

NETPRE - A DECISION SUPPORT SYSTEM FOR ANALYZING PREFERENCES IN A NETWORK SETTING

Raimo P. Hämmäläinen and Risto Karjalainen
Systems Analysis Laboratory
Helsinki University of Technology
Otakaari 1 M, 02150 Espoo, Finland

ABSTRACT

NETPRE is a software product running on personal computers that helps a decision maker to analyze problems that are modelled as networks. The system allows the decision maker to specify feedback mechanisms between elements of the problem. One of the aims has been to make the interaction between the user and the system as easy as possible. Interactive modelling is achieved through menu-based procedures, and the user is guided through the process of assessing preferences between the elements of the decision. Future directions in the development of interactive decision support systems are also outlined.

AHP NETWORK MODEL - THE ANALYTIC NETWORK PROCESS

The theory of the Analytic Hierarchy Process, AHP (Saaty 1980), is now well established. A hierarchical model for a decision may be built following the method. Recently increasing interest has been focused in problems modelled as networks (Saaty 1980, 1987, Hämmäläinen et al. 1986b). In the network approach, called the Analytic Network Process (ANP), the capacity to express dependencies is increased. The consequence is, however, that the analysis and interpretation of its results become problematic. The practical value of the theory remains to be seen through experimentations. In this paper we shall describe a decision support system (DSS) utilizing this approach. The software product, NETPRE, is implemented for IBM compatible personal computers. It is a completely revised version of an earlier system of ours, AME, that was used e.g. for studying dependencies between the short and long term goals when evaluating the alternative energy policies in Finland (Hämmäläinen et al. 1986a, 1986b).

ANP has the potential to enrich decision maker's (DM's) insight into the problem structure. The problem is decomposed into segments, called elements. In addition to a conventional hierarchy representation, NETPRE provides the DM with means to build a model with arbitrary dependencies between the elements. Thus e.g. feedback loops may be identified and implemented in the model.

The actual model for a decision consists of several elements, which may in turn have subelements. This terminological convention differs from that used by Saaty, who divides components further into elements. Each element may depend on other elements. The set of elements forms a directed graph, where a unit weight is distributed from each node, i.e. element, along the arcs to the connected elements. This is visualized in Figure 1; the weight flows in the direction of arrows. The DM assesses weights to the connected elements, with respect to each element. These weights express

his or her preferences. In a simple hierarchy, weights descend from the top, i.e. the goal of the decision, to the bottom, usually consisting of the decision alternatives. The bottom level weights correspond directly to the DM's strength of preference between the elements. In a network, however, there is neither top nor bottom, and final weights are obtained only after the cyclic interaction between the elements has taken place. The resulting weights correspond to the importance of the elements with respect to the whole system (Saaty 1980).

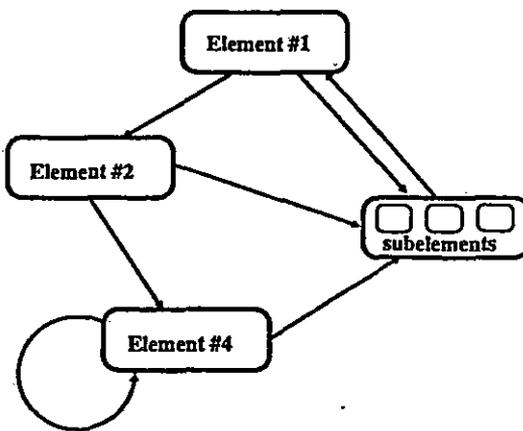


Figure 1. A network.

In ANP, the weights are assessed through pairwise comparisons, as in AHP. The questions are like "With respect to the element GOAL, which is more important, element CRITERION #1 or element CRITERION #2 ?". Thereafter, the DM is asked to assess the strength of preference on the scale from 1 (equally important) to 9 (extremely more important). The matrix of pairwise comparisons is constructed, and the eigenvector corresponding to the largest eigenvalue gives the desired initial weights.

The sum of weights emerging from one element is normalized to unity. In this manner the matrix of all initial weights is column stochastic, and its limiting power - if it exists - gives the final weights after infinitely many periods of interaction. The theory (Saaty et al. 1986, Saaty 1987) parallels that of Markov chains, and the limiting weights correspond to the stationary distribution. However, one must not interpret the initial weights as transition probabilities.

HOW TO STRUCTURE A PROBLEM WITH NETPRE

As explained, NETPRE offers a way to decompose a complex decision. Let us suppose that the DM is faced with a decision to rank certain alternatives, or to choose among alternative ways to proceed in a specified situation. First, the DM determines the relevant elements of the decision. These may correspond e.g. to the main criteria used to evaluate the consequences of the decision alternatives. Second, the defined elements are further divided into subelements capturing the essence of the problem to any required level of details. The two levels of decomposition retain the clarity of the model, while still offering tools for a comprehensive analysis.

Model definition

In practise, the DM chooses from a menu an option to create a new model. Presented another menu, he or she chooses to add an element. In this phase the DM is asked to enter the element name, and new elements are included until the DM terminates the interaction simply by not giving any further element names.

Having created the elements, the DM chooses an option that leads to a subelement menu. This menu contains the name of each element, and the DM chooses the element which is to be decomposed. A new menu appears, and a choice to add subelements can be made. Subelements are defined in exactly the same manner as the elements themselves. During all the steps through the menus, all the preceding menus and choices remain visible, as shown in Figure 2, so the DM knows what is the current task and how he or she got there.

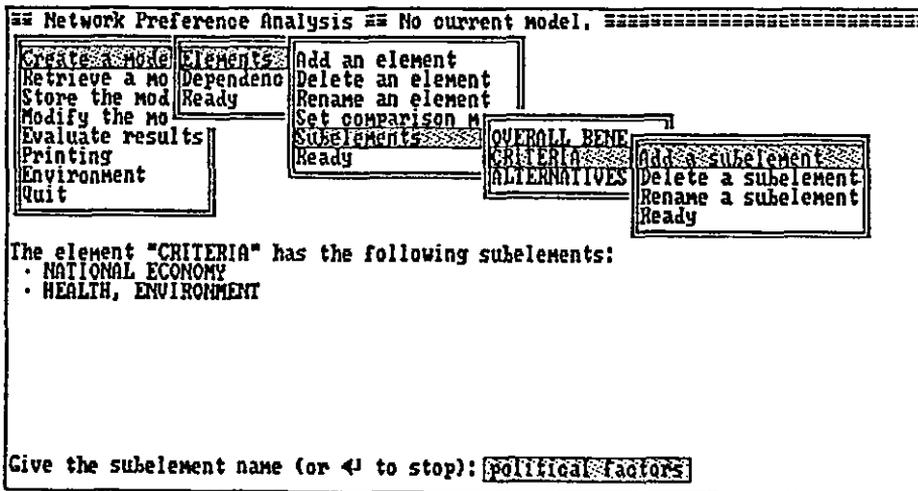


Figure 2. Menu structure for subelement definition.

Once the elements and subelements have been defined, a menu for specifying dependencies is presented. The option to build a hierarchy provides a simple way to create a conventional hierarchical model. This is done by pointing out all the elements belonging to a certain hierarchy level. The DM is guided through the levels by the system.

The choice to create a network causes a menu of elements to be presented. Here, the DM first chooses an element, and from another menu points out the elements on which the chosen one depends. An element may depend on several elements, including the element itself. The dependencies may also be defined by directly modifying the dependence matrix. The matrix is presented as a two-dimensional array with entries as arrows or points, depending whether respective arcs in the network exist or not. Menu choices are provided to fill, clear or toggle rows, columns or single entries of the matrix, as well as the whole matrix.

When the model is complete, the DM may wish to modify it. NETPRE provides means to add, delete or rename elements or subelements and change the dependencies. All these functions are available also during the creation of the model.

Weights assessment

The next step in the preference assessment is weight elicitation (Saaty 1980). The DM is given the option to assess initial weights in the entire model. This leads to a session where the DM is guided through all the necessary pairwise comparisons between elements and subelements. For each element, elements on which it depends are compared in a pairwise manner. First, a menu of possible weighting operations is presented, as shown in Figure 3. The option of asking the weights brings another menu of each two-element-pair of the connected elements. When the DM has chosen the more important one with respect to the element in turn, a menu of levels of preference strength appears. The menu includes the levels from 1 to 9, together with descriptions of the meaning of the numbers. When the DM has answered to each question, options for modifying assessments are still available. Consistency of comparisons is automatically calculated as the so-called C.R.-ratio. In case of inconsistent comparisons, the DM is given the choice of either repeating the procedure or leaving the results as they are.

The procedure described above is repeated for subelements of respective elements. In any phase, the DM may quit weighting; he or she is not obliged to complete the possibly time-consuming operation at one time. Later, the DM may reassess the initial weights in the model. Then, he or she is given an option of having only the missing comparisons requested. Also, it is possible to directly move to a specific element, and assess weights either for elements or subelements. Initial weights can also be specified directly.

Evaluation of limiting weights in the network may be done after all the weights have been elicited. If some of the comparisons are missing, the location where the first such incident takes place is reported to the DM. The limiting network weights are presented in a tabular format. The DM may choose to directly examine the supermatrix and its desired powers, but normally there is no need for that. The system can detect a hierarchical model structure, and in such a case presents the re-

sults as weights with respect to the goal of the model. Further, if there are several top level goals, the DM can choose the one with respect to which he or she wishes to see the weights.

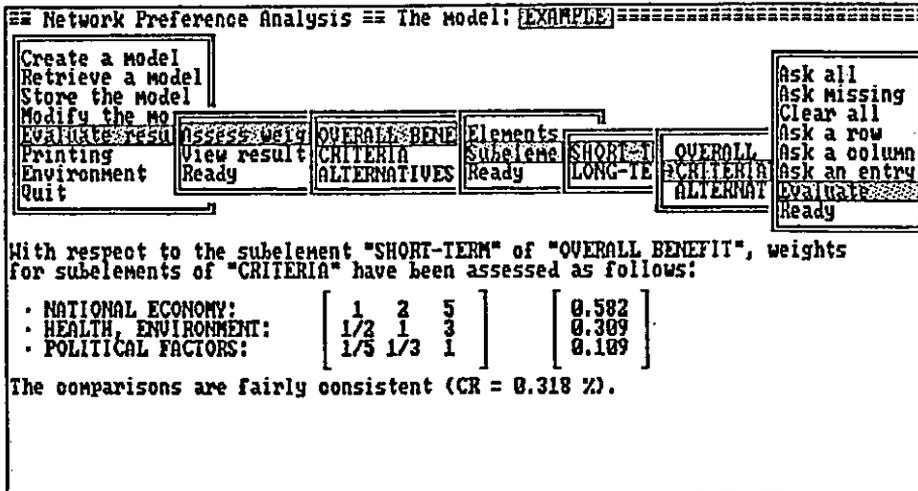


Figure 3. Assessing weights for elements.

FUTURE RESEARCH DIRECTIONS

We have gained experience in the use and implementation of DSSs through projects related e.g. to the planning of the Finnish energy policy. On the personal computer side, an AHP software of ours, IDA, has been used by the Parliament members in 1984 (Hämäläinen et al. 1985, Hämäläinen 1988). On artificial intelligence workstations, one project has just been completed. It involved the development of a DSS, called SETELI (Salo et al. 1988), for financial planning of the Finnish Post and Telecommunications Agency, and the system is in active use. For the future, we shall next outline some potential research directions.

Virtually all decision support systems begin the interaction with the DM after the phase where the problem structure has been defined. However, in many cases the identification of relevant factors to be considered poses the initial and even major difficulties. The decision situation is seldom clear to the DM. At times the DM just feels that "something must be done" with no well defined options in mind. There may be occasions when the problem solution is evident after the explicit structuring has been accomplished. So far, problem structuring has rather been an art than a science. Methods of artificial intelligence may be of use when developing practical procedures in this area.

As computational requirements of a DSS are often met by low-cost personal computers, they remain as potential devices for the actual implementation. Moreover, new concepts for the working

environment are brought in from more powerful engineering and artificial intelligence workstations. These include the use of windows, parallel processing, and a user interface driven by a mouse as a pointing device. There is also a growing number of software development aids available. All these factors contribute to the building of more sophisticated and user-friendly DSSs. Still, a successful implementation requires a synthesis of two separate skills, namely those of decision analysis and software engineering. Being an expert in either of the two ensures no success in the combination.

The usefulness of a DSS is largely determined by the ease of interaction with a DM. A clear, well-thought logic must lie behind a successful implementation of any method for decision support. The leading principles must be reflected in the design of the user interface. The goal should be to concentrate on the problem, not on the mastering of tricks of a certain software product. Thus, normally there should be no need for a DM to consult any manuals or other written instructions during a DSS session. Choosing the actions through menus is an approach that makes utilization of a DSS possible also for inexperienced users.

Once the problem evaluation has been completed through a DSS, the question of interpreting the results must be addressed. The step to offer a graphical representation as an alternative to a numerical counterpart is an obvious one, and not to be skipped. As a consequence of the availability of more expressive programming tools, a relatively minor effort here may lead to notable improvements. We followed the path described when improving an earlier product of ours, HIPRE (Hämäläinen et al. 1986). It is an implementation of AHP, and in Figure 4 the overall weights of energy alternatives are presented, together with the weight composition from the diverse decision criteria.

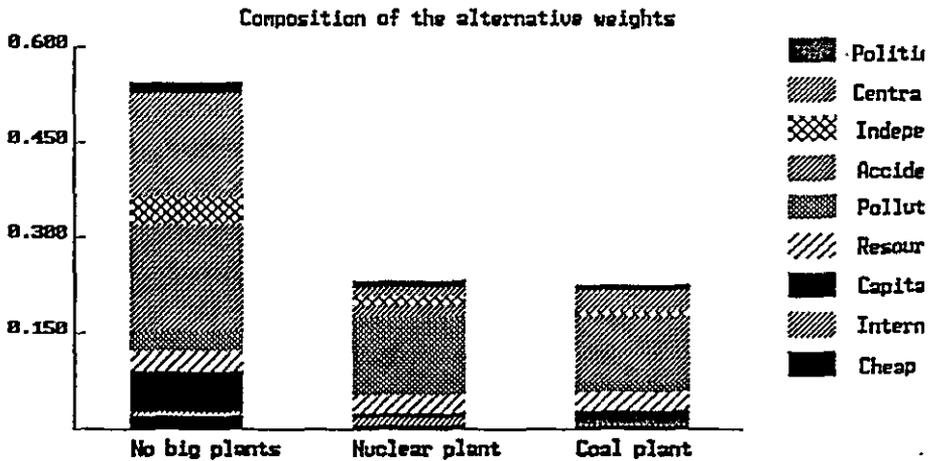


Figure 4. A graphical representation of the weights of the alternative energy policies.

Sensitivity analysis is a vital part of any evaluation of decision alternatives. It may be conducted with respect to the values of the DM or with respect to judgmental facts about alternatives. In either case the automation of the process may well be worth the effort. A systematic approach is to present few variations for values affecting the decision, for weighting of the relevant attributes as well as for the assessed 'hard' facts. These variations and resulting changes in the overall evaluation can be shown graphically. As an example - taken from HIPRE - see Figure 5, where criteria weights obtained by AHP are varied, and the respective shift in final alternative weights is shown. Obviously, the decision is also sensitive to which and whose values are taken into account, i.e. the problem definition.

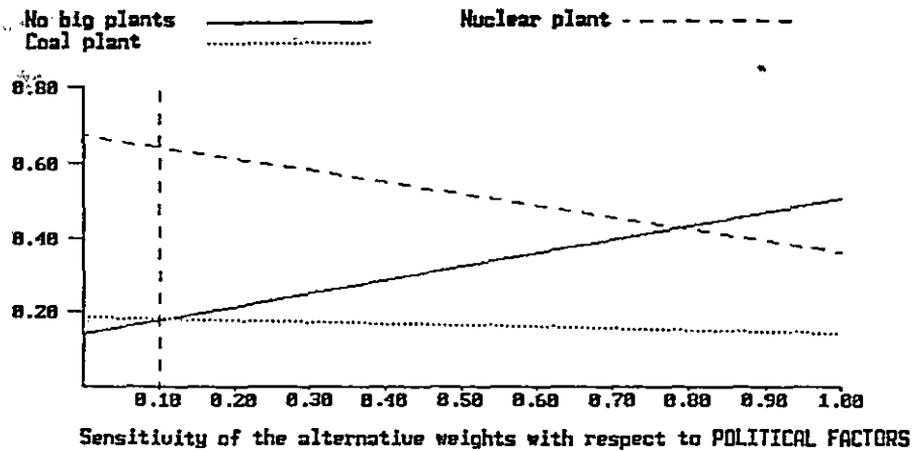


Figure 5. A graphical sensitivity analysis.

CONCLUSIONS

We have described one approach in the development of a DSS that requires no mathematically sophisticated DM to operate it. It must be emphasized that usefulness of most theoretical methods of decision analysis can only be tested after actual computer implementations have been carried out. It may be argued that a straightforward software doing strictly the minimum that the theory requires does the job. We think, however, that one has to be aware of what is going on in the computer world, and apply the successful, proved techniques of software engineering in DSS implementation. Dwelling on aspects of implementation is not to be seen as playing with the new toys, but rather as a serious effort to improve the usability of DSSs, in order to gain insight into matters that make a DM want to use a DSS to solve his or her problems.

REFERENCES

- Hämäläinen, R.P. (1988) "Computer assisted energy policy analysis in the parliament of Finland", *Interfaces* (to appear)
- Hämäläinen, R.P., Oehlandt, K., Seppäläinen, T. and Ruusunen, J. (1985) "An approach to decision support in national energy policy planning", *IEEE Conference on Systems, Man and Cybernetics*, Tucson, pp. 1008-1012
- Hämäläinen, R.P., Seppäläinen, T. and Ruusunen, J. (1986a) "A microcomputer-based decision support tool and its application to a complex energy decision problem", *17th Hawaii International Conference on Systems Sciences*, pp. 494-502
- Hämäläinen, R.P. and Seppäläinen, T. (1986b) "The analytic network process in energy policy planning", *Socio-Economic Planning Sciences*, Vol. 20 No. 6, pp. 399-405
- Saaty, T.L. (1980) "The Analytic Hierarchy Process", McGraw-Hill, New-York
- Saaty, T.L. (1987) "How to handle dependence with the Analytic Hierarchy Process", *Mathematical Modelling*, Vol. 9, No. 3-5, pp. 369-376
- Saaty, T.L. and Takizawa, M. (1986) "Dependence and independence: from linear hierarchies to nonlinear networks", *European Journal of Operational Research* 26, pp. 229-237
- Salo, A. and Hämäläinen R.P. (1988) "SETELI - The Strategy Expert for Telecommunications Investments" (unpublished manuscript)