Jana Kalčevová

Petr Fiala

IZAR – THE CONCEPT OF UNIVERSAL MULTICRITERIA DECISION SUPPORT SYSTEM

Abstract

Many real decision making problems are evaluated by multiple criteria. To apply appropriate multicriteria methods it is necessary to have software support. The paper presents the universal multicriteria decision support system IZAR. The main component of the system IZAR is an expert system helping the user with choosing the most suitable method for available information about the problem. IZAR not only suggests the most suitable method, but it also applies it immediately to the problem being solved. By means of properly formulated questions the expert system selects the right procedure for solving the problem, taking into account its peculiarities regarding entries and additional information which can be assigned to the system by the user. An appropriate classification of multicriteria models and methods is needed because of universality of the system. The system solves discrete and continuous problems. Basic models for multiattribute evaluation and multiobjective optimization problems are included. Methods for multiattribute evaluation problems are classified by means of types of preference information and by calculation principles. Preference information is given as aspiration levels, ordinal information, and cardinal information. Basic calculation principles are utility maximization, minimization of distance from the ideal alternative, and evaluation by preference relation. Methods for multiobjective optimization problems are classified by means of types of user information, as a priori information, a posteriori information, and progressive information. The system is available on web pages for all interested users and it can be also distributed on CD for users without Internet connection.

Keywords

Multiple criteria, multiattribute evaluation, multiobjective optimization, models, methods, decision support system.

INTRODUCTION

IZAR[1] is a non-commercial software package for students of decision theory. The final version of IZAR should be the universal system for the solution of discrete and continuous single objective and also multiobjective optimization problems. The first part of this software package is common background with an expert system of IZAR. This part has already been implemented and at present the authors focus on the second part of IZAR: continuous problems. The idea is concentrated on linear programming models, but in the future we will probably expand IZAR to include some non-linear methods. Also the third part – discrete problems – is now in the development phase.

The entire IZAR system is implemented in Smalltalk/X, a dialect Smalltalk-80. A Smalltalk/X virtual machine and runtime environment is available for both MS Windows and Linux, and also for FreeBSD.

The IZAR system is divided into three main parts:

– a core mathematical library consisting of a set of basic mathematical types and operations (matrixes and matrix operations such as matrix multiplication),
– a user interface for convenient communication with the user,
– a set of implemented methods.

All methods are implemented as external independent programme units that can be dynamically loaded to or unloaded from the running system. The methods are implemented in a slightly modified version of Pascal [4] which is interpreted by a specialized build-in interpreter. Pascal is also used as a scripting language, so the user can work with IZAR system non-interactively.

This design gives an opportunity to study and explore the IZAR system, especially by means of the implemented methods, by the user. This is one of the most important features of the system.

1. SINGLE OBJECTIVE CONTINUOUS PROBLEMS

The simplest problem that can be solved by IZAR is a continuous problem with the set of feasible solutions given by linear constraints in the form of inequalities and with only one objective function. The problem can be written as:

[1] This project is supported by grant agency of Czech Republic GA ČR 402/07/0166.

Where *f(x)* and *g(x)* are linear functions, *n* is a number of variables and *m* is a number of constraints. Constraints can be written in the form of equations or inequalities. All constraints with non-negativity conditions comprise the feasible set.

Single objective problems can be solved by the simplex method. This method is implemented in IZAR and is used for searching for the optimum of minimum or maximum objective functions on the feasible set. In the case of constraints of the type "less than or equal to" (with non-negative right hand sides) one-phase simplex method is used. In the case of a feasible set with at least one constraint of the type "equal" or "greater than or equal to" twophase simplex method is used. Algorithms for both methods are described in [3] and are implemented in the modified simplex method form. For now, a multiplicative version of the simplex algorithm is used, but this can be changed to achieve better performance, since the simplex algorithm is a basic building block for most methods implemented.

IZAR solves the problem by this method; the result window shows results and simplex tables. All non-integer coefficients are presented in fraction form.

2. MULTIOBJECTIVE CONTINUOUS PROBLEMS

Multiobjective continuous problems differ from the single objective ones as regards the number of objective functions [1]. These functions can be minimal or maximal and the model can be written in the form:
Where *f(x)* and *g(x)* are linear functions, *n* is the number of variables, *m* is the number of constraints and *v* is the number of objective functions. As previously, constraints can be written in form of equation or inequality.

2.1. Data sources

We want IZAR to be a very friendly software package and consequently we focus on easy data input. The data can be loaded from a data file in CPLEX format (for single criteria problems) or in augmented CPLEX format for multicriteria problems (the format has to correspond to the model described in the second and third parts of this paper). Data can be also entered manually. The window for data dimension has four parts. The first window shows the name of the problem (named by default "New problem"). Figure 1 displays the problem of furniture factory that produces tables and chairs; time, wood, and chrome are needed for production. The factory maximizes profit and minimizes total costs at the same time. The second part of the window deals with variables. The user can change the number of variables and their names. The default names of variables are x1, x2, etc. The third part makes it possible to type in the number of constraints and their names (named by default Constraint 1, Constraint 2, etc.). The last part deals with objectives functions. The user can change their number and names. Note that the numbers of variables, constraints, and objective functions are unbounded. They are limited only by the computational power of CPU and the available memory.

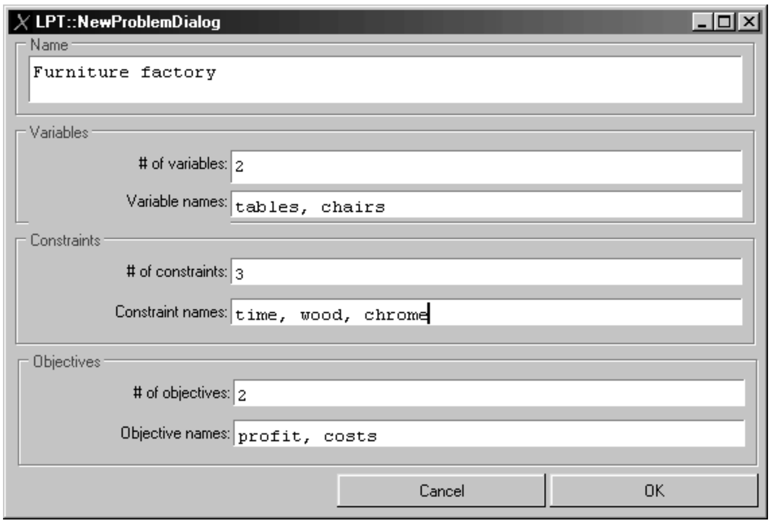

Fig. 1. Problem dimension

Next window (Figure 2) shows the subwindow for data submission. The upper part of the window is designed to input objective functions. The user should enter the prices and types of the extremes of objective functions. The next subwindow is intended for structural coefficients and right hand sides of constraints described previously. Here the user can change the sign of inequalities. Below is a place for method selection. A menu lists the methods; there is also a possibility of selecting an expert system for automatic selection of method. When the method is selected, IZAR finds a compromise solution and displays the results.

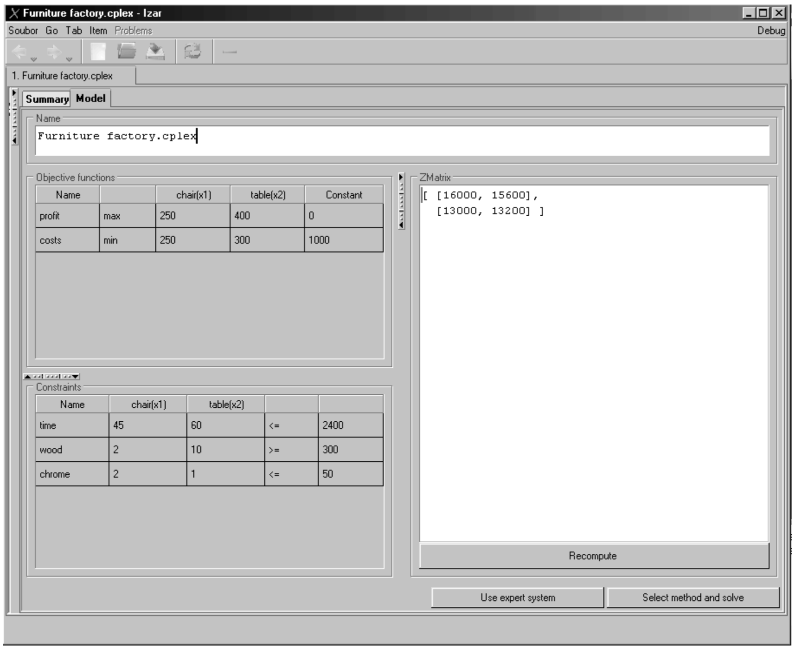

Fig. 2. Selection of functions and method (or expert system)

2.2. The expert system of IZAR for continuous problems

An expert system can be very useful. We propose a similar expert system for multiobjective optimization problems. The main menu for method selection offers eleven methods for these problems:

1) maximal utility method,
2) minimal distance from the ideal solution method,
3) lexicographic method,
4) goal programming,
5) maximal probability method,
6) minimal component method,
7) multicriteria simplex method,
8) GDF method,
9) Zionts-Wallenius method,
10) STEM,
11) Steuer method.

The main component of the IZAR system is an expert system helping the user with choosing the most suitable method for the available information about the problem. By means of properly specified questions the expert system selects the right procedure for solving the problem, taking into account its peculiarities regarding entries and additional information, which user can assign to the system. The choice of the suitable method is based on the following classifications and questions.

Methods are classified by means of setting user information:

– a priori information (methods 1-6),
– a posteriori information (method 7),
– progressive information (interactive methods 8-11).

The user chooses the way of specification of the importance of each criterion:

– weights,
– order of the criteria,
– goal (ideal) values.

Weights can be assessed:

– directly,
– by ordinal ordering of the criteria,
– by cardinal evaluation of criteria.

The information, which can affect the choice of the method is a calculation principle:

– maximization of the utility,
– minimization of the distance from goal (ideal) values.

The form of substitution information:

– explicitly expressed value of substitution (rates of substitution),
– implicitly expressed value of substitution.

IZAR not only suggests the most suitable method, but also applies it immediately to the problem being solved.

2.3. Compromise solution

Methods for multiobjective optimization problems are classified using types of user information. A priori information is given before the start of problem solving, a posteriori information is given after the computation, and progressive information is given during the calculation process.

After the method has been selected, the calculation process is started and additional information is required. In the case of maximal utility method and minimal component method weights are needed, while goal values have to be given for goal programming. Minimal distance from the ideal solution method works with both (weights and ideal values), but ideal value is calculated automatically. The order of the criteria is required for the lexicographic method, while for the maximal probability method and the multicriteria simplex method no information is needed. Progressive information is given in the case of all four interactive methods: GDF, Zionts-Wallenius, STEM, and Steuer methods.

Each method provides a compromise solution. The user can solve the same problem by another method or by the same method, but with different additional information. The user can also change the model or exit IZAR. The results of all methods are saved in a table for easy comparison (Figure 3).

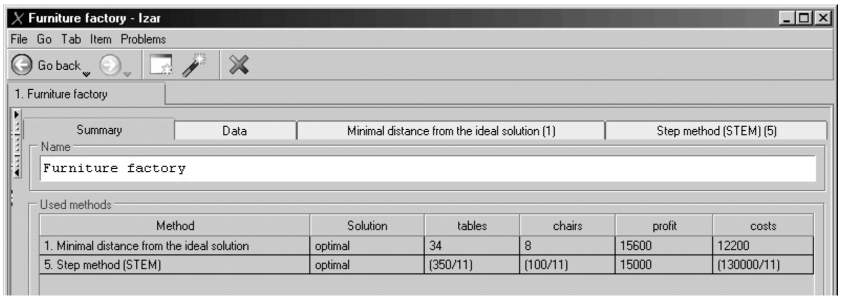

Fig. 3. Summary of results for comparison

3. MULTIATTRIBUTE EVALUATION (DISCRETE PROBLEMS)

The discrete problem with several criteria is given by the list of p alternatives, the list of k criteria, and also by the evaluation of each alternative by each criterion. Let a_i be the i-th alternative ($i = 1,2,\ldots,p$) and f_j be the j-th criterion ($j = 1,2,\ldots,k$). Then the evaluation of the alternative a_i by the criterion f_j is written as $f_j(a_i)$ or shortly as y_{ij}. Each criterion f_j can be minimized or maximized. The user searches for the most appropriate alternative from the list of alternatives; evaluation is based on the values y_{ij}. These values can be represented by the criteria matrix Y where each element y_{ij} denotes the evaluation of the alternative a_i by the criterion f_j for all $i = 1,2,\ldots,p$ and $j = 1,2,\ldots,k$.

3.1. Data sources

The data for discrete problems can be also loaded from a data file which is in a special IZAR format: *.idm (Izar Discrete Model). Data can be also entered manually. The window for data dimension (see Figure 4) displays the name of the problem, the number of criteria and their names, and the number and names of alternatives. All these information can be changed in the same way as in the continuous case. Note that the numbers of criteria and of alternatives are unbounded. They are limited only by computational power of CPU and the available memory.

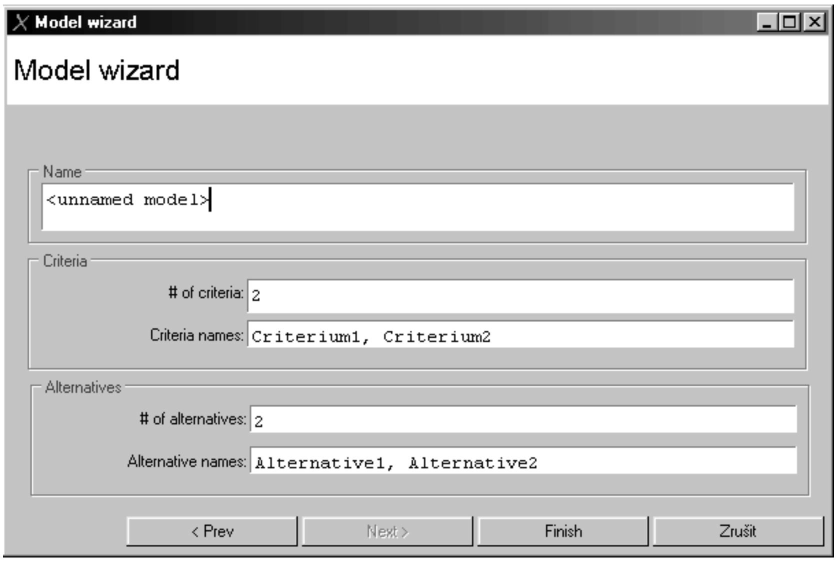

Fig. 4. Problem dimension for discrete problems

The next window (Figure 5) shows the subwindow for data input. The user should enter the evaluation of each alternative by each criterion. Below is a box for method selection. The methods are listed in the menu together with the possibility to choose an expert system for the automatic selection of method. Next, IZAR finds a compromise alternative and displays the results.

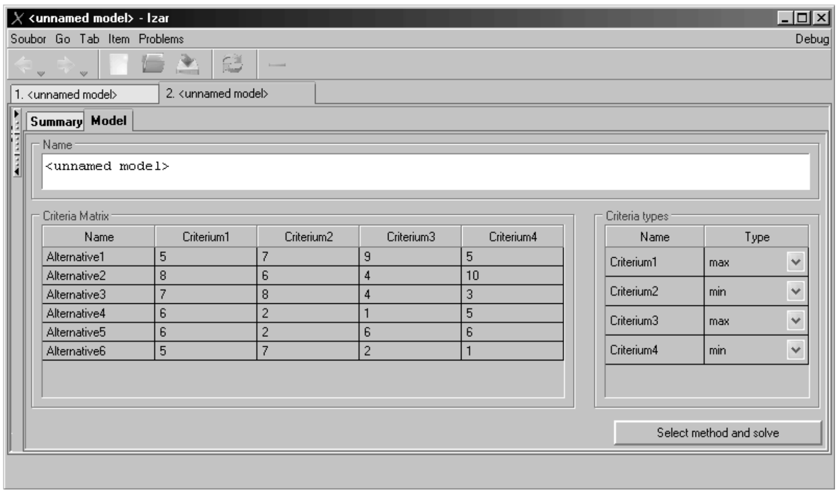

Fig. 5. Evaluation and method (or expert system) selection

3.2. Expert system of IZAR for discrete problems

An expert system for discrete problems is similar to that for continuous problems and was proposed in [2]. The choice of the suitable method is based on the following classifications and questions.

Methods are classified by the goal of multicriteria analysis:

- selection of the set of "good" alternatives,
- selection of the best alternative,
- ranking of all alternatives.

The user chooses the manner of specification of the importance of each criterion:

- weights,
- order of the criteria,
- aspiration levels.

Weights can be assessed:

- directly,
- by ordinal ordering of the criteria,
- by cardinal evaluation of the criteria.

The information which can be assigned to the alternatives:

- criteria matrix values,
- the order of the alternatives according to a single criterion,
- pairwise preferences of the alternatives according to a single criterion.

The pairwise preferences are of two types:

- the ordinal preferences of the alternatives,
- the cardinal preferences of the alternatives.

The information which can affect the choice of the method is a calculation principle:

1. Maximization of the utility – this principle is based on measuring the utility which is related to the alternatives regarding each of criterion; for this purpose it is necessary to carry out the transformation of the original data by:
   - linear normalization – simple scalarization of the multicriteria task where the transformed data resembles normalized values,
   - utility function – piecewise linear utility function can be constructed.
2. Minimization of the distance from the ideal criteria values – this principle is based on the choice of an alternative with the lowest distance from the ideal alternative according to various metrics.
3. Preference relation – relation between two alternatives, taking into account values reached in corresponding criteria.

IZAR not only suggests the most suitable method for discrete problems, but it also applies it immediately to the problem being solved. For users advanced in multicriteria analysis, to the extent that they do not need an expert system, there exists a possibility of the direct choice of the solving method.

7.3. Compromise alternative

After the method has been chosen, the calculation process is started and additional information is required. Aspiration levels are needed for Disjunctive method, Conjunctive method, and PRIAM. The Permutation, Lexicographic, and ORESTE methods need the order of criteria (or alternatives, respectively). WSA, UFA, and AHP are based on the maximization of the utility and they need cardinal information of criteria as well as TOPSIS (minimizing distance from the ideal values). Finally, AGREPREF, ELECTRE, PROMETHEE, and MAPPAC are based on the preference relations.

The Disjunctive and Conjunctive methods provide a set of "good" alternatives, while PRIAM displays the best alternative. The Permutation method, Lexicographic method, ORESTE, WSA, UFA, AHP, TOPSIS, AGREPREF, ELECTRE III, PROMETHEE I, II, and MAPPAC provide complete ranking of the set of all alternatives and they can be used for the choice of the best alternative. ELECTRE I provides a list of efficient and inefficient alternatives.

In the case of methods with aspiration levels, IZAR needs a criteria matrix and aspiration levels (repeatedly). Methods with ordinal information use a criteria matrix or order of alternatives according to each criterion and ordinal information of criteria. Methods with cardinal information need a criteria matrix and weights of criteria.

Each method provides a compromise alternative or complete ranking of the set of all alternatives. Obviously, the user can solve the same problem by another method or by the same method, but with different additional information (such as weights, aspiration levels, etc.). The user can also change the model or exit IZAR. The results of all methods are saved in a table for easy comparison, as in the case of continuous problems.

CONCLUSION

The IZAR system is in a development stage. The continuous problems are implemented and in the near future this part of the IZAR system will be tested at Department of Econometrics in the Economic University in Prague, as part of the course in Decision Theory.

The discrete problems are prepared for implementation and now six methods (Lexicographic, TOPSIS, WSA, AGREPREF, ELECTRE I, and ELECTRE III) are included in IZAR and tested. Each discrete problem is represented by a list of alternatives, a list of criteria, and a matrix with evaluation of alternatives by criteria.

The system was created for modeling and analyzing multiobjective combinatorial auctions (see Fiala's paper in this volume), but its utilization is more general: it can be used to solve many types of multicriteria decision making problems.

The IZAR system has several important advantages. The first is the fact that IZAR is a non-commercial software package for students and its design gives the user a possibility to study and extend the IZAR system. A user with elementary knowledge of the Pascal language can read the programme code for each method and study the method in detail, and a experienced one in Pascal programming can implement his own methods or improve the existing components of the IZAR system.

The second advantage is its unlimited number precision. Most of currently available software packages represent numbers (i.e. value of variables, coefficients, constants, etc.) as floating-point numbers because floating-point arithmetic is quite fast (the majority of operations can be done in hardware). Although floating-point arithmetic is fast and quite easy, it introduces almost unsolvable problems with correctness of results because it is subject to rounding errors. For that reason IZAR represents all numbers as fractions, i.e. as pairs of two (unlimited) integers – nominator and denominator, and so it is possible to represent any rational number within the IZAR system without any precision lost. The usage of fractions is transparent to the user. The main disadvantage of the fraction number representation approach is its computational and memory complexity – the solution of large problems is quite slow. But the CPUs become faster and faster and the authors believe that it is much more important to obtain correct results after a (possibly) long time than to obtain a (possibly) imperfect result in a short time. However, the problem of solving large sets of large problems is solvable. The IZAR system provides non-interactive (batch) mode, in which no user interaction is needed. This is one solution of this problem. On the other hand, since all methods are written in the Pascal language, it is possible to compile one specialized programme using an ordinary Pascal compiler (for example) that produces highly optimized code for target platform using floating-point arithmetic. The resulting specialized programme will be much faster, but may produce incorrect results (because of rounding errors in floating-point arithmetic).

The third advantage is the unbounded dimension of problems that can be solved. Programmes for solving linear programming models are usually limited by the number of variables and the number of constraints. This programme has no limits. The user can solve problems with an arbitrary number of variables, constraint, and objective functions, limited by the computer capacity only.

The fourth advantage is the number of implemented methods. IZAR knows eleven methods for multiobjective optimization problems (they are named in Section 3.2). As it is mentioned above, the user can extend this system by introduction of additional methods. In the final version, methods for multiattribute evaluation problems will be implemented. If necessary, the system can be extended, for example, by data envelopment analysis (DEA models) or by other methods.

All methods are included in the expert system which chooses the most suitable method for the given problem. By asking specific questions the expert system selects the right procedure for the problem being solved and so the system can be used by users unfamiliar with decision theory.

Last, but not least, an advantage of the IZAR system is the existence of this system. Many systems for linear programming problems exist, but they are all focused on one objective function. The IZAR system is the first system for multiobjective optimization problems that is accessible to students of Czech universities. And not only to them: The final version placed on web sites will be open to all people interested in decision problems. At the moment, the current version of IZAR is available on http://moon.felk.cvut.cz/~vranyj1/wiki/doku.php?id=izar:download.

REFERENCES

1. Fiala P.: Modely a metody rozhodování. (Models and Methods of Decision--Making). Oeconomica, Praha 2003.
2. Fiala P., Fábry J.: SYMCA-System for Multicriteria Analysis. “Central European Journal for Operations Research and Economic” 1993, vol. 2, No. 2, pp. 177-184.
3. Lagová M., Jablonský J.: Lineární modely. (Linear Models). VŠE, Praha 1999.
4. Wirth N., Kathy J.: PASCAL – User Manual and Report. Springer-Verlag, 1974.
5. Jablonský J.: Operační výzkum – kvantitativní modely pro ekonomické rozhodování. Professional Publishing, 2002.
6. Kalčevová J., Fiala P.: IZAR – Multicriteria Decision Support System. In: Mathematical Methods in Economics 2006, Pilsen 2006, pp. 277-282.