ALGORITHMS FOR THE EXTENSION OF PRECISE AND IMPRECISE CONDITIONAL PROBABILITY ASSESSMENTS: AN IMPLEMENTATION WITH MAPLE V

VERONICA BIAZZO

In this paper, we illustrate an implementation with Maple V of some procedures which allow to exactly propagate precise and imprecise probability assessments. The extension of imprecise assessments is based on a suitable generalization of the concept of coherence of de Finetti. The procedures described are supported by some examples and relevant cases.

1. Introduction.

The analysis of many real problems, involving uncertainty, often requires some probabilistic assessments on a suitable family $\mathcal{K}$ of random quantities. Such family has not necessarily any particular algebraic structure, therefore the de Finetti’s methodology is the most suitable one. Following this approach we examine some procedures by means of which some given conditional probability assessments can be propagated in a coherent way to a further conditional event. Based on the linear programming technique, the checking of the coherence and the extension of precise or imprecise assessments have been studies in many papers (see for example [5], [6], [7], [10], [12]). In [2] the fundamental theorem of de Finetti has been applied to conditional events and some theoretical results have been obtained. Moreover, in the quoted paper an

Entrato in Redazione il 25 ottobre 2000.
algorithm has been proposed to determine the interval \([p', p'']\). The consistency problem when an imprecise probability assessments.

\[ A_n : P(E_i | H_i) \geq \alpha_i, \quad i = 1, \ldots, n, \]

is defined on a family of \(n\) conditional events \(F_n = \{E_1 | H_1, \ldots, E_n | H_n\}\) can be examined by choosing some suitable definition of the concept of coherence (see [5], [7], [14], [15]). In particular, the definitions adopted in [5] and [7] are based on the coherence principle of de Finetti. In [14] and [15] some conditions of coherence, involving random quantities that can be interpreted as random gains, are introduced. Making a comparison ([12]) one has that the definitions adopted in [14] and [15] are stronger than that ones introduced in [5] and [7].

In [3] the extension of the results obtained in [2] to the case of imprecise assessments is examined. In particular, we have considered a suitable version of the fundamental theorem of de Finetti relative to the extension of an imprecise assessment \(A_n\) defined on \(F_n\) to a further conditional event \(E_{n+1} | H_{n+1}\). In this paper we examine in detail some procedures by means of which the theoretical results obtained in [2] (and also in [3]) can be applied. A preliminary version of the implementation of these procedures, relative to the case of precise conditional probability assessments, has been given in [1].

2. Preliminaries.

Given a coherent probability assessment \(P_n = (p_1, \ldots, p_n)\) on a family \(F_n = \{E_1 | H_1, \ldots, E_n | H_n\}\), let \(E_{n+1} | H_{n+1}\) be a further conditional event. As well known, there exists an interval \([p', p'']\) \(\subseteq [0, 1]\) such that, for every \(p_{n+1} \in [p', p'']\), the assessment \(P_{n+1} = (P_n, p_{n+1}) = (p_1, \ldots, p_n, p_{n+1})\) is a coherent extension of \(P_n\) to the family \(F_{n+1} = F_n \cup \{E_{n+1} | H_{n+1}\} = \{E_1 | H_1, \ldots, E_n | H_n, E_{n+1} | H_{n+1}\}\).

Concerning the case of imprecise assessments we recall that, given a family \(F_n = \{E_1 | H_1, \ldots, E_n | H_n\}\) and a vector \(A_n = (\alpha_1, \ldots, \alpha_n)\) of lower bounds \(P(E_i | H_i) \geq \alpha_i\), with \(i = 1, \ldots, n\), the following definition of generalized coherence (g-coherence) has been adopted in [7].

**Definition 1.** The vector of lower bounds \(A_n\) on \(F_n\) is said g-coherent if and only if there exists a (precise) coherent assessment \(P_n = (p_1, \ldots, p_n)\) on \(F_n\), with \(p_i = P(E_i | H_i)\), which is consistent with \(A_n\), that is such that \(p_i \geq \alpha_i\) for each \(i\).
Then, based on the definition above, a necessary and sufficient condition of \( g \)-coherence for the extension of imprecise assessments will be given below. Let \( \Pi_{\mathcal{A}_n} \) be the set of coherent precise assessments \( \mathcal{P}_n \) on \( \mathcal{F}_n \) which are consistent with \( \mathcal{A}_n \). We observe that for each coherent assessment \( \mathcal{P}_n \) on \( \mathcal{F}_n \) there exists an interval \([p^{'}, p^{'\prime}]\) of the coherent extensions of \( \mathcal{P}_n \) to \( E_{n+1}|H_{n+1} \). In [3] it is proved that, defining \( p^\circ = \max_{p_e \in \Pi_{\mathcal{A}_n}} p^\prime \), for each \( \alpha \in [0, p^\circ] \) the following assessment on \( \mathcal{F}_{n+1} \):

\[
P(E_i|H_i \geq \alpha_i, i = 1, \ldots, n, \ P(E_{n+1}|H_{n+1}) \geq \alpha
\]

is coherent.

In the same way, given a \( g \)-coherent vector of upper bounds \( \mathcal{B}_n = (\beta_1, \ldots, \beta_n) \) on a family \( \mathcal{F}_n \) and a further conditional event \( E_{n+1}|H_{n+1} \), the vector \( \mathcal{B}_{n+1} = (\beta_1, \ldots, \beta_n, \beta) \) on \( \mathcal{F}_{n+1} \) is \( g \)-coherent if \( \beta \in [p_\circ, 1] \), where \( p_\circ = \min_{p_e \in \Pi_{\mathcal{B}_n}} p^\prime \). A similar result has been obtained in [12] using the (stronger) definition of coherence given in [15].

Then, in [3] the following result has been obtained.

**Theorem 1.** Given a \( g \)-coherent imprecise assessment \( \mathcal{A}_n = (\{\alpha_i, \beta_i\}, i \in J_n) \) on the family \( \mathcal{F}_n = \{E_i|H_i, i \in J_n\} \), the extension \([\alpha_{n+1}, \beta_{n+1}]\) of \( \mathcal{A}_n \) to a further conditional event \( E_{n+1}|H_{n+1} \) is \( g \)-coherent if and only if the following condition is satisfied

\[
[\alpha_{n+1}, \beta_{n+1}] \cap [p_\circ, p^\circ] \neq \emptyset.
\]

We briefly introduce some preliminary aspects. Let us consider a family \( \mathcal{F}_{n+1} = \{E_i|H_i, \ldots, E_n|H_n, E_{n+1}|H_{n+1}\} \) and a coherent precise assessment \( \mathcal{P}_n \) (in particular, a \( g \)-coherent vector \( \mathcal{A}_n \) of lower bounds) on \( \mathcal{F}_n \). We also enclose within square brackets the modifications relative to the case of imprecise assessments.

We consider the following assessment on \( \mathcal{F}_{n+1} \)

\[
(\mathcal{P}_{n+1}) : P(E_i|H_i) = p_i, 1 = 1, \ldots, n, \ P(E_{n+1}|H_{n+1}) = p_{n+1},
\]

\[
[(\mathcal{A}_{n+1}) : P(E_i|H_i) \geq \alpha_i, 1 = 1, \ldots, n, \ P(E_{n+1}|H_{n+1}) = p_{n+1}]
\]

where \( p_{n+1} \) is not fixed.

Let us denote by \( \Pi \) the partition of \( \Omega \) obtained by expanding the expression

\[
(E_1H_1 \lor E_1^cH_1 \lor H_1^c) \land \ldots \land (E_{n+1}H_{n+1} \lor E_{n+1}^cH_{n+1} \lor H_{n+1}^c)
\]

and by \( C_1, \ldots, C_m \) the atoms or constituents of \( \Pi \) contained in \( H_0 = H_1 \lor \ldots \lor H_{n+1} \). Moreover, for each given \( i \), we denote respectively by \( \Gamma_i \) and \( F_i \).
the sets of subscripts $r$ such that $C_r \subseteq H_i$ and $C_r \subseteq E_iH_i$. We associate with pair $(\mathcal{F}_{n+1}, \mathcal{P}_{n+1}[(\mathcal{F}_{n+1}, \mathcal{A}_{n+1})]$ the following system in the unknowns $\lambda_1, \ldots, \lambda_m$, $p_{n+1}$

$$
\begin{align*}
\sum_{r \in \mathcal{F}_{n+1}} \lambda_r &= p_{n+1} \sum_{r \in \mathcal{F}_{n+1}} \lambda_r \\
\sum_{r \in \mathcal{F}_i} \lambda_r &= p_i \sum_{r \in \mathcal{F}_i} \lambda_r, & i = 1, \ldots, n; \\
\sum_{r} \lambda_r &= 1; & \lambda_r \geq 0.
\end{align*}
$$

[replace the equalities in the second row by $\sum_{r \in \mathcal{F}_i} \lambda_r \geq \alpha_i \sum_{r \in \mathcal{F}_i} \lambda_r$ $i = 1, \ldots, n$].

In what follows we assume that $p_{n+1}$ has a fixed value. Then, we denote respectively by $\Lambda$ and $\delta$ the vector of unknowns and the set of solution of the system (2). Moreover, for each $j$ we define the linear function $\phi_j(\Lambda) = \sum_{r \in \mathcal{F}_j} \lambda_r$, and we denote by $I_0$ the strict subset of $J_0 = \{1, \ldots, n+1\}$ defined as

$$
I_0 = \{j \in J_0 : \max_{\Lambda \in \delta} \phi_j(\Lambda) = 0\}.
$$

The following algorithm (see [2], [3]) allows to compute $p'$ (respectively $p''$) [$p_0$ (respectively $p^0$)], where the modifications needed to compute $p_0$ and $p^0$ are enclosed within square brackets. The case of precise conditional probability assessments has been also studied in [4], [5], [6], [13].

**Algorithm 1.** Let be given the pair $(\mathcal{F}_n, \mathcal{P}_n)[(\mathcal{F}_n, \mathcal{B}_n)$ (respectively $(\mathcal{F}_n, \mathcal{A}_n)])$ and the conditional event $E_{n+1}|H_{n+1}$.

1. **Step 0.** By expanding the expression

$$
\bigwedge_{i \in I} (E_iH_i \lor E_i^cH_i \lor H_i^c),
$$

where $I = \{1, \ldots, n+1\}$, determine the constituents $C_r$ contained in $H_0$. Then, construct the system (2).

2. **Step 1.** Check the compatibility of the system (2) under the condition $p_{n+1} = 0$ (respectively $p_{n+1} = 1$). If the system (2) is not compatible goto Step 2, otherwise go to Step 3;

3. **Step 2.** Solve the linear programming problem:

$$
\text{Compute } \gamma' = \max_{r \in \mathcal{F}_{n+1}} \sum_{r \in \mathcal{F}_{n+1}} \lambda_r
$$
and/or \( \gamma'' = \min \sum_{r \in F_{n+1}} \lambda_r \)

subject to:

\[
\begin{align*}
\sum_{r \in F_i} \lambda_r &= p_i \sum_{r \in \Gamma_j} \lambda_r, & i = 1, \ldots, n; \\
\sum_{r \in \Gamma_{n+1}} \lambda_r &= 1; & \lambda_r \geq 0.
\end{align*}
\]

The minimum \( \gamma' \) (respectively the maximum \( \gamma'' \)) of the objective function coincides with \( p' \) (respectively with \( p'' \)), and the procedure stops; [by replacing the equalities in the first row of the constraints by \( \sum_{r \in F_i} \lambda_r = \alpha_i \sum_{r \in \Gamma_j} \lambda_r, i = 1, \ldots, n \), and applying the algorithm with \( p_{n+1} = 1 \) (respectively \( p_{n+1} = 0 \) when the pair \((F_n, B_n)\) is considered) we obtain \( \gamma'' = p^o \) (respectively \( \gamma' = p_o \))]

- **Step 3.** For each subscript \( j \), compute the maximum \( M_j \) of the function \( \Phi_j \), subject to the constraints given by the system (2) with \( p_{n+1} = 0 \) (respectively \( p_{n+1} = 1 \)). We have the following three cases:

  1. \( M_{n+1} > 0 \);
  2. \( M_{n+1} = 0 \), \( M_j > 0 \) for every \( j \neq n + 1 \);
  3. \( M_j = 0 \) for \( j \in I_0 = J \cup \{n + 1\} \), with \( J \neq \emptyset \).

In the first two cases it is \( p' = 0 \) (respectively \( p'' = 1 \)) and the procedure stops.

In the third case replace the pair \((F_n, \mathcal{P}_n)\) by \((F_j, \mathcal{P}_j)\) and \( I \) by \( I_0 \), then, go to Step 0.

The algorithm ends in a finite number of runs by computing the minimum \( p' \) (respectively the maximum \( p'' \)) of the values \( p_{n+1}[\beta_{n+1} \text{ (respectively } \alpha_{n+1})] \) which are coherent (g-coherent) extensions of \( p_1, \ldots, p_n \)[(\( \beta_1, \ldots, \beta_n \)) (respectively \( \alpha_1, \ldots, \alpha_n \))].

### 3. Some procedures and examples.

In this section we examine the implementation of some procedures which allow to expand expression (1) and to construct the system (2), that is to execute the instructions at step 0 of the algorithm (1). The procedures concerning step 1
and step 2 exploit the *simplex package* of Maple V, therefore are not reported in this paper. Finally, some examples are considered.

In order to compute the constituents the logic operators *and*, *or* and the procedure *distrib*, which expands a boolean expression, have been used. In order to introduce in a simpler way the logical relations among the constituents, the *not* operator has been replaced by the following equivalent procedure, termed *c*, which allows to define easily partitions of the certain event.

**Procedure c**

```plaintext
c := proc() local a,i,n,m,k; a := [args];
if type(op(a), function and op(0, op(1,a))) = c then RETURN (op(1,op(1,a))) fi;
if has(a, &or') then
    n := [seq(op(i, op(a)), i = 1 .. nops(op(a)))];
    m := '&and'('seq(c(n[k]), k = 1 .. nops(n)));
    RETURN(m) fi;
if has (a,'&and') then
    n := [seq(op(i, op(a)), i = 1 .. nops(op(a)))];
    m := '&or'('seq(c(n[k]), k = 1 .. nops(n)));
    RETURN(m) fi; 'c'(op(a)) end.
```

Given a partition \(\{E_1, \ldots, E_n\}\) of the certain event, the procedure *c* is exploited in the procedure *partition Om* described below to obtain, for each event \(E_i\), its contrary

\[
E_i^c = E_1 \lor \cdots \lor E_{i-1} \lor E_{i+1} \lor \cdots \lor E_n;
\]

**Procedure partition Om**

```plaintext
partition Om := proc() local k;
if nargs = 2 then c(args[1]) := args[2]; c(args[2]) := args[1]
else for i to nargs do if i = 1
then c(args[i]) := '&or'('seq(args[k], k = 2 .. nargs))
else c(args[i]) := seq(args[k], k = 1 .. i-1) &or seq(args[k], k = i+1 .. nargs)
fi od fi end.
```

To take into account logical relations, such as implication and so on, the following procedure *int-imp* drops from expression (1) the impossible terms by labelling them with *false*. To detect them the procedure changes each term to a list \(m\). Then the term is labelled *false* if there exists an element of \(m\) such that its contrary contains some other element of \(m\).

**Example.** Let us consider a partition \(\{A, B, C\}\). In this case

\[
A^c = B \lor C, \quad B^c = A \lor C, \quad C^c = A \lor B,
\]
so that the event \( ABC \), for example, is impossible. In fact, the procedure changes \( ABC \) to the list \([A, B, C]\) and since \( B \subset A' \) the procedure returns \textit{false}.

\textbf{procedure int-imp}

\texttt{int-imp := proc(x)local i,m,j,v,r,k,s; contr := x; r := x;}
\texttt{if has(r, ‘&and’) then m := convert(r, list); v := false;}
\texttt{for j to nops(m) while v = false do}
\texttt{if has(c(m[j]), ‘&or’) then}
\texttt{for k to nops(c(m[j])) do}
\texttt{if member(op(k, c(m[j])), m) = true then}
\texttt{contr := false; RETURN(false) else v := false fi od else}
\texttt{if member(c(m[j]), m) then contr := false; RETURN (false) else v := false fi fi od; contr end.}

Concerning the relation of implication, if for example there are events \( A_1, A_2; B_1, B_2, B_3; \ldots \) such that \( A_1 \Rightarrow A_2; B_1 \Rightarrow B_2 \Rightarrow B_3; \ldots \), each relation is represented by an ordered list and then the set of relation is represented by the following list \([[[A_1, A_2], [B_1, B_2, B_3], \ldots], \text{named logic-list}].\)

Then, to determine the constituents, using a general procedure (named \textit{Constituents}) the presence of \([A_1, A_2]\) and \([B_1, B_2, B_3]\) in logic-list is taken into account by executing the logical conjunction between expression (1) and, respectively, the events \( A_1^c \lor A_2; B_1^c \lor B_2; \lor B_3, \) and so on.

Obviously, unconditional events are represented as conditional ones, with the conditioning event being the sure event which is denoted by the symbol \( OM \).

The following procedures allow to construct the system (2). within the procedure \textit{Chk-ind} (described below) it is exploited the procedure \textit{Bln}, by means of which it is checked if the conjunction between two given events (denoted with \( x \) and \( y \)) coincides with the impossible event. The number of partitions of the sure event, given in input, is assigned as the value of a variable named \textit{n-part}.

\textbf{procedure Bln}

\texttt{Bln := proc(x,y) local conj;}
\texttt{c(true) := false; OM := true; conj := distrib(x and y);}
\texttt{if 0 < n–part then if has(conj, ‘&or’) then}
\texttt{conj := distrib(map(int-imp, conj))}
\texttt{else conj := distrib(int-imp(conj))}
\texttt{fi fi; if evalb(conj = false) = false then ans2 := true}
\texttt{else ans2 := false fi; OM := evaln(OM); ans2 end.}

For each conditional event \( E_i \mid H_i \) in the given family it is possible to determine \( \sum_{r \in F_i} \lambda_r \) and \( \sum_{r \in F_i} \lambda_r \) using the procedure \textit{chkind}. Concerning the first
sum the inputs must be the event $E_iH_i$ and the list of constituents, respectively
denoted with $x$ and $y$. To determine the second sum the inputs must be the
event $H_i$ and the list of constituents. For each constituent $C_i$, a variable $\lambda_i$ is
associated with the conjunction $\land C_i$ if the condition $\land C_i \neq \emptyset$ is satisfied.
Finally, the expression representing the sum (named $Inds$) is defined.

**procedure Chk-ind**

Chk-ind := proc(x,y) local j,k,q,r,s,o;
  $s := [seq(k,k = 1 .. nops(y))]; j := 0;$
  for $r$ to nops(y) do
    if Bln(x,op(r,y)) = true then
      $j := j+1; q[j] := s[r]; o := [seq(q[k],k = 1 .. j)]$;
      else $j := j$ fi od;
  $\text{Inds} := \text{sum}('L.(op(k,o))'{k' = 1 .. j}); \text{Inds end.}$

Many other procedures concerning geometrical aspects (also studied in
[8] and [9]) have been implemented. In particular they allow to determine
the convex hull $I$ associated with a precise conditional probability assessment
$\mathcal{P}$, defined on a finite family $\mathcal{F}$ of conditional events, and check the condition
$\mathcal{P} \in I$. Due to the lack of space, these procedures here are not described.

In the following we examine two examples, by showing the outputs
produced by the algorithm (1) implemented with Maple V: in the first one we compute the minimum $p'$ and the maximum $p''$; in the second one we compute the values $P_c$ and $p^\circ$. The second example has been also studied in [11].

**Example 1.** Given the family $\mathcal{F} = \{B|A, BC|A\}$ and the precise probability
assessment $\mathcal{P} = (1/4, 1/3)$ on $\mathcal{F}$, we consider the extension of $\mathcal{P}$ to the conditional event $C|AB$. In our case the constituents, $C_0, C_1, \ldots, C_4$, are respectively

$$A^c, AB^c, ABC, ABC^c, AB^c C^c.$$ 

Then, we have

$$
\Phi_A = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4, \quad \Phi_{AB} = \lambda_2 + \lambda_3.
$$ 

The associated system

\[
\begin{align*}
\lambda_2 &= p(\lambda_2 + \lambda_3) \\
\lambda_2 &= \frac{1}{4}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\
\lambda_2 + \lambda_3 &= \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\
\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 &= 1, \lambda_r \geq 0
\end{align*}
\]

with the position $p = 0$ is infeasible.
Moreover, $\max \Phi_{AB}$ is positive and, based on the Algorithm 1, the following linear programming problem must be solved.

Compute

$$\min \lambda_2$$

Subject to the constraints:

$$\begin{cases} 
\lambda_2 = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\
\lambda_2 + \lambda_3 = \frac{1}{4}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\
\lambda_2 + \lambda_3 = 1, \lambda_r \geq 0
\end{cases}$$

The minimum $p'$ is 1. Concerning the computation of $p''$, we observe that the system (5) with the position $p = 1$ is feasible and the maximum of the function $\Phi_{AB}$ is positive, then the assessment ($\mathcal{P}, \lambda$) on $\{B|A, BC|A, C|AB\}$ is coherent, therefore $p'' = 1$.

**Remark 2.** In [11] the restrictive assumption that the probabilities of the conditioning events are positive is made. In the next example, since we don’t make this assumption, a result different from that given in [11] is obtained, confirming that the de Finetti’s approach is more general.

**Example 2.** Given the family $\mathcal{F} = \{B|A, A|B, C|B, B|C\}$ and the imprecise probability assessment $\mathcal{A} = ([1/3, 1/2], [0, 1/2], [1, 1], [1/3, 1/2])$ on $\mathcal{F}$, we consider the extension of $\mathcal{A}$ to the conditional event $C|A$. In this case the constituents, $C_0, C_1, \ldots, C_7$, are respectively

$A^c B^c C^c, AB^c C, ABC, ABC^c, A^c BC, A^c BC^c, AB^c C^c, A^c B^c C$.

Moreover it is

$$\Phi_A = \lambda_1 + \lambda_2 = \lambda_3 + \lambda_6, \quad \Phi_B = \lambda_2 \lambda_3 + \lambda_4 + \lambda_5$$

$$\Phi_C = \lambda_1 + \lambda_2 + \lambda_4 + \lambda_7.$$ 

The associated system

$$\begin{cases} 
\lambda_1 + \lambda_2 = p(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_6), \\
\lambda_2 + \lambda_3 = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_6), \\
\lambda_2 + \lambda_3 \geq \frac{1}{5}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_6), \\
\lambda_2 + \lambda_3 \leq \frac{1}{2}(\lambda_2 + \lambda_3 + \lambda_4 + \lambda_5), \\
\lambda_2 + \lambda_4 = \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5, \\
\lambda_2 + \lambda_4 \geq \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_4 + \lambda_7), \\
\lambda_2 + \lambda_4 \leq \frac{1}{2}(\lambda_1 + \lambda_2 + \lambda_4 + \lambda_7), \\
\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 = 1, \lambda_r \geq 0
\end{cases}$$
with the position $p = 0$ is feasible, so that we check the coherence of the assessment $(\mathcal{P}, 0)$ on the family $\{B|A, A|B, C|B, B|C, C|A\}$. The set $I_0 = J \cup \{5\}$ is $\{1, 5\}$. Then we consider the assessment $([1/3, 1/2])$ on the subfamily $\mathcal{F}_J = \{B|A\}$ and the conditional event $C|A$. The constituents, $C_0, C_1, \ldots, C_4$, generated by $B|A, C|A$, are respectively

$$A^c, AB^cC, ABC, ABC^c, AB^cC^c.$$ 

then we have

$$\Phi_A = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4.$$ 

The associated system

$$\begin{array}{l}
\lambda_1 + \lambda_2 = p(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4), \\
\lambda_2 + \lambda_3 \geq \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4), \\
\lambda_2 + \lambda_3 \leq \frac{2}{3}(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4), \\
\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1, \quad \lambda_r \geq 0
\end{array}$$

with the position $p = 0$ is feasible. Obviously the maximum of the function $\Phi_A$ is positive, so $p_\circ = 0$. concerning the computation of $p^\circ$, the system (7), with the position $p = 1$, is feasible and the maximum of the function $\Phi_A$ is positive, so $p^\circ = 1$.

**Remark 3.** We observe that, concerning the example above, in [11] the following values have been obtained: $p_0 = \frac{1}{3}$, $p^\circ = 1$.

**REFERENCES**


Dipartimento di Matematica e Informatica,
Viale A. Doria 6,
95125 Catania (ITALY)