IRIS 1.0

(Interactive Robustness analysis and parameters' Inference for multicriteria Sorting – Version 1.0)

Methodology by:

Luís Dias(1,2), Vincent Mousseau(3), José Figueira(1,2,3), João Clímaco(1,2), and Carlos Gomes Silva(1,4)

IRIS software design and development by:

Luís Dias(1,2), Vincent Mousseau(3), and Carlos Gomes Silva(1,4)

(1) INESC Coimbra, Rua Antero de Quental, 199, 3000-033 Coimbra, Portugal
(2) Faculdade de Economia, Universidade de Coimbra, Av. Dias da Silva 165, 3004-512 Coimbra, Portugal
(3) LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France
(4) Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, 2401-951 Leiria, Portugal

[See also the Homepage of IRIS]

IRIS (Interactive Robustness analysis and parameters' Inference for multicriteria Sorting) is a Decision Support Software designed to address the problem of sorting a set of actions (alternatives, projects, candidates, etc.) into predefined ordered categories, according to their evaluations (performances) on multiple criteria. For instance, it may be used to sort funding requests according to merit categories (e.g. “Very good”, “Good”, “Fair”, “Not eligible”), or to sort loan applicants into categories (e.g. “Accept”, “Require more collateral”, “Reject”), or to sort employees in a company into categories that define incentive packages, etc.
IRIS implements the methodology presented in Dias et al. (2002), using a pessimistic concordance-only variant of the ELECTRE TRI method. Rather than demanding precise values for the ELECTRE TRI parameters, IRIS allows to enter constraints on these values, namely assignment examples that it tries to restore. It adds a module to identify the source of inconsistency among the constraints when it is not possible to respect all of them at the same time, according to a method described in Mousseau et al. (2002). On the other hand, if the constraints are compatible with multiple assignments for the actions, IRIS allows drawing robust conclusions by indicating the range of assignments (for each action) that do not contradict any constraint.

The main characteristics of IRIS are the following:

- IRIS implements a concordance-only variant of the pessimistic ELECTRE TRI.
- IRIS accepts imprecision concerning the criteria weights and the cutting level. The users may indicate intervals for each of these parameters, as well as linear constraints on the weights, rather than being forced to indicate precise values for all these parameters. Furthermore, the constraints may be defined indirectly, as indicated in the next item.
- IRIS accepts assignment examples, where the users indicate minimum and maximum categories for some of the actions, according to their holistic judgement (e.g. “action a-1 is a typical element of C3”, or “action a-2 should be placed in category C3 or higher”, or “I hesitate: action a-2 should be placed in category C3 or C4”). These assignment examples are translated into constraints on the parameter values, meaning that the assignments of ELECTRE TRI should restore these examples.
- When the constraints are inconsistent, IRIS infers a combination of parameter values that least violates the constraints, by minimizing the maximum deviation. Then, it shows the sorting that corresponds to these parameter values (see example in Fig. 1). Furthermore, a module becomes available to determine the alternative subsets of constraints that must be removed to restore the consistency (see example in Fig. 2).
- When the constraints are consistent, IRIS infers a "central" combination of parameter values by minimizing the maximum slack. For each action, it depicts the category corresponding to that combination, as well as the range of categories where the action might be assigned without violating

any constraint (robustness analysis). For each category in the range IRIS may also determine a combination of parameter values that assigns the action to that category (see example in Fig. 3).

- Moreover, when the constraints are consistent, IRIS may compute some indicators concerning the precision of the inputs (by estimating the volume of the polyhedron of all feasible combinations of parameter values) and the precision of the outputs (by indicating the geometric mean of the number of possible assignments per action). See example in Fig. 4.
- These features allow decision makers to build sorting models in a progressive and interactive manner, where the output at a given iteration is used to guide the revision of the input for the following iteration. The general idea is to start with few constraints of the parameter values, adding more inequalities as a product of an interactive learning process about the problem and the method. This process should aim at progressively reducing the set of accepted combinations of parameter values, until the end users (decision makers, problem owners) are satisfied with the results' precision, and yet comfortable with and confident about the constraints introduced.

The final outputs of the procedure are:

- a set of constraints and assignment examples defining a set of acceptable combinations of parameter values;
- an inferred combination of parameter values defining a model in a precise manner;
- a precise assignment or range of assignments for each action in A that is robust with respect to the constraints inserted.

However, the most important outcome may be that the end users will increase the insight on their view of the problem, learn about their preferences, and will possibly modify their opinions.

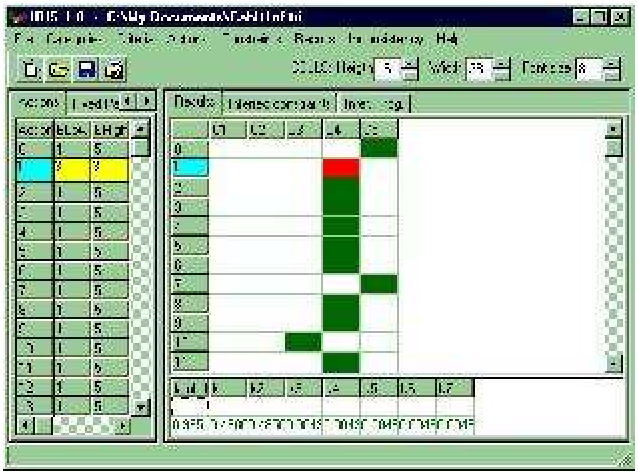

Figure 1. The proposed sorting does not restore the assignment example that a1 belongs to C3 due to inconsistent constraints. It corresponds to the parameter values indicated on the right bottom of the screen.

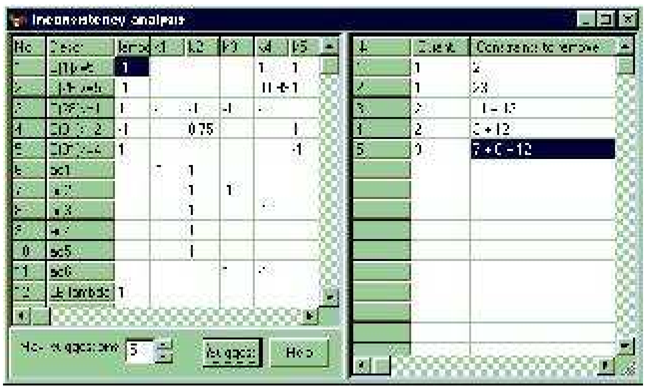

Figure 2. Given an inconsistent system of constraints (on the left), IRIS suggests five alternative ways to restore the consistency by removing constraints. The first suggestion is to remove constraint no. 2; the fifth suggestion is to remove constraints no. 7, 8, and 12.

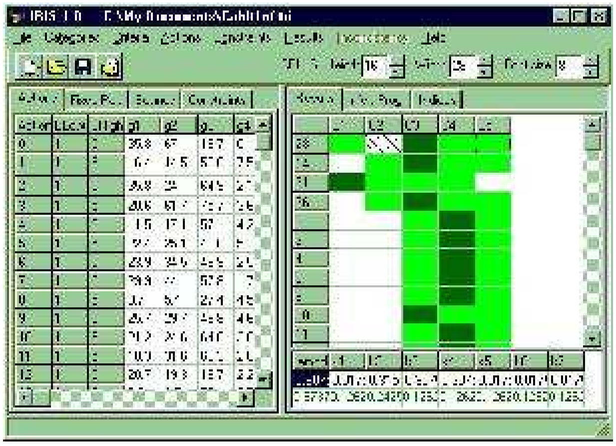

Figure 3. There is a range of categories where each action may be assigned to without violating any constraint (e.g. a robust conclusion is that a2 is not worse than C3). The proposed assignment (darker cell) corresponds to the inferred parameter values shown in the last row of the grid on the right. The parameter values shown in the penultimate line of that grid lead to the assignment of a28 to C5, corresponding to the selected cell. If the user chooses another cell these values will change. IRIS also shows that a28 cannot be assigned to C2, regardless of the parameter values that are chosen.

Figure 4. (Left:) the constraints define a 7-dimension polytope of very small volume; from the combinations of parameter values that satisfy the bounds, about 14.3% also respect the remaining constraints. (Right:) the geometric mean of the number of categories where each action may be assigned (respecting all the constraints) is now 1.357, which is less 47.7% relatively to the previous iteration.

MORE INFORMATION

INESC Coimbra
C/O Luís Dias
Rua Antero de Quental, 199, 3000-033 Coimbra, PORTUGAL
Fax: +351 239 824692, e-mail: LDias@inescc.pt
http://www4.fe.uc.pt/lmcdias/iris.htm

REFERENCES

Mousseau, V. J. Figueira, L. Dias, C. Gomes da Silva, J. Clímaco (2002), "Resolving inconsistencies among constraints on the parameters of an MCDA model", to appear in the European Journal of Operational Research.
Dias, L., V. Mousseau, J. Figueira, J. Clímaco (2002), "An Aggregation/Disaggregation Approach to Obtain Robust Conclusions with ELECTRE TRI", European Journal of Operational Research, vol 138, 332-348.

[Back to EWG-MCDA Software page]