represents "hesitation between preference and indifference" (see Tsoukiàs and Vincke, 1997 for related issues).

Definition 1.3 *(see Tsoukiàs and Vincke, 1999)*
A PQI preference structure on a finite set A is a PQI interval order iff there exist two real valued functions l and r, such that $\forall\, x, y \in A$:
- i. $r(x) > l(x)$;
- ii. $P(x, y) \Leftrightarrow r(x) > l(x) > r(y) > l(y)$;
- iii. $Q(x, y) \Leftrightarrow r(x) > r(y) > l(x) > l(y)$;
- iv. $I(x, y) \Leftrightarrow r(x) > r(y) > l(y) > l(x)$ or $r(y) > r(x) > l(x) > l(y)$.

A PQI interval order extends conventional interval orders in the sense that, while comparing two intervals three possibilities are considered:
- one interval is completely to the right of the other (strict preference);
- one interval is to the right of the other, but they have a non empty intersection (weak preference);
- one interval is included in the other (indifference).

Our problem is double: define the necessary and sufficient conditions for which a PQI preference structure is a PQI interval order and define an algorithm which operationally verifies if the conditions of the theorem are satisfied by a given PQI preference structure.

2 PQI interval orders

The basic theorem which gives the necessary and sufficient conditions for a PQI preference structure to be a PQI interval order is the following.

Theorem 2.1 *A PQI preference structure on a finite set A is a PQI interval order iff it exists a partial order I_l such that:*
i) $I = I_l \cup I_r \cup I_o$ where $I_o = \{(x, x),\ x \in A\}$ and $I_r = I_l^{-1}$;
ii) $(P \cup Q \cup I_l).P \subset P$;
iii) $P.(P \cup Q \cup I_r) \subset P$;

iv) $(P \cup Q \cup I_l).Q \subset P \cup Q \cup I_l$;
v) $Q.(P \cup Q \cup I_r) \subset P \cup Q \cup I_r$;

Proof See Tsoukiàs and Vincke, 1999.

It is easy to see that the theorem is a formula in a second order logic (a formula where predicates can be variables). Generally the satisfaction of second order formula can be undecidable. Moreover the theorem does not give a constructive procedure for verifying its satisfaction. In the following we give a second theorem, equivalent to theorem 2.1, which enables to define an algorithm detecting if a PQI preference structure is a PQI interval order.

Theorem 2.2 *A PQI preference structure on a finite set A is a PQI interval order iff it exists a partial order* I_l *such that:*
i. $I = I_l \cup I_r \cup I_o$ *where* $I_o = \{(x,x),\ x \in A\}$ *and* $I_r = I_l^{-1}$;
ii. $P.Q \cup Q.P \cup P.P \subset P$ *and* $Q.Q \subset P \cup Q$;
iii.$(P.Q^{-1} \cap I) \subset I_l$;
iv.$(P^{-1}.Q \cap I) \subset I_l$;
v. $(I.I \cap P) \subset I_l.I_r$;
vi. $(I.I \cap (Q \cup Q^{-1})) \subset ((I_l.I_r) \cup (I_r.I_l))$
vii. $I_l.I_l \subset I_l$;

Proof See Tsoukiàs and Vincke, 1999.

We remind to the readers that a partial order is a reflexive and transitive binary relation.

3 The algorithm

Let S be a PQI preference structure on a finite set A. The algorithm will first verify condition ii and then construct I_l by applying directly conditions iii to vii of theorem 2.2. By definition, $I_r = I_l^{-1}$, i.e., the construction of I_l implies that of I_r. If the algorithm is able to build a relation I_l satisfying conditions of the theorem 2.2, then the PQI preference structure under investigation is a PQI interval order. If on the other hand it fails, then the PQI preference structure under investigation is not a PQI interval order. Failure of the algorithm can occur either because condition ii is not satisfied or because during the construction of I_l a contradiction occurs. A contradiction is defined as either a violation of the mutual exclusion of P, Q, I ($I_l(x,y)$ is established for $(x,y) \in P \cup Q \cup P^{-1} \cup Q^{-1}$) or a violation of the asymmetry of I_l (both $I_l(x,y)$ and $I_l(y,x)$ are established). The demonstration of formal correctness of the algorithm is in Appendix A.

Algorithm 3.1

Step 1: *if not* $(P.Q \cup Q.P \cup P.P \subset P$ *and* $Q.Q \subset P \cup Q)$ *then failure;*
$I_l = \emptyset$;

Step 2: $\forall\, x,y,z \in A,$ *if* $I(x,y) \wedge P(x,z) \wedge Q(y,z)$ *then*
if $(y,x) \in I_l$ *then failure else* $I_l = I_l \cup \{(x,y)\}$;

Step 3: $\forall\, x,y,z \in A,$ *if* $I(x,y) \wedge P(z,x) \wedge Q(z,y)$ *then*
if $(y,x) \in I_l$ *then failure else* $I_l = I_l \cup \{(x,y)\}$;

Step 4: $\forall\, x,y,z \in A,$ *if* $I(x,y) \wedge I(y,z) \wedge P(x,z)$ *then*
if $(y,x) \in I_l \vee (y,z) \in I_l$ *then failure*
else $I_l = I_l \cup \{(x,y),(z,y)\}$;

Step 5: $I'_l = I_l$;
repeat
$\forall\, x,y,z \in A,$ *if* $I_l(x,y) \wedge I(y,z) \wedge (Q \cup Q^{-1})(x,z)$ *then*
if $(y,z) \in I_l$ *then failure else* $I_l = I_l \cup \{(z,y)\}$;
$\forall\, x,y,z \in A,$ *if* $I_l(x,y) \wedge I(z,x) \wedge (Q \cup Q^{-1})(y,z)$ *then*
if $(z,x) \in I_l$ *then failure else* $I_l = I_l \cup \{(x,z)\}$;
until $I'_l = I_l$

Step 6: $\forall\, x,y,z \in A,$ *if* $I_l(x,y) \wedge I_l(y,z)$ *then*
if $(z,x) \in I_l \cup P \cup Q \cup P^{-1} \cup Q^{-1}$ *then failure*
else $I_l = I_l \cup \{(x,z)\}$;

Step 7: *If there is one* $I(x,y)$ *not yet established as* I_l *or* I_r*, choose one of them and set it as* $I_l(x,y)$*. Then return to 5. Otherwise stop.*

Steps 1 to 4, are deterministic, in the sense that each I_l established is mandatory. If a contradiction occurs, the algorithm fails. Steps 5 and 6 however, use already established I_l in order to establish further I_l. The problem arises from Step 7 where I_l is arbitrarily chosen. When the algorithm goes back to Step 5 to continue with establishing I_l, if a contradiction occurs, intuitively, it should backtrack to the last $I_l(x,y)$ established, reverse it to $I_l(y,x)$ and try again. In other terms the algorithm appears to have to explore a "tree structure" defined by the branches created by each arbitrary choice. In such a case the risk is to have to make an exhaustive research of the whole "tree".

In the following we will demonstrate that the algorithm previously presented is "backtracking free". In other words, any contradiction implies the non-existence of a PQI interval order on A and the algorithm can stop immediately without backtracking. Actually any failure in steps 1 to 6 will induce the algorithm to end with a negative answer. This is the reason for which the algorithm is presented without backtracking.

Theorem 3.1 *The algorithm 3.1 is backtracking free.*

Proof We elaborate the demonstration observing how the setting of $I_l(x, y)$ (steps 5, 6) is propagated and analyzing contradictory situations. The demonstration consists in decomposing the problem in smaller cases and showing for each of them that when a contradiction occurs there is no backtracking necessity and the algorithm fails (the PQI preference structure is not a PQI interval order).

Before reaching step 7 the first time, the process is deterministic, we can therefore construct the graph $G_0 = (A, V_0)$ where A is the usual set of objects on which the PQI preference structure applies and $V_0 = P \cup Q \cup I \cup I_l$ where I consists of (x, y) which are not yet set. The undirected graph associated to G_0 is complete and all its arcs can be directed except the ones in I. In the following we denote as a *"triangle"* a set of three elements in A (x, y, z) such that $x\Phi y\Psi z\Theta x$, where Φ, Ψ, Θ are any among $P, P^{-1}, Q, Q^{-1}, I_l, I_l^{-1}, I$.

Proposition 3.1 *In G_0, a triangle with at least an I-arc must be one of the following:*
1 - $I.I.I$
2 - $I.I.I_l$
3 - $I.I.Q$
4 - $I.I_l.I_l^{-1}$
5 - $I.I_l^{-1}.I_l$
6 - $I.P.P^{-1}$
7 - $I.P^{-1}.P$
8 - $I.Q.Q^{-1}$
9 - $I.Q^{-1}.Q$

Proof.
Immediate from Theorem 2.2 which excludes all other possibilities.

■

Denote as X-arc any arc representing relation X, X being one of P, Q, I, I_l. Denote as I-path a path where each of its arcs is an I-arc. Consider then the partial graph $G^* = (A, V_1)$ where $V_1 = \{(x, y) | x \neq y, \ \exists\ I\text{-path from } x \text{ to } y\}$.

Proposition 3.2 *G^* consists of connected components which:*
i. undirected associated graphs are complete;
ii. do not contain any P-arc;
iii. are closed to the propagation of the setting of I_l.

We have proved that G^* consists of connected components in which the propagation of the setting of $I_l(x, y)$ is limited. Each component contains only Q- or I- or I_l- arcs, while P-arcs exist only among such components. Therefore, we can limit ourselves in analyzing only one connected component, denoted by $G_1 = (A_1, V_1)$.

Let (x^*, y^*) be the I-arc arbitrarily chosen in step 7 to become an I_l-arc. Denote as I_l^k the set of I-arcs set in I_l in the current step and as I_l^K the cumulative set of I-arcs set in I_l until the current step included. We have that $I_l^K = I_l^k \cup I_l^{K-1}$. Conventionally, in step 5, (x^*, y^*) is added to I_l^k,i.e., as it is set in the step 5.

Proposition 3.3 *I-arcs set to I_l by transitive closure (step 6) are never used when the algorithm iterates step 5.*

Denote as a q-path a path whose arcs are Q or Q^{-1} ones. In the set A, let us consider now the following equivalence relation: $\Theta(x, y) \Leftrightarrow \exists$ a q-path from x to y and use X, Y, Z to denote equivalence classes. Therefore we can see graph G_1 as composed by equivalence classes of nodes each of which contains only Q-, I- and I_l- arcs. Further on among such equivalence classes only I- and I_l- arcs do exist.

Proposition 3.4 *In step 5*
i - the propagation of $I_l(x, y) \in X \times Y$ is limited to $X \times Y$.
ii - when $X \neq Y$,the propagation of I_l covers the whole set $X \times Y$.
iii - If $(x^*, y^*) \in X \times X$ then $I_l^k \subset X \times X$
iv - If $(x^*, y^*) \in X \times Y$, $X \neq Y$ then $I_l^k = X \times Y$.
v - Whatever (x, y) is chosen to be set in I_l in Step 5 the result is the same.
vi - If $I_l(y^*, x^*)$ is chosen instead of $I_l(x^*, y^*)$ then all the settings in this step will be reversed.

Proposition 3.4 states that, during the k-th iteration of the algorithm, Step 5 sets to I_l some I-arcs included in an equivalence class (of relation Θ) and all I-arcs among the equivalence classes. Consider now Step 6. In each application of step 6, setting $I_l(x, z)$ from $I_l(x, y)$ and $I_l(y, z)$, implies that at least one arc, let's say (x, y), has to be set during, either this step, or the two last steps 5,7. In a formal notation we have:

Proposition 3.5 *In Step 6:*
i - If $(x, y) \in X \times X$ *then* $z \in X$.
ii - If $(x^*, y^*) \in X \times X$ *then* $I_l^k \subset X \times X$.
iii - If it exists $I_l^k(x, z) \in X \times Z,\ X \neq Z$ *then* $X \times Z \subset I_l^k$.
iv - If $(x^*, y^*) \in X \times Y,\ X \neq Y$ *((*x^*, y^**) chosen in the last step 7), only arcs connecting different classes are set in Step 6 (in other terms if* $I_l(x, z) \in X \times Z$ *is set in Step 6 then* $Z \neq X \wedge Z \neq Y$*).*

These results show that if we choose an arc (x^*, y^*) to set in I_l, if it is inside one equivalent class it does not propagate I_l outside this class, while if it connects two different classes, it does not propagate I_l into any class. Furthermore, as the algorithm has passed through steps 5, 6 before the establishment of G_1 at least once, all the arcs between two classes X, Y are of the same type (either I-arcs or I_l-arcs). Therefore, the problem can be further decomposed into two sub-problems:
a) - Outside all the equivalent classes, we consider the same problem with G_1 replaced by $G_2 = (A_2, V_2)$ where A_2 is the quotient set A^Θ and V_2 consists of two types of arcs: $I(X, Y)$ if $\exists(x, y) \in X \times Y$ such that $I(x, y)$ holds, and $I_l(X, Y)$ if $\exists(x, y) \in X \times Y$ such that $I_l(x, y)$ holds.
b) - Inside each equivalent class, we consider the same problem with G_1 replaced by $G_3 = (A_3, V_3)$.

The sub-problem a) is trivial, as the graph G_2 contains only I or I_l arcs, furthermore, the part of G_2 covered by I_l-arcs is already I_l transitively closed since the algorithm has already gone through Step 6. The problem is reduced to the construction of a linear order. Therefore, we have to deal only with the sub-problem (b).

We have to demonstrate now that the algorithm is backtracking free on G_3 where the arcs are Q, I_l, I and there is a q-path connecting any two different nodes. We consider now the possible situations where a contradiction may occur.

Proposition 3.6 *In step 5*
i - $I_l^k(x, y) \wedge I_l^k(y, z) \Rightarrow I_l^k(x, z)$ *i.e. if* (x, y) *and* (y, z) *are set in this step, then so is* (x, z).
ii - $I_l^k(x, y) \wedge I_l^{K-1}(y, z) \wedge I_l^k(z, t) \Rightarrow I_l^k(x, t)$.
iii - $I_l^{K-1}(x, y) \wedge I_l^{K-1}(y, z) \Rightarrow I_l^{K-1}(x, z)$.

N.B. We may emphasise that, while in Step 5, $I_l^k(x, y) \wedge I_l^{K-1}(y, z)$ does not necessarily imply $I_l^k(x, z)$.

Proposition 3.7 *In step 5, an I_l-circuit occurs only with a contradiction.*

Proposition 3.8 *If the first contradiction occurs at step 6, then there must be an I_l circuit at the end of step 5 (an I_l^{K-1} circuit).*

Proposition 3.9 *If the first contradiction occurs at step 5, then the problem has no solution.*

From Proposition 3.8 if a contradiction occurs in Step 6 there is an I_l circuit at Step 5. From Propositions 3.6 and 3.7 if such a circuit exists in Step 5 it has to exist also a contradiction in Step 5. And from Proposition 3.9 if a contradiction occurs at Step 5, the problem has no solution and it is not necessary to make any backtracking. And this concludes our demonstration.
■

4 Matrix version of the algorithm

From the previous discussion it is easy to see that the critical part of the PQI graph to analyze is the G_3 graph, so we may study complexity with respect to this subgraph. In the following we give a way to implement the algorithm and discuss its complexity. Let $A = \{a_1, a_2, ...a_n\}$ and let P, Q, I, L be $n \times n$ matrixes representing relations P, Q, I, I_l respectively, where: $X_{ij} = 1 \Leftrightarrow a_i X a_j$, otherwise $X_{ij} = 0$, X being one among P, Q, I, I_l.

Theorem 4.1 *Algorithm 3.1 is in polynomial time ($O(n^5)$)*

Proof The algorithm presented in the previous section can be represented in the following way (including some small variations discussed immediately after):

Algorithm 4.1
Step 1: $P_{ij} + P_{jk} \leq 1 + P_{ik}$, $P_{ij} + Q_{jk} \leq 1 + P_{ik}$, $Q_{ij} + Q_{jk} \leq 1 + P_{ik} + Q_{ik}$ $\forall\, i, j, k = 1..n$;
Step 2: $I_{ij} = P_{ik} = Q_{jk} = 1 \Rightarrow L_{ij} = 1 \;\forall\, i, j, k = 1..n$;
Step 3: $I_{ij} = P_{ki} = Q_{kj} = 1 \Rightarrow L_{ij} = 1 \;\forall\, i, j, k = 1..n$;
Step 4: $P_{ij} = I_{ik} = I_{kj} = 1 \Rightarrow L_{ik} = L_{kj} = 1 \;\forall\, i, j, k = 1..n$;
Step 5: $Q_{ij} + Q_{ji} = I_{ik} = I_{kj} = 1 \Rightarrow L_{ik} = L_{kj} \;\forall\, i, j, k = 1..n$;
Step 6: $L_{ij} = L_{jk} = 1 \Rightarrow L_{ik} = 1 \;\forall\, i, j, k = 1..n$;
Step 7: For $I(x, y)$ not yet established as I_l or I_r, choose arbitrarily $I_l(x, y)$. If the I_l established belongs to an equivalence class established in Step 5, put all the elements of the class equal to 1. Return to 6 (instead of 5).

A critical step in this algorithm is step 5 since it introduces implicitly a recursive establishment of I_l. In order to avoid an infinite recursion and the associate contradictions it is necessary to "fix" I_l as soon as it is generated by step 5 so that only $I(x, y)$ which are not yet established may still be considered in the recursive application of step 5. This is possible partitioning the set of non zero elements of the matrix I into classes which will have the same value of L_{ij} because of step 5. Then as soon as one element of one of these classes turns to 1, the whole class will turn to 1. Under such an adjustment the following positive consequences hold:
- if there is no solution then a contradiction in establishing an I_l will appear before step 6;
- after step 7 you just have to return to step 6.

We can now discuss complexity. Steps 1 to 4 are obviously in $O(n^3)$ as step 6 (transitive closure) is. Step 5 is in $O(n^5)$ as can be seen by the following implementation (remark that in the worst case $n = |G_3|$):

```
function step5: boolean
      forall i, j, k
           if (Iik*Ikj*(Qij+Qji) == 1)
                if ( not setLabel(i,j,k) )
                     return false
      return true
function setLabel(i,j,k: integer)
      if (Lik, Lkj no label)
           set new label to Lik and Lkj
      else if (Lik = L1, Lkj no label)
           set Lkj to L1
      else if (Lik no label, Lkj = L2)
           set Lik to L2
      else if (Lik = L1 et Lkj = -L1)
           return false (conflict)
      else if (Lik = L1 et Lkj = L2)
           unify these two labels
      endif
      return true
```

Furthermore it is easy to see that the decomposition of the PQI graph in G_1 and its connected components, the decomposition in G_2 and G_3 and the construction of the linear order in G_2 are all in polynomial time. Therefore the whole algorithm is in polynomial time.

■

5 Conclusions

The paper presented an operational solution to how a PQI preference structure on a finite set A can be checked to be or not a PQI interval order. In other words verify if it is possible to associate to each element of A an interval such that if the interval associated to x is completely to the right of the interval associated to y, then x is strictly preferred to y, if one interval is included to the other, then x is indifferent to y and if the interval associated to x is to the right of the interval associated to y, their intersection being not empty, then x is weakly preferred to y.

In the paper the necessary and sufficient conditions for such a case are introduced and an algorithm for the satisfaction of such conditions is presented. We first demonstrate that the algorithm, although appears that has to explore a tree generated by branches of arbitrary choices, is backtracking free and then we demonstrate that runs in polynomial time. We consider such a result very promising, since it enables an efficient check of the existence of PQI interval orders which are very common in many different cases, including preference medelling and temporal logic. In fact, PQI interval orders are very useful in representing discrete states of preference hesitation. Being able to detect if a PQI preference structure is a PQI interval order allows to know if its numerical representation is meaningful. Further on, since we conjecture that this result can be generalised in the case of preference structures with multiple thresholds, the existence of an efficient algorithm allows to hope for an easy extension of this theory in the case of multiple interval orders, a long time open problem in preference modelling.

Acknowledgements

This paper has been presented during the IFORS 99 conference in Beijing as a French national contribution. We acknowledge the support received by ROADEF (the French OR society). The first author gratefully acknowledge the support received by ESSEC in order to attend this conference. Two anonymous referees helped us in improving the presentation of the paper. We gratefully acknowledge their help.

References

1. Fishburn, P.C., 1985. Interval Orders and Interval Graphs. J. Wiley, New York.

2. Pirlot, M., Vincke, Ph., 1997. Semi Orders. Kluwer Academic, Dordrecht.

3. Roubens, M., Vincke, Ph., 1985. Preference Modeling. Springer Verlag, Berlin.

4. Tsoukiàs, A., Vincke, Ph., 1997. Extended preference structures in MCDA. In: Clímaco, J. (Ed.), Multicriteria Analysis, Springer Verlag, Berlin, pp. 37-50.

5. Tsoukiàs, A., Vincke, Ph., 2000. A characterisation of PQI interval orders. To appear in Discrete Applied Mathematics.

6. Vincke, Ph., 1988. P,Q,I preference structures. In Kacprzyk, J., Roubens, M., (Eds.), Non conventional preference relations in decision making, LNEMS 301, Springer Verlag, Berlin, pp. 72-81.

Appendix A

Proof of Algorithm 3.1

When step 1 succeeds, condition ii holds and $I_l = \emptyset$. As P, Q are invariable during steps 2 to 7, condition ii always holds.
When step 2 succeeds, condition iii holds for I_l. As I_l is never reduced in all the following steps, condition iii always holds. In the same way, conditions iv, v hold after steps 3 and 4, and always hold thereafter.
It is obvious that I_l is always asymmetric after each step.
When the algorithm passes through the step 6 the first time, the following condition holds:
$(V): \ (I.I \cap (Q \cup Q^{-1})) \subset ((I_l.I_r) \cup (I_r.I_l) \cup ((I \setminus (I_l \cup I_r \cup I_0)).(I \setminus (I_l \cup I_r \cup I_0)))$
and $I_l.I_l \subset I_l$ and I_l asymmetric.
It is easy to verify that steps 7,5,6 form a loop accepting (V) as an invariant (a condition that always holds before and after each iteration of the loop).
The ending condition of the loop is $(E): \ I \setminus (I_l \cup I_r \cup I_0) = \emptyset$.
When the algorithm succeeds, both (V) and (E) hold which induces conditions i, vi, vii and I_l is a partial order.

Proof of Proposition 3.2

i. If x, y belongs to a connected component then exists a path (in G^*) $a_0 = x, a_1, ...a_k = y$. $\forall \ i = 0...k-1$, if exists an I-path from a_i to a_{i+1} then exists an I-path from x to y and therefore $(x, y) \in V_1$.
ii. If a P-arc exists, denote it $P(x, y)$ and consider the length k of the I-path $a_0 = x, a_1, ...a_k = y$ to be minimal (among all P-arcs and all (x, y)). Consider then the arc a_1, a_k (it exists from the completeness of the component), then from proposition 3.1 we have $P(a_1, a_k)$ and therefore we have another P-arc with length of the I- path $< k$. Impossible.
iii. Immediate from conditions vi and vii of Theorem 2.2 (steps 5 and 6 of the algorithm).

■

Proof of Proposition 3.3

First consider (x_1, x_2) such that $I_l^k(x_1, x_2)$ in step 6. Therefore it exists $I_l(x_1, x_3) \wedge I_l(x_3, x_2)$. If, for example, (x_1, x_3) was also established in I_l^k (in the current step 6) then it exists $I_l(x_1, x_4) \wedge I_l(x_4, x_3)$ and so on until an I_l^{K-1}-path is obtained. Therefore for all (x, y) such that I_l is established in the current step 6 exists an I_l^{K-1}-path from x to y.
Let now (x, y) to be an arc set to I_l in the last step 6, participating to the setting of arc (x, z) in step 5 through let's say $Q(z, y)$. Let us consider the

situation in the last step 6:
$I_l^k(x,y) \Rightarrow \exists\ I_l^{K-1}$-path $t_0 = x, t_1, ...t_k = y$.
Consider the triangle z, t_{k-1}, y where $Q(z,y) \wedge I_l^{K-1}(t_{k-1}, y)$.
If $Q(t_{k-1}, z)$ then $Q(t_{k-1}, z) \wedge Q(z, y) \Rightarrow (P \cup Q)(t_{k-1}, y)$, conflict with $I_l(t_{k-1}, y)$.
If $I(t_{k-1}, z)$ then $I_l^{K-1}(t_{k-1}, y) \wedge Q(z, y) \wedge I(t_{k-1}, z) \Rightarrow I_l^{K-1}(t_{k-1}, z)$ (at least in the last step 5). Therefore it exists an I_l^{K-1}-path from x to z, that is $I_l(x,z)$ must be set at least at the same time as (x,y). We conclude that $Q(z, t_{k-1})$. Repeat this procedure, and we get at last $Q(z, t_1)$, which together with $I_l^{K-1}(x, t_1)$ gives $I_l^{K-1}(x,z)$ i.e. (x,z) must have been set before (x,y).

■

Proof of Proposition 3.4
i - In each application of step 5, consider $(x,y) \in X \times Y$ such that $I_l(x,y)$. Relation I_l will propagate to (x', y) or (x, y'), x', y' arbitrary. There have to exist q-paths from x to x' and from y to y'. Therefore $(x', y') \in X \times Y$.
ii - $(x', y') \in X \times Y$ implies that there exist q-paths $a_0 = x, a_1, ...a_k = x'$, and $b_0 = y, b_1, ...b_l = y'$. Applying consecutively step 5 on these two paths we obtain the setting in I_l of $(x,y), (a_1, y), ...(x', y)$ and then of $(x', b_1), (x', b_2), ...(x', y')$.
iii and iv - Immediate from propositions (3.3), (3.4.i) and (3.4.ii).
v and vi - Immediate from Theorem 2.2.

■

Proof of Proposition 3.5
i - Otherwise, consider the first setting with $z \in Z \neq X$. It implies that $I_l(y,z) \in X \times Z,\ Z \neq X$ and since (x,z) is the first such setting, $I_l^{K-1}(y,z)$ holds. We have $x, y \in X \wedge z \in Z \wedge I_l^{K-1}(y,z)$ which implies $I_l^{K-1}(x,z)$ as it must be set at least in the last step 5 (proposition (3.4.ii)). Contradiction.
ii - Immediate from (3.4.iii),(3.5.i).
iii - Otherwise it should exist $(x', z') \in X \times Z \setminus I_l^k$. In the next step 5 (x,z), which is set in this step 6, will propagate I_l to (x', z'), which is impossible because of (3.3) ((x,z) set in step 6, cannot be used in the step 5).
iv - Suppose that $(x^*, y^*) \in X \times Y,\ X \neq Y$ is introduced in step 5. Then all the arcs of $X \times Y$ are set to I_l and only these arcs. The setting in step 6 is the propagation of such arcs. Let $I_l(x,y) \wedge I_l(y,z) \Rightarrow I_l(x,z),\ x \in X,\ y \in Y,\ z \in Z$ the first setting in step 6 with $I_l^{K-1}(x,y)$, $I_l^{K-1}(y,z)$ and (x,y) set in the last steps 5,7, i.e. $(x,y) \in X \times Y$ and $X \times Y \subset I_l^{K-1}$. If $Z = X$ then (z,y) should have been set at the same time as (x,y), which contradicts $I_l(y,z)$. We conclude that $Z \neq X$ (and similarly that $Z \neq Y$).

$\blacksquare$

Proof of Proposition 3.6
i - Let $\mathcal{Q} = Q \cup Q^{-1}$, in an equivalence class we have $\forall(x,y)\ \Psi(x,y)$ where $\Psi = I_l$ or I_l^{-1} or $\mathcal{Q}$.
If in Step 5 we had $\mathcal{Q}(x,z)$ then $I_l^k(x,y) \wedge \mathcal{Q}(x,z) \wedge I(z,y) \Rightarrow I_l^k(z,y)$, in contradiction with $I_l^k(y,z)$. Therefore we have $\neg\mathcal{Q}(x,z)$ and $\Psi = I_l$ or I_l^{-1}. The transition from $I_l^k(x,y)$ to $I_l^k(y,z)$ in step 5 passes through 2 q-paths $x_1 = x, x_2, ...x_n = y$ and $y_1 = y, y_2, ...y_n = z$ where ($x_i = x_{i+1}$ and $y_i \neq y_{i+1}$ and $I(x_i, y_i), I(x_{i+1}, y_{i+1})$) or ($x_i \neq x_{i+1}$ and $y_i = y_{i+1}$ and $I(x_i, y_i), I(x_{i+1}, y_{i+1})$). We consider the two different transitions from (x,y) to (y,z).
1. If $y_2 = y$ then $x \neq x_2$ and therefore $\mathcal{Q}(x,x_2) \wedge I(x_2,y)$. We have then $\mathcal{Q}(x,x_2) \wedge I_l^k(x,y) \wedge I(x_2,y) \Rightarrow I_l^k(x_2,y)$. But $\mathcal{Q}(x_2,z)$ is in contradiction with $I_l^k(x_2,y)$. Therefore $I_l^k(y,z) \Rightarrow \neg\mathcal{Q}(x_2,z) \Rightarrow I(x_2,z)$.
We have $\mathcal{Q}(x,x_2) \wedge I(x_2,z) \wedge \Psi(x,z) \Rightarrow \Psi(x_2,z)$. Therefore the situation is not changed (x_2 plays now the role of x).
2. If $y_2 \neq y$ then $x = x_2$. Therefore $\mathcal{Q}(y,y_2) \wedge I(x,y_2) \Rightarrow I_l^k(x,y_2)$. If $\mathcal{Q}(y_2,z)$ holds we have $\mathcal{Q}(y_2,z) \wedge I_l^k(x,y_2) \wedge I(x,z) \Rightarrow I_l^k(x,z)$. Otherwise, $\mathcal{Q}(y,y_2)$ and $I_l^k(y,z)$ give $I_l^k(y_2,z)$ and the situation is not changed (y_2 plays now the role of y).
In order to pass from y to z, it must exist a k such that $y_{k+1} = z$ and $y_k \neq z$, i.e. $\mathcal{Q}(y_k, y_{k+1}) \Rightarrow \mathcal{Q}(y_k, z) \Rightarrow I_l^k(x,z)$ (since it always holds $I_l(x, y_k)$).
ii - Let $\Psi(x,t)$. Since $\mathcal{Q}(y,t)$ or $\mathcal{Q}(x,z)$ is in contradiction with $I_l^k(x,y)$, $I_l^{K-1}(y,z)$ and $I_l^k(z,t)$ we have $I(y,t)$ and $I(x,z)$.
If $\Psi = \mathcal{Q}$ then $I_l^k(x,y) \wedge I(y,t) \wedge \mathcal{Q}(x,t) \Rightarrow I_l^k(t,y)$. But $I_l^k(t,y) \wedge I_l^k(z,t) \Rightarrow I_l^k(z,y)$ (3.6.i) in contradiction with $I_l^{K-1}(y,z)$. Therefore $\Psi = I_l \cup I_{l-1}$. The transition from $I_l^k(x,y)$ to $I_l^k(z,t)$ in step 5 passes through 2 q-paths $x_1 = x, x_2, ...x_n = z$ and $y_1 = y, y_2, ...y_n = z$ where ($x_i = x_{i+1}$ and $y_i \neq y_{i+1}$ and $I(x_i, y_i), I(x_{i+1}, y_{i+1})$) or ($x_i \neq x_{i+1}$ and $y_i = y_{i+1}$ and $I(x_i, y_i), I(x_{i+1}, y_{i+1})$).
We consider the two different transitions from (x,y) to (z,t).
1. If $y_2 = y$ then $x \neq x_2$ therefore $\mathcal{Q}(x,x_2)$.
$\mathcal{Q}(x,x_2) \wedge I_l^k(x,y) \wedge I(x_2,y) \Rightarrow I_l^k(x_2,y)$. If $\mathcal{Q}(x_2,t)$ we have
$\mathcal{Q}(x_2,t) \wedge I_l^k(x_2,y) \wedge I(y,t) \Rightarrow I_l^k(t,y)$. But $I_l^k(z,t) \wedge I_l^k(t,y) \Rightarrow I_l^k(z,y)$ (3.6.i), in contradiction with $I_l^{K-1}(y,z)$.
We have then $\neg\mathcal{Q}(x_2,t) \Rightarrow I(x_2,t)$. $\mathcal{Q}(x_2,x) \wedge I(x_2,t) \wedge \Psi(x,t) \Rightarrow \Psi(x_2,t)$ and the situation is not changed (x_2 plays now the role of x).
2. If $y_2 \neq y$ then $x_2 = x$, therefore $\mathcal{Q}(y,y_2)$.
$\mathcal{Q}(y,y_2) \wedge I_l^k(x,y) \wedge I(x,y_2) \Rightarrow I_l^k(x,y_2)$.

If $\mathcal{Q}(y_2, t)$, we have $\mathcal{Q}(y_2, t) \wedge I_l^k(x, y_2) \wedge \Psi(x, t) \Rightarrow I_l^k(x, t)$.
Otherwise, if $\mathcal{Q}(y_2, z)$ then $\mathcal{Q}(y_2, z) \wedge I_l^k(x, y_2) \wedge I(x, z) \Rightarrow I_l^k(x, z)$.
Therefore, we have $I_l^k(x, z) \wedge I_l^k(z, t) \Rightarrow I_l^k(x, t)$ (3.6.i).
If $I(y_2, z)$ then $\mathcal{Q}(y, y_2) \wedge I_l^{K-1}(y, z) \wedge I(y_2, z) \Rightarrow I_l^{K-1}(y_2, z)$ and the situation is not changed (y_2 plays now the role of y).
So, we have either $I_l(x, t)$ when it exists $\mathcal{Q}(y_i, z)$ or the only way to pass from y to t is through some $y_{k+1} = t$ and $y_k \neq t$ i.e. $\mathcal{Q}(y_k, y_{k+1}) \Rightarrow \mathcal{Q}(y_k, t) \Rightarrow I_l(x, t)$.
iii - If $I_l^{K-1}(x, y)$ and $I_l^{K-1}(y, z)$ are set at least in the last step 6 then $I_l^{K-1}(x, z)$ is also set at least in the last step 6.

■

Proof of Proposition 3.7

Let an I_l-circuit with arcs I_l^k or I_l^{K-1}. With (3.6.i), we can replace all I_l^k-paths with I_l^k-arcs. With (3.6.iii), we can replace all I_l^{K-1}-paths with I_l^{K-1}-arcs. We get at last an I_l-circuit with alternative I_l^k-arcs and I_l^{K-1}-arcs. Let l to be the length of the circuit.
If $l > 4$ then exists $I_l^k(x, y) \wedge I_l^{K-1}(y, z) \wedge I_l^k(z, t)$ which can be replaced by $I_l^k(x, t)$ and we obtain a new circuit with alternative I_l^k-arcs and I_l^{K-1}-arcs and its new length $l' = l - 2$. We get at last $l' = 3$ or $l' = 4$. If $l = 3$
$I_l^k(x, y) \wedge I_l^{K-1}(y, z) \wedge I_l^k(z, x) \Rightarrow I_l^k(z, y)$ (3.6.i) in contradiction with $I_l^{K-1}(y, z)$.
$I_l^{K-1}(x, y) \wedge I_l^k(y, z) \wedge I_l^{K-1}(z, x) \Rightarrow I_l^{K-1}(z, y)$ (3.6.iii) in contradiction with $I_l^k(y, z)$.
If $l = 4$ $I_l^k(x, y) \wedge I_l^{K-1}(y, z) \wedge I_l^k(z, t) \wedge I_l^{K-1}(t, x) \Rightarrow I_l^k(x, t)$ (3.6.ii) in contradiction with $I_l^{K-1}(t, x)$.

■

Proof of Proposition 3.8

Suppose that the first contradiction occurs at step 6, i.e. an arc (x, z) already set in Q or in Q^{-1} or in ${I_l^{-1}}_{K-1}$, is set in I_l. Therefore it exists an I_l^{K-1} path $x = z_1, ..., z = z_n$. If ${I_l^{-1}}_{K-1}(x, z)$ is the case then we have an I_l circuit $x..., z, x$ at the end of step 5. If $Q(x, z)$ or $Q^{-1}(x, z)$ is the case, consider the arc (x, z_{n-1}), If $I(x, z_{n-1})$ is the case then we have $I_l^{K-1}(z_{n-1}, x)$ (at least in step 5 for $I_l^{K-1}(z_{n-1}, z)$). We have then an I_l^{K-1} circuit $x, ..., z_{n-1}, x$ at the end of step 5. If $Q(x, z_{n-1})$ or $Q^{-1}(x, z_{n-1})$ is the case, we continue considering the arc (x, z_{n-2}) and so on. The process is finite and must end with either an I_l^{K-1} circuit or with $Q \cup Q^{-1}(x, z_3)$ which leads to a contradiction with $I_l^{K-1}(x, z_2)$ and $I_l^{K-1}(z_2, z_3)$.

■

Proof of Proposition 3.9
Consider (x^*, y^*) an arc arbitrarily set in I_l at the last step 7, and let $I_l(x, y)$ the first contradiction occurring at step 5, i.e. $I_l(y, x)$ has already been set before. From (3.3.ii) we know that $I_l(y, x)$ was not set in a previous Step 6. By (3.4.v) and (3.4.vi), we know that $I_l(x, y)$ has to be set during the current step 5, because otherwise (x^*, y^*) had to be also set and we could not choose it in Step 7. The problem has then no solution because if we choose to set (y*,x*) instead of (x*,y*), all the arcs set in step 5 in I_l^k will be reversed but the same contradiction will occur.

■