OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

This document has been produced without formal United Nations editing. The designations employed and the presentation of the material in this document do not imply the expression of any opinion whatsoever on the part of the Secretariat of the United Nations Industrial Development Organization (UNIDO) concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries, or its economic system or degree of development. Designations such as "developed", "industrialized" and "developing" are intended for statistical convenience and do not necessarily express a judgment about the stage reached by a particular country or area in the development process. Mention of firm names or commercial products does not constitute an endorsement by UNIDO.

FAIR USE POLICY

Any part of this publication may be quoted and referenced for educational and research purposes without additional permission from UNIDO. However, those who make use of quoting and referencing this publication are requested to follow the Fair Use Policy of giving due credit to UNIDO.

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

COMPUTER AND AUTOMATION INSTITUTE OF THE
HUNGARIAN ACADEMY OF SCIENCES

TRANSFER OF COMPUTERIZED TOOLS FOR INDUSTRIAL PLANNING

UNIDO-UNIDPLAN SEMINAR AND WORKSHOP

BUDAPEST, HUNGARY

10-22 SEPTEMBER 1990

FINAL REPORT

Budapest
1990

1. ORGANISATION OF THE SEMINAR AND WORKSHOP

In the framework of the UNIDO-UNIDPLAN project UC/UT/INT/90/061 an international seminar and workshop has been organised in Budapest, Hungary.

The organiser was the National UNIDPLAN Focal Point (Computer and Automation Institute of the Hungarian Academy of Sciences) contracted by UNIDO through the TESCO company.

1.1 Participants

The participants were from four countries; Egypt, France, Hungary and the Philippines:

- -Mr. Rafaat Radwan (Egypt)
- -Mr. Emad Alin (Egypt)
- -Mr. H. Passeron (France)
- -Mr. A. Thabor (France)
- -Mr. M. Biro (Hungary)
- -Mr. P. Vasarhelyi (Hungary)
- -Mr. e. Valencia (Philippines)
- -Ms. E. Maximo (Philippines)

1.2 Agenda

The agenda of the meeting was the following:

08 September:		arrival of participants
10 September:	9.30 a.m.	pick-up of participants in the hotel
	10.00 a.m.	Opening (Mr. P. Vasarhelyi, Head, UNIDPLAN National Focal Point)
	11.00	Presentation of a French database and expert system for strategic planning at sectoral level and in small-medium enterprises (Mr.Thabor. France)
	12.30	Lunch
	2.00 p.m.	Presentation of the CONDOR Group decision support tool (Mr. Biro. Hungary)
11 September:	a.m.	Mr. Thabor, continuation
	p.m.	Application of Condor on examples from the Philippines and Egyipt (Mr. Biro)
12 September	a.m.	Discussion on the ways in which the French tool can be adapted to conditions in the Philippines and Egypt
	p.m.	Presentation of MILP: a linear programming tool for defining optimal product mix, etc. (Mr. Biro)
13 September	a.m.	DIVA French model for harmonized planning at the level of the sector and in large enterprises having numerous intersectoral relations (Mr. Passeron, France)
	p.m.	DIVA presentation continued
14 September	a.m.	Ways of adapting DIVA for use in the Philippines and Egypt
	p.m.	DIVA continuation

17 September	a.m.	Integrating MILP and Condor (Mr. Biro)
	p.m.	On-the-job training on Microsoft Windows under which Condor can be used (Mr. Biro)
18 September	a.m.	Adapting Condor/MILP
	p.m.	Microsoft windows training
19 September	a.m.	Adapting Condor/Milp
	p.m.	Microsoft Windows training
20 September	a.m.	Presentation of a computerised planning tool by the representative of Egypt
	p.m.	Microsoft windows training
21 September	a.m.	Presentation of the Foreign trade database developed in cooperation with ITC by the Philippines
	p.m.	Microsoft Windows training
22 September		departure of participants

Pick-up at the hotel was every morning at 9.00 a.m., morning session started at 9.30 a.m., lunch was at 12.30 in the institute. The afternoon session started at 2.00 p.m.,ended according to the logic of presentation and the wishes of participants.

1.3 Hand-outs and teaching material

The following documentation has been handed over to the participants:

- -CR-NET Commercial representative network (Egypt)
- -Electricity Pricing System (Egypt)
- -CONDOR-GDSS Group decision support system (Hungary)
- -MILP Linear Programming system (Hungary)
- -dSID Strategic decision support system (France)
- -Business plan and organizational audit (France)
- -EXPONET CTIS export promotion database system (Philippines)

The documentation is attached under Annexes 1,2,3,4,5 and 6.

1.4 Software transmitted

The Hungarian party transmitted to the participants three software tools ready for use in industrial planning processes:

- -Microsoft Windows, version 3.0
- -CONDOR GDSS, version 1.1
- -MILP

One copy of the above tools was handed over to the delegation of Egypt and the Philippines each.

2. RECOMMENDATIONS

2.1 Promotion of export oriented industrial production

The participating countries will take the necessary steps and recommend to UNIDO action along the following lines:

a) Exchange of detailed information on the Philippine and Egyptian foreign trade decision support systems, with a view to identifying the modules available in one which could be taken over by the other

b) Loading of international data taken over by Egypt from international sources in the databases under establishment in the Philippines and Hungary, in particular

- identification of data on general market characteristics available now in the Egyptian database and
- arranging for appropriate agreements with the owners of these data with a view to obtaining authorization for their downloading into the Philippine and Hungarian databases. This would allow the saving of the work and cost involved in the selection, collection, entry, validation of these data.

c) Making use of the Egyptian experience regarding the taking over of country-specific data (i.e. of data of interest to the Philippines or Hungary only) from international sources. In this connection a technical team of 2-3 persons from Egypt could pay a visit to the Philippines and advise on the methods to be used.

d) After the completion of the above the work aiming at the establishment of the databank on suppliers could be shared, each country working on a complementary module or data file.

e) In a later stage the value added could be considerably increased by the means of introducing economic analysis in addition to the provision of raw data. In particular

- common algorithms could be identified and the programming work could be shared
- the models under development in Egypt for the
  - debt management
  - economic reform

  would be of great interest to the Philippines and Hungary
- the tools under development in the Philippines for the analysis of the impact of social and economic reforms would be of interest to Egypt and Hungary
- the Hungarian model for linear programming could be included in the electricity pricing model of Egypt increasing thus the decision support power of the Egyptian model
- the algorithms developed in harmony with the above could become utility functions of the CONDOR Group Decision Support Tool leading to an efficient integrated foreign trade decision support system for use by all parties involved.

f) The experience gained in the participating countries regarding the implementation of international communication networks should be shared.

2.2 Sectoral planning

The database and expert system developed by the French society CIM/AERIS and demonstrated by Mr. Thabor could be of interest in supporting decision makers in the following sectors

- -food industry, in particular
  - -milk
  - -meat
- -electronics
- -chemical industry
- -car industry

The adaptation and introduction could be envisaged in the following steps:

- -selection of a specific sector in the country for a pilot project
- -a preliminnary initiation of the local case team regarding the approach, the dSID system and the underlying concepts and models
- -implementation of a sectoral analysis in the selected sector, based on individualised studies of the representative sample of companies from the sector. This could be a training exercise for the local team as well as the basis for the adaptation of the system to local conditions and constrains
- -evaluation of the usefulness of the system on the basis of the results obtained

Should the evaluation be favourable, a second phase could be initiated:

- -acquisition of the system
- -modification and improvement of the current system by its authors, in co-operation with the local team to ensure its necessary familiarization with the system
- -creation of a local structure and action plan to use the decision support tool in local industrial development

2.3 Macroeconomic model

The DIVA model developed [illegible] framework of the efforts aiming at the development of a macroeconomic model for North-East Africa.

IDSC (Egypt) is interested in adjusting the model to the multilingual context, in particular in its Arabisation.

3. RESULTS ACHIEVED

The UNIDO/UNIDPLAN project UC/UT/INT/90/061 achieved the following main results:

- Two computer aided planning tools: the CONDOR group decision support tool and the MILP linear programming tool has been transfered with the necessary Microsoft windows environment to Egypt and the Philippines, ready for use in the industrial planning process, with unlimited right for their use in the framework of UNIDPLAN program

- Two specialists from Egypt and the Philippines obtained personalised training on the use of the above computer aided planning tools.

- Experience gained in Egypt and the Philippines in the field of computerised decision support tools of interest to industrial planners could be exchanged for the benefit of the industrial planning organizations of both countries

- The activities in progress in Egypt, Hungary and the Philippines in connection with the establishment of the databases and information network required for the development of export oriented industries can be harmonised, resulting in each country in a decrease of the cost and time of implementation and in the increase of efficiency.

- An appropriate method and tool has been identified for the purpose of more efficient sectoral planning and a project concept has been elaborated for its introduction in the participating countries under eventual UNIDO assistance.

Computer and Automation Institute
Hungarian Academy of Sciences

United Nations
Industrial Development Organization

CONDOR-GDSS

CONsensus Development and Operations Research Tools
Group Decision Support System

developed by

Miklós Bíró
Péter Csáki
Mátyás Vermes

A TUTORIAL

CONTENTS

1. INTRODUCTION

A prevalent theme of modern industrial societies is that group activities are necessary, efficient as means of production and reinforcing of democratic values. Managers and knowledge workers spend a significant proportion of their time working in groups. Estimates of this proportion range from 60-80%. Unfortunately, most group meetings are not as productive as they could be. Companies' loss due to ineffectively managed meetings can be estimated in millions of dollars in a year [Nunamaker, 1988].

While the concept of improving group activity through the use of information technology (IT) has been discussed for several years, most early efforts to develop systems to support meetings met with limited success.

Recent advances have been driven by applying new information system technology. The widespread use of LAN based E-Mail, office automation systems, and decision support packages has accelerated the process.

There are many barriers to implementation of functional systems. Most of the reports are from the USA, where several university and industry groups have developed IT based systems to support meetings and group work: University of Arizona (PLEXSYS), University of Minnesota (SAMM), Microelectronics and Computer Technology Corporation (GROVE), Xerox PARC (Colab), Claremont Graduate School.

GDSS efforts are underway in European countries as well:
A prototype of a GDSS has been implemented at University of Leeds, UK [Burns et al., 1987], the MEDIATOR system of Jarke and Jelassi,INSEAD, France [Jarke et al, 1986], the BABA system of Gelleri et al, Budapest, Hungary [Gelleri et al, 1988], SCDAS, IIASA [Lewandowski, 1988].

The following two chapters attempt to make the reader familiar with the group decision support concept and give a brief historical review of GDSS systems. The last two ones report about the first version of the CONDOR GDSS system, which has been developed in the Operations Research Department of the Computer and Automation Institititute, HAS, Budapest.

2.DEFINITIONS

Even though the concept of decision support is well known, people should constantly be reminded that the systems are <u>support</u> systems, not decision making systems. Decision support by computers is very difficult and, despite their problems and limitations, human beings still far outperform computers in accomplishing many unstructured tasks. However, GDSS are being used to support activities such as collaborative writing and drawing, communication, negotiation, evaluating alternatives, planning, consensus building and even decision making.

The existing systems can be grouped into two broad classes: GDSS (group decision support system) and CSCW (computer based systems for cooperative work).

The distinction between these two types of systems is in the primary type of group support they each were designed to provide. GDSSs are more task oriented in that they provide a means for a group to work on and complete a task, such as reaching a decision, planning, or solving problems. CSCWs, on the other hand, are more driven by communication needs. They provide a means for small groups to communicate more efficiently, enabling them to jointly create or critique a document, for example. The distinctions between these two classes of systems are blurring. Some software tools developed e.g. as part of the PLEXSYS project (University of Arizona) exemplify the common area between GDSS and CSCW. [Dennis et al, 1988], [Nunamaker et al, 1989]. In time these two classes of systems will completely overlap, representing a single class of information technology systems to support electronic meetings.

GDSS has been defined as

"An interactive computer-based system that facilitates the solution of an unstructured problem by a set of decision makers working together as a group." [DeSanctis and Gallup, 1985] It also enhances communication among group members [Bui, 1987; Bui and Jarke, 1984; Kraemer and King, 1986].

Some of the existing systems and the tendency of future development shows the need to extend the definition and concept of GDSS. Meetings involve more than just decision making, they can also involve problem structuring, idea generation, idea organization, planning, creating and even elicitation of knowledge in the construction of expert systems. Group members can be located in different places and in different times, yet still work together to accomplish some common purpose. A complete recording of the group session aids productivity in subsequent sessions.

The new term proposed by [Dennis et al, 1988] is electronic meeting systems (EMS).

EMS combines the task-orientation of GDSS and the communication-orientation of CSCW.

EMS is defined as

"An information technology (IT)-based environment that supports group meetings, which may be distributed geographically and temporally. The IT environment includes, but is not limited to, distributed facilities, computer hardware and software, audio and video technology, procedures, methodologies, facilitation, and applicable group data. Group tasks include, but are not limited to, communication, planning, idea generation, conflict resolution, systems analysis and design, and collaborative group activities such as document preparation and sharing."

3.TECHNOLOGICAL FORMS OF GROUP DECISION SUPPORT (ELECTRONIC MEETING) SYSTEMS

Over the past decade, several forms of GDSS have been developed. A review of the literature suggests to derive a classification of GDSSs from the array of possible technologies. It it useful to conceive of a GDSS as a sociotechnical package comprised of

- hardware: conference facility, computing, telecommunications, and audiovisual equipment;
- software: general information processing software, database management systems, generalized application packages, decision modeling language,

  decision structuring techniques (e.g. brainstorming, nominal group technique, stakeholder analysis, Delphi technique)

  decision analysis technique (e.g. utility and probability assessment, multiattribute weighing analysis)

  communications software for both local and long distance text, data, voice, video transmission;
- organizationsware:

  organizational data (local and remote), group processes (authoritaran or democratic, consensual or conflictual, political or rational) management procedures for collaborative group work (developing participation, securing commitment to the decision, maintaining support of the group);

- people: participants in the group, support staff for facilitating the group's activities (f.e a facilitator, who operates the technology required /conducts the group meeting /trains the participants)

In terms of this four elements six kinds of GDSSs can be distinguished. The first two forms: electronic boardroom and teleconferencing facility, have primarily developed from the technology perspective, this is why we discuss them very briefly. The information center and the group network have also been developed primarily from an engineering point of view but with the goal of making "friendly" technologies that users find genuinely helpful. The decision conference and collaboration laboratory are the results of an attempted union of the psychosocial and technology perspectives. At the most advanced forms of GDSS we use, instead of 'software' the term 'groupware' for systems which support two or more, possibly simultaneous, users and that provide an interface to a shared environment.

3.1. The Electronic Boardroom

This is the most elementary type of the GDSSs and differs little from the nonelectroning decision room except that the audiovisual technology is computer based.

These early versions of the electronic boardroom have either disappeared from use or have been transformed into one of the other types of GDSSs.

3.2. The Teleconferencing Facility and Computerized Conferencing

The Teleconferencing Facility is designed primarily to facilitate meetings between groups at two or more locations. It has a conference room to mimic face-to-face meetings. A modern teleconferencing facility is usually suitable for 6-12 active and another 12-24 passive participants. It is being used increasingly for internal meeting in the USA. Some of them are being hooked up to corporate computers and databases to provide additional capabilities of a corporate information center.

Computerized Conferencing is much closer to GDSS than teleconferencing. Its first generation is an asynchronous form of a conference. The 'meetings' take place over an extended time period, usually around some prearranged topic.

Hardware:

- offices with workstations connected to public and private networks (Internet, CSNET, BITNET, Tymenet), telephone

Software:

- communication software for message broadcasting, conferencing software

Organizationsware:

- the conference chair initiates, conducts, summarizes and ends the meeting

People:

- participants, conference chair

There has been a great deal of experimentation with computerized conferencing [Hiltz and Turoff, 1981], [Hiltz, Turoff, and Johnson, 1989]. Most participants report that it hardly feels like a meeting at all. Its second generation will be discussed in 3.6.

3.3. The Information Center

The aim of the information center is to provide users with computer power, databases, software, and technical support that enables them to directly control their information environment. It started out as a way of distributed computing to managers and professional users throughout an organization. In many cases it operates in that way. But in some instances the center and its tools have become tailored to serve particular group of users, such as the marketing department, the planning staff, the corporate staff, and managers. It is this group oriented use of information centers that most properly falls under the GDSS label.

Hardware

- public areas with terminals connected to a mainframe computer,
- private cubicles or offices where center staff can work with individual users,
- a conference facility with large screen video projector(s), computer(s), display terminals.

Software

- packaged programs for data management, data retrieval and query, report generation, text handling, statistical analysis, mathematical and simulation modeling.

Organizationsware:

- the center operates on the organization's main computing facilities and accesses corporate databases as well as secondary data sources.

People:

- a manager
- several technical staff who, in turn, are backed up by the organizations data processing staff
- small-to-medium-sized groups.

3.4. The Decision Conference

This facility is discussed in the literature under the labels of GDSS, decision analysis, and group decision aid. What distinguishes the decision conference from the other GDSSs is its explicit focus on improving decision making by groups and its emphasis on the use of structured decision processes, mainly involving computer models but increasingly involving group process models as well.

Hardware

- a medium sized conference room with a large-screen video projector, a computer with terminals for different use: video terminals, terminals for voting or other input by the participants, and a control terminal for presenting participants inputs in graphic form and for accessing other sources of information (e.g. databases, general reference materials, results from previous conferences).

Software

- decision analytic technique: decision trees and influence trees, multiattribute expected utility models for single stage decisions, hierarchical evaluation structures for multiattribute utility analysis, algorithms for negotiations, spreadsheet models.

  Software for graphics, for vote tally and display.

Organizationswa~e

- meeting protocols regarding who participates, on what basis, with what voting rights, and with what consequences and commitments resulting from the process. Most of them emphasize "democratic" protocols rather than "authoritaran", "hierarchical" or "authoritative" ones.

People

- the actual people in the organization who are involved in making decision
- decision analysts who explain the available decision analytic tools and work with the participants in modeling their decision problem
- group process facilitator.

Illustration of the decision conference in industry are provided by GROUP DECISION AID of Perceptronics,Inc and the Decision Conferences on Decisions and Designs,Inc. In universities, the Decision Techtronics Group at SUNY (State University of New York), the Planning Laboratory at the University of Arizona (PLEXSYS), the SMU Decision Room, and the GDSS Laboratory at the University of Minnesota (SAMM).

Decision Techtronics Group has hosted approximately 60 decision conferences for private and public sector organizations located troughout the United States [McCartt and Rohrbaugh, 1989]. The PLEXSYS system has been used by more than 90 public and private organizations [Nunamaker et al, 1989].

3.5. The Collaboration Laboratory

The Collaboration laboratory is focused on computer support for face-to-face group work. Although decision making and problem solving might be involved in group work, the laboratory does not involve the use of formal decision models and quantitative techniques. Rather, it focuses on writing and argumentation and involves verbal models and qualitative techniques through the manipulation of text-oriented data and graphical images, which are the most common forms of data used in group meetings.

Hardware:

- workstations built into a conference table (eye contact!)
- shared electronic chalkboard.

Software (Groupware)

- text oriented tools including a common human-machine interface, mostly with WYSIWIS 'what you see is what I see' characteristics for presentation of images of shared information for all participants,
- public (shared) and private (not shared) windows on the workstations,

- applications

  group method of preparing outlines of ideas and associated text, (similar to individual tools for outlining but includes additional features to specifically aid for group collaboration),

  group method of evaluating plans and programs that have already been developed.

Organizationsware

- protocols mostly for equalizing rights

People

- participants.

There are three illustrations of the collaboration laboratory: Colab at Xerox Palo Alto Research Center, California; Project NICK at the MCC (Microelectronics and Computer technology Corporation) in Austin, Texas; Capture Lab at the Center for Machine Intelligence in Toronto (Canada).

3.6. The Group Network

The group network has its root in computerized conferencing but is also a response to its limitations.

The group network is focused on interactive computer support for small groups in geographically dispersed but nearby locations such as offices within a building or building complex.It is real time and interactive.

Hardware:

- networked microcomputer workstation with a keyboard, pointing device, and speaker telephone

Software (Groupware):

- each workstation has public and shared spaces,
- meeting scheduler application (the meeting scheduler allows participants to vote for a preferred meeting time and displays the result at each workstation; when a time is agreed upon, each participant's public schedule is automatically posted with the meeting time),
- shared applications, such as graphics, word processing, and spreadsheeting, which permit all participants to create, edit, or simply exchange graphics, text, or numbers.

Organizationsware

- a meeting chair, who calls, conducts, and terminates meetings; determines who has permission to enter calendar commands at any given time, when voting begins and ends, and when control of shared space is given or taken away.

People

- participants in two or more local places,
- group leader.

Prototype of group network exists at the MIT Laboratory for Computer Science, Massachusetts. (RTCAL, MPCAL, MBlink)

More detailes about these sytems, and evaluation of experiences with them can be found in [Kramer and King, 1988], and ten papers of the special issue of DSS on GDSS edited by Nunamaker (DSS,5,2,1989).

4. DISTINCTIVE FEATURES OF CONDOR

(CONsensus Development and Operations Research tools)

4.1 View of a General Purpose Framework of the Aspects of Group Decision Support Systems

CONDOR is a group decision support system. The basic purpose of a group decision process is the coordination of the decision related activities of the involved individuals or subgroups who may have different perspectives or priorities.

Many early GDSS were task driven. There were designed to meet the needs of one group performing one task. Obviously, the tendency is to provide a GDSS as universal as possible. This can be done with the toolkit approach.

Toolkits are collections of specific tools that address various part of the meeting's process. The key advantage provided by toolkits is flexibility.

The CONDOR programsystem has been designed according to the toolkit concept.

A conceptual taxonomical framework has been worked out, which includes the already implemented features and those to be implemented in future versions.

4.2 Hierarchy of the aspects of GDSS

The hierarchy of the aspects of group decision support systems is represented by the numbering of the aspects in this written document.

Those features which are followed by a '*' are implemented in CONDOR Version 1.0, those followed by a '-' will be implemented in forthcoming versions.

4.2.1. Decision Support

4.2.1.1. Passwords and Rights

Group decision making usually involves a group of decision makers and a technical supervisor (facilitator) of the decision making process. Each of them has the right to access or modify specific data in specific phases of the process. These rights (*) together with the users' names and passwords (*) can only be accessed and modified by the supervisor.

4.2.1.2. Criteria

The decision making process usually begins with the definition of the problem, which, besides the verbal description, essentially consists of the development of the criteria used in evaluating any relevant alternatives.

4.2.1.2.1. List of Criteria

This is the simplest form of the definition of the criteria. In fact, some systems are not even prepared to handle any other mode of definition. In this case, individual views can always be expressed by simple weighing of the alternatives. The lack of interest of a decision maker in a criterion can be expressed by assigning zero weight to it.

4.2.1.2.2. **Hierarchy of Criteria**

The hierarchical development of the criteria (*) facilitates the consideration and weighing of all relevant aspects of the problem, and at the same time, it contributes to their expected independence.

4.2.1.2.2.1. **Consensus on Hierarchy**

In some cases, it may be desirable to reach a consensus on the hierarchy of criteria before starting any other phase of the decision making process. From a different perspective, this approach relieves the decision makers from the burden of constructing a hierarchy of criteria by themselves.

4.2.1.2.2.2. **Individual Hierarchies**

The weights assigned to the criteria are individual for each decision maker. However, the differences in the structural views of the criteria cannot be expressed by weighing. On the other hand, individual hierarchies (-) can be meaningfully used only if the ranking of the alternatives is ordinal, that is only their order and not their rate is significant.

4.2.1.3. **Alternatives**

The purpose of any decision is a choice between alternatives whose following categorization is defined by the nature of the decision making problem.

4.2.1.3.1. **Naturally Given Alternatives**

There is no need for expert involvement in the determination of the alternatives if they are naturally given. This is the case of bid evaluation for example, and of any selection from a predefined finite set in general. This is the only case supported by many systems.

4.2.1.3.2. **Alternatives to be Generated**

If the alternatives are not predifined then they must usually be generated by experts or the decision makers themselves.

4.2.1.3.2.1. **Model Solutions**

If a problem is structured enough to lend itself to modeling (-), then efficient solutions to the model yield usually valuable alternatives. The construction of a model is a task that requires special expertise. The use of models for generating alternatives is usually not integrated with group decision support systems.

4.2.1.3.2.1.1. **Modeling Expert**

The direct involvement of a modeling expert is the immediate solution.

4.2.1.3.2.1.2.**Modeling Support System**

The task of partly replacing modeling experts by so called modeling support systems is the subject of intensive study nowadays. The knowledge of modeling experts must be captured and incorporated into the system. Such a tool is under development by our group.

4.2.1.3.2.2. **Unstructured Problem**

If a problem is not structured then the decision must rely on alternatives proposed by the decision makers. While the optimality of these alternatives is not guaranteed, the data requirements of this approach are inferior to those of a mathematical model.

4.2.1.3.2 2.1. **Expert Involvement**

The decision makers may invite experts for the generation of efficient alternatives (*) which they can choose from afterward.

4.2.1.3.2.2.2. **Expert System**

A more advanced approach is the use of an expert system developed for the specific problem domain under consideration.

4. 2.1.3.3.**Dynamic Alternatives**

The decision alternatives belong to this class if their attributes change over time, are dependent on choices or random events related to other alternatives. The idea of integrating the handling of dynamic alternatives (-) into a general purpose group decision support system is original. The models that can be used in this case are related to the fields of simulation, decision analysis and project management.

4.2.1.4. **Rankings**

After the identification of the criteria and of the alternatives, a ranking of the latter is advanced first by each individual, then by the whole group of decision makers. Individual and group acceptance and aspiration levels may be set for each criterion (-).

4.2.1.4.1. **Individual Rating and Ranking**

4.2.1.4.1.1. **Objective Data Available**

The availability of objective data may have two different meanings.

4.2.1.4.1.1.1. **Utility Evaluation**

The objective data may mean measured values with given measurement units. In this case the utility of these values must be evaluated. (The meaning of utility is explained below under the title Utility Functions.)

4.2.1.4.1.1.2. **Model Experiments**

If the availability of objective data induces the application of a model as mentioned earlier, then the model provides the utilities of the alternatives. If the model is already built, then the execution of model experiments with parameters reset according to the judgement of the decision maker, may yield more efficient or acceptable alternatives.

4.2.1.4.1.2. **Subjective Rating**

If no objective or detailed data are available then the subjective judgement of the decision makers must be called upon. Methods like the Analytic Hierarchy Process (-) or ELECTRE (-) are appropriate in this case. If more precise, cardinal utility measurements can be obtained, then Multi-attribute Utility Theory methods (*) can be used.

4.2.1.4.1.2.1. **Utility Functions**

Utility functions are used to reduce evaluations with respect to incommensurable criteria to a common scale. Utility functions may be constructed according to the requirements of the users. Some usual forms are staircase, piecewise linear, ordered symbolic functions, and utilities assigned to the satisfaction of some relations called rules.

4.2.1.4.2. **Group Ranking**

Group ranking could be considered theoretically as a special multicriteria decision making problem, where each member of the group has his own set of criteria. A different approach is necessary however, since in this case a social consensus has to be reached.

4.2.1.4.2.1. **Choice of Voting Mechanisms**

There is theoretical evidence that there is no single method of aggregation of individual decisions which results in a acceptable group decision in all cases under realistic requirements. As a solution, several methods of aggregation are offered by the system (*).

4.2.1.4.2.1.1. Weighted Average with Voting Power

This is the only voting mechanism included in most systems.

4.2.1.4.2.1.2. Borda Count

4.2.1.4.2.2. Identification of Consensus Preventing Factors

If no voting mechanism yields an acceptable decision then the decision makers are supported in identifying the factors preventing the consensus (-).

4.2.1.4.2.2.1. Discordance Indicators

Numbers indicating the discordance of the individual decisions from the group decision (-).

4.2.1.4.2.2.2. Consensus Seeking

Provides support for negotiating those crucial questions whose resolution is most suspectible to yield consensus (-).

4.2.2. Man-Machine Interaction

Besides its functional features listed above, CONDOR is equipped with a new generation interface based on the standard Microsoft Windows operating environment. This permits the raising of the user-friendliness of the IBM PC/AT based system to a workstation level.

The window based multitasking graphics environment allows a

* simulataneous (parallel contact),
* visual,
* active (ability to modify anything in the sphere of authority),
* object oriented (related objects, objects and their attributes are kept together) contact with all necessary information and several parallel processes.

4.2.2.1. **Contact with All Necessary Information**

Typical information which the user can simultaneously overview without losing any visual or even active contact are the following for example:

* decision makers,
* hierarchy of criteria,
* weights of the criteria,
* alternatives,
* evaluation of the alternatives with respect to criteria,
* graphical representations (charts, maps).

The system provides a guided mode of operation (-) for novice users and an expert mode of operation (*) where the facilities of the system can be used as flexibly as required by the experienced user. Unnecessary information is removed from the screen when a new function is invoked, in order to avoid screen clutter (*).

4.2.2.2. Contact with Several Parallel Processes

Some of the parallel processes with which simultaneous contact may be highly desirable are:

* a calculator (Microsoft Windows),
* a spreadsheet (Microsoft Excel),
* a calendar with personal notes and alarm function Microsoft Windows),
* any dependently or independently implemented method.

Data exchange between the above processes is made possible by the DDE (Dynamic Data Exchange) feature of the Microsoft Windows environment through the clipboard accessible by all processes.

5.UTILIZATION OF CONDOR

5.1 Basic features

The first step of a decision making process supported by CONDOR is the input of the problem components by a facilitator.

The following steps are typically the assignment of weights to the importance of the criteria, the evaluation of the alternatives with respect to the basic level of criteria, and the individual then group ranking of the alternatives.

The steps are illustrated trough the following example.

The Smith family has to make decision on what car to buy. Each family member has some idea about the new car. They consider different criteria: technical characteristics, color, number of doors, price, social status appropriateness etc. There may be differences in opinion, but all family members agree in desiring an optimal solution which is satisfactory to all.

This is a group decision making problem in a soft negotiation situation.

Weighing the criteria

Decision makers have preferences in criteria, and according to their preference, independently from each other, they should assign weights ranging from 1 to 100 to each of the criteria. For example, a technically oriented family member will associate a weight of 90 to the consumption and maximal speed of the car, and much less to its color.

Evaluating the alternatives

Each of the alternatives should be evaluated on a scale of 1 to 100 with respect to the basic level criteria in the hierarchy. In the car buying example maximal speed, oil consumption, size, price, .. will be 'scored' for each type of car available. In this version of CONDOR no distinction will be made between measurable (f.e. oil-consumption) and non-measurable (f.e. social status appropriateness) criteria. Decision makers will evaluate the alternatives independently from each other.

Individual ranking

Once the weighing of the criteria and the evaluation of the alternatives is completed, the decision makers' individual preference list is be provided by the CONDOR programsystem. For each decision maker, his evaluation of the alternatives with respect to the basic level criteria is combined with the weights of criteria, and the alternatives are ranked according to their final value. This is method is called MultiAttribute Utility Decomposition (MAUD) for the individual ranking of alternatives.

Group ranking

Individual decisions are to be aggregated. According to Arrow's theorem known in the theory of social choice, there is no single method of aggregation individual preferences which results in an acceptable group decision in all cases. As a solution, two methods of aggregation are offered by CONDOR: weighted average and the Borda count. The Borda count of an alternative is the sum for all decision makers of the number of alternatives ranked below the given alternative.

5.2 Use of the System

The CONDOR Version 1.0 works on a single IBM PC compatible or PS/2 computer in MS-Windows 2.03 operating environment. (640 KB RAM, EGA card, and mouse are required.)

The user must be familiar with the basic feature of the MS-Windows system: menu selection, moving icons by mouse, data input into dialogue boxes, etc. (See User's Manual for MS-Windows).

CONDOR is based in the AROMA (AggRegated Object MAnagement) system, through which self-explanatory tree handling functions are performed.

The decision facilitator and the decision makers generate the hierarchy of criteria and the alternatives using the AROMA system.

CONDOR is delivered on a single floppy disk which serves as key disk. Its contents camn be copied to a hard disk, but the program can only be used if the key disk is inserted into drive A.

The directory containing MS Windows must be in the current DOS path.

CONDOR is started by typing: WIN CONDOR.

Menu items in the main menu:

DECISION TASK

INDIVIDUAL DECISION

GROUP DECISION

The mainmenuitems consist of pop-up menus.

The users' burden of moving from one menuitem to the other is minimized. The decision facilitator will work in the first menuitem while preparing the decision task. He will work in the third menuitem only for groupdecision processing. The decision makers will evaluate the alternatives and establish the ranking in the second menu environment, moving to the first menuitem will be necessary for loading/saving the decision task only.

Submenus of main menu items:

DECISION TASK submenus:

New task

Load task

Save task

Decision makers

Alternatives

Criteria

INDIVIDUAL DECISION submenus:

Password

Weighing criteria

Rate alternatives

Rank alternatives

View ranking

GROUP DECISION submenus:

View rates/weights

Compare DM's weights

Weighted average

Borda count

View group ranking

Comments to the DECISION TASK main menu item:

New task: This menuitem is for creating/modifying a decision task. The earlier version can be saved before modification. For the operation to be successful, there must be enough space on the disk and the name of the file must be valid.

Load task: This menuitem is for loading a decision task created and saved earlier.

Save task: This menuitem is used for saving to the disk a decision task created previously in the memory. After successful saving the decision task will remain in the memory.

Decision makers: This menuitem is used for defining/editing the names, passwords, and rights of the decision makers. The decision makers will be associated with a node on the tree of decision makers. The structure of this tree does not affect the decision process, however, for convenience, the facilitator with the name 'SUPERVISOR' will associate himself to the root of the tree, and all other nodes will be connected to the root. Each node has attributes. The PASSWORD of the decision maker will be written into the text field of the attribute by the facilitator, the RIGHT will be coded into the numerical field.

Alternatives: This menuitem is used for creating the tree of alternatives. The tree structure has no effect in the decision making process. The attributes of the nodes are not relevant here.

Criteria: This menuitem is used for describing the criteria. A hierarchical approach can be followed by first identifying the general aspects of the decision problem, then gradually decomposing them into more basic aspects. The tree structure is important for efficient use of the MAUD technique. Without a tree structure, and with more than about seven criteria the decision task will be hardly digested by the decision makers.

Comments to the INDIVIDUAL DECISION main menu item:

Password: This menuitem prompts for the name and the password of the decision maker. Distinction is made between lower case and upper case letters. The proper name and passwords are needed for successful logon. The rights are automatically assigned subsequently.

Weighing criteria: This menuitem is used for entering the weights of the criteria. Each node of the criterion tree should be visited, and the weight of the criterion should be typed into the numerical field of the node's attribute. The weights are in the range 1..100. There is no restriction on the sum of the weights. The weights of the brother nodes will be divided by the sum of the weights when calculated the weighted sums.

Rate alternatives: This menuitem is used for rating (evaluating) the alternatives with respect to the basic level criteria. After selecting this item, a listbox will appear listing the criteria corresponding to the leaves of the criterion tree. A sample of the tree of alternatives will appear after selecting any item from this list. Values from 1..100 should be typed into the numerical field of the attributes of the alternatives. All alternatives must be evaluated with respect to all basic criteria. The evaluation process can be interrupted or terminated by pressing the 'Ready' button in the dialogue box. This must be done by first uncovering the dialogue box if it become covered by the other windows.

Rank alternatives: This menuitem is used for calculating the overall ranks of alternatives by the decision makers. The weighted sum of the evaluation of an alternative with respect to the criteria is calculated. This calculation is performed for all of the alternatives. The resulting values appear as the attributes of a sample of the tree of alternatives.

View ranking: This menuitem shows the (ordered) individual preference lists.

Comments to the GROUP DECISION main menu item:

View rates/weights: This menuitem invokes a special view of the data previously entered by decision makers. Three listbox will appear on the screen: one for the decision makers, one for the criteria, and the third one for the alternatives. Select a decision maker from the first box, a basic level criterion from the second one, and the third box will list the alternatives and their ranks according to the evaluation given by the selected decision maker. If the criterion selected in the second listbox is not a basic level criterion, then the third listbox shows the importance weights assigned to its subcriteria by the decision maker.

Compare DM's ratings: This menuitem invokes a further special view of the data previously enterd by the decision makers. Three listbox will appear again: one for the alternatives, one for the ordered list of the of the basic level criteria, and the third one for the decision makers. Select an alternative from the first box, a basic level criterion from the second one, and the third box will list the evaluations the selected alternatives by by all decision makers.

Weighted average: This menuitem is used for aggregating the individual preference lists into a group preference list which is the final outcome of the group decision process.

The decision makers are equally weighted in the current version.

Borda count: This is an alternate method for aggregating the individual preference lists. In this case the rank of an alternative is determined by the sum for all decision makers of the number of alternatives ranked below the given alternative.

View group ranking: This menuitem shows the aggregated preference list in a listbox.

The Chart descendents submenuitem of the Attributes main menuitem of the **AROMA** window can also be used for graphically overviewing the preference lists or weights assigned to the descendents of a node.

2. Hardware and software requirement

IBM PC/AT with

640 K memory,

Hard disk,

EGA or VGA monitor,

mouse,

MS-DOS 3.3 operating system,

Microsoft Windows 2.0 operating environment.

3. Contact person

Dr. Miklós Biró

Computer and Automation Institute

Hungarian Academy of Sciences,

Budapest,

Victor Hugo u. 18.

H-1132

Hungary

Tel: (36)(1) 149-7531

Fax: (36)(1) 129-7866

Email: h676bir@ELLA.UUCP

REFERENCES

[Bui, X.T., 1987]
Co-oP, A Group Decision Support System for Cooperative Multiple Criteria Group Decision Making, Lecture Notes in Computer Scienc`. Eds: Goos and Hartmanis, Springer-Verlag, 1987.

[Crowston, K. and Malone, T.W.]
Intelligent Software Agents, Byte, December (1988), 267-271

[Dennis, A.R., George, J.F., Jessup, L.M., Nunamaker, J.F., Vogel, D., 1988]
Information Technology to Support Electronic Meetings, MIS Quarterly, 12, 4 (1988)

[DeSanctis, G. and Gallupe, B.,1987]
A Foundation for the Study of Group Decision Support Systems, Management Science, 33, 5, (1987), 589-609

[Gelleri, P.,1988]
Decision support systems (in Hungarian), Computerworld Szamitastechnika, 3, 9 (1988), 30-32

[Gray, P. and Olfman, L. 1989]
The User Interface In Group Decision Support Systems, Decision Support Systems 5 (1989), 119-137

[Hiltz, S.R., Turroff,M., 1985]
Structuring computer-mediated communication systems to avoid information overload. Communications of the ACM, 28, 7, 680-689.

[Horak, W., 1985]
Office document architecture and office document interchange formats: current status of international standardization. Computer, 1985, October, 50-60.

[Jarke, M., Bui, X.T., Jelassi, M.T., 1986],
Micro mainfraime DSS for remote multi-person decisions, in: Managers, Micros and Mainfraimes, John Whily and Sons Ltd (1986)

[Jelassi, M. T. and Beauclair, R., 1987]
An Integrated Framework for Group Decision Support Systems, North-Holland Information & Management 13 (1987) 143-153

[Jelassi, M.T. and Foroughi, A., 1989]
Negotiation Support Systems: An Overview of Design Issues and Existing Software, Decision Support Systems 5 (1989), 167-181

[Kraemer, L.K. and King, J.L., 1988]
Computer-Based Systems for Cooperative Work and Group Decision Making, ACM Computing Surveys, 20, 2 (1988), 115-146

[Lewandowski, A., 1988]
SCDAS-Decision Support System for Group Decision Making, IIASA WP-88-48

[Malone,T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., Cohen, M.D., 1987]
Intelligent information-sharing systems. Communications of the ACM, 30, 5, 390-402.

[Nunamaker, J.F., Applegate, L.M., Konsysnsky, B.R.,1988]
Computer-Aided Deliberation: Model Management and Group Decision Support, Operations Research, 36, 6 (1988), 826-848

[Nunamaker, J.F., Vogel, D., Konsysnsky, B., 1989]
Interaction of Task and Technology to Support Large Groups, Decision Support Systems 5 (1989) 139-152

[Nunamaker, J.F., Vogel, D., Heminger, Martz,B., Grohowski,R., McGoff, C., 1989]
Experiences at IBM with Group Support Systems: A Field Study, Decision Support Systems 5 (1989) 183-196

[Olson, M.H., Lucas, H.C., 1982]
The impact of office automation on the organization: Some Implications for research and practice. Communications of the ACM, 25, 11, 838-847.

[Oper, S., 1988]
A Groupware Toolbox, Byte, December (1988), 275-282

[Pinsonneault, A., Kraemer, K.L., 1989]
The Impact of Technological Support on Groups: An Assesment of the Empirical Research, Decision Support Systems 5 (1989) 197-216.

[Sarin, S., Greif,I., 1985]
Computer-based rel-time conferencing systems. Computer, 1985, October, 33-45.

[Stefik, M., Foster, G., Bobrow, D.G., Kahn, K., Lanning, S., Suchman, L., 1987]
Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. Communications of the ACM, 30, 1, 32-47.

[Vogel, D., Nunamaker, J., Applegate, L.M., Konsysnsky, B.R.,1987],
Group Decision Support Systems: Determinants of Success, Proceedings of the &th International Conference on Decision Support Systems, June 8-11, 1987.

[Yu, P.L., 1973]
A class of solutions for group decision problems. Management Science, 19, 936-946.

CONDOR - GDSS

CONsensus Development and Operations Research tools

Group Decision Support System

Short Description

1. Purpose of the system

The purpose of CONDOR-GDSS is the support of a group of decision makers in the selection from a set of alternatives according to a system of criteria as complex as necessary.

First the decision makers' names, access rights, and passwords are entered by the system facilitator.

Next, the system supports the construction and subsequent modification of the hierarchy of criteria influencing the evaluation of the alternatives. The relative significance of the criteria and the merits and drawbacks of the alternatives with respect to these, are individually judged and commented by each of the decision makers.

CONDOR automatically computes first the individual then the group ranking of the alternatives with respect to the overall system of criteria. In order to overcome possible difficulties resulting from voting anomalies, CONDOR offers a choice of voting mechanisms including the Borda count, which has recently been proven to be the best in some sense.

The method of evaluation can be promptly customized according to a desired specification.

CONDOR supports the above decision making process with the most advanced operating environment on the IBM/PC category of personal computers. The Microsoft Windows environment allows for a simultaneous, active contact with all necessary information (hierarchy of criteria, evaluations, charts, etc...) and independent tools (calculator, calendar, spreadsheet, etc...).

If a customer is in possession of the Microsoft Windows development tool kit, a version of the system is available, where the form and method of the evaluations can be altered by mostly graphical manipulations using a dialog box editor. This feature is made possible by a UIMS (User Interface Management System) built into CONDOR.

M I L P

LINEAR PROGRAMMING SYSTEM

A concise user's guide
for version V1.50

Budapest, September of 1990

Computer and Automation Institute,
Hungarian Academy of Sciences (MTA SZTAKI)
H-1518 Budapest, P.O.B. 63.

CONTENTS:

A. General Introduction

The MILP program system can be used to solve linear programming (LP) problems given in general form. It runs on IBM PC/XT, AT or compatible computers. The system has a version which runs only with an arithmetic coprocessor installed (which means higher speed). There is an other version which is able to run without the arithmetic coprocessor. The system has both simple, and double precision versions.

The optimization algorithm is based on the upper-bounded revised simplex method, employs a multiple pricing technique, and it is equipped with a special first phase procedure [Maros,I.: "A General Phase-I Method in Linear Programming", European Journal of Operational Research, Vol. 23(1986) pp. 64-77]. In addition, it offers some new algorithmic techniques described in section B. The procedure stores the inverse matrix in product form. The identifier of the procedure used at reinversion: Real Time Pivot Procedure (RTPP). Its data storage and data handling techniques are primarily helpful in the efficient solution of large size, low density problems.

The constraints of the LP problem may be of range, or non-restricting type, in addition to the usual <,>,= types. The objective function must be given as part of the constraint matrix (it can be any row of the latter). The form of a constraint is:

$$\sum_{j=1}^{n} a(i,j)*x(j) \;\begin{matrix} < \\ = \\ > \end{matrix}\; b(i)$$

The variables x(j), included in the above formulation, are called structural variables.

The MILP system transforms the > constraints to < constraints by multplying with -1, then makes an equality out of all constraints by introducing a logical variable y(i):

$$y(i) + \sum_{j=1}^{n} a(i,j)*x(j) = b(i)$$

Individual bounds can be given for each variable:

$$l(j) <= x(j) <= u(j) \quad j = 1,...,n$$

$$0 <= y(i) <= r(i) \quad i = 1,...,m$$

The above bounds [l(j), u(j), r(i)] may be finite or infinite. (Infinity must be given to MILP as 1.0E35.) The individual bounds given for logical variables mean range type constraints, since the double constraint

$$b(i) - r(i) <= \sum_{j=1}^{n} a(i,j)*x(j) <= b(i)$$

becomes a collective and an individual constraint

$$y(i) + \sum_{j=1}^{n} a(i,j)*x(j) = b(i)$$

$$0 <= y(i) <= r(i).$$

In case of a non-restrictive constraint, the logical variable is a free variable (unrestricted in sign).

MILP can accept the LP problem to solve in two different formats: in MPS format, which nowadays is an industrial standard, and an own input format, which is mainly advantageous in case of problems generated by data processing programs. The description of both formats is included in sections C. and D. of this documentation.

There is a possibility for the intermediate solutions to be saved (DUMP) with regular frequency to a disk file, from where the run can be continued next time. The dump is always made into the file MILP.DMP, thus it is suitable to rename this file after the run for later use. The program can read a dump at restart from a file with arbitrary name.

B. Parameters of the Optimization

The optimization process is controlled by several parameters that can be given by the user as well. All of these has a default value, which appears after the problem is read in. The possible changes can be done interactively.

The parameters are the following:

identifier	default value	meaning
epiv	1.0E-7 (1.0E-6)	pivot tolerance, for the absolute value of the pivot element
eopt	1.0E-5 (1.0E-4)	optimality tolerance, for the absolute value of the shadow prices
erel	1.0E-13 (1.0E-5)	tolerance of the relative difference, for additive operations

Remark: The parentheses contain the default values of the simple precision version.

diro	1	direction of the optimization (1=max, -1=min)
nsub	4	the maximal number of columns included in multiple pricing (1 <= nsub <= 4)
ivfr	30	frequency of the reinversion (measured in the number of changes of basis)
dufr	9999	frequency of dumping (measured in the number of iterations)
trac	1	level of the iteration report about the run: 0 : no iteration report 1 : one line per major iteration 2 : one line per minor iteration

The parameters are modified by first answering the following prompt with '2':

```
--------------------------------------------------------------
Give command (1:Display, 2:Modify, 3:Tune algorithm, 4:Run):
--------------------------------------------------------------
```

then typing the name of the appropriate parameter, an equal sign, and the new value of the parameter. For example:

---

nsub=3

---

or

---

diro=-1

If we answer '1', then the current values of the parameters appear.

The MILP system offers the possibility of using one of the algorithmic techniques best suited to the specific properties of the problem at hand, in order to improve the efficiency of the solution. This selection is initiated by answering '3' to the above prompt. A sub-menu will appear giving the following choices:

DYN: column selection technique based on dynamic scaling of the shadow prices. Its use is suitable if there are major deviations in the magnitude of the data.

DEG: column selection technique to prevent degeneration. Its use is suitable if the right hand side contains many zero entries.

ADC: a special composite phase-I technique, which can dynamically take into account the real objective function during the search for a feasible solution. Its use is suitable when the first phase is expected to have a large number of steps. One of the signs of this fact is that the starting basis is highly infeasible, that is it is far from feasibility.

DYN and DEG are mutually exclusive, they both can be combined with ADC separately.

It must be noted, that all three algorithmic techniques result in extra computing work by major iterations, as compared to the normal iterations. At the same time, a significant reduction in the total number of iterations can be expected in the suggested cases, which results eventually in the reduction of the solution time.

If there are no more changes to be made, then we answer '4' to the reappearing prompt, and the program proceeds by asking what the starting basis should be. If there is a previously dumped basis, then we can start from there; otherwise, we start from the basis consisting of purely logical vectors (trivial basis=unit matrix).

If we start from a dump, it does not have to be compatible in size with the current problem to solve. In these cases, MILP makes use of as much information from the dump as it can interpret. A dump file can also be created by an editor, if we know the dump format. One way of doing this is to make a dump during the solution of a problem, and to use its format.

C. <u>MILP Input Format</u>

The following information must be given in a record oriented ASCII character (text) file (without tab characters):

Record 1: name of the problem (max. 40 characters)
Record 2: dimensions of the problem: number of constraints (m), number of variables (n), index of the objective function row (mc)
Beginning with record 3, m relation codes, 10 codes per record, with the following meaning:

<=	1
=	0
>=	-1
<>	3 (non-restrictive constraint)

m right-hand-side values, beginning on a new record, five entries in a record, decimal point must be provided;

m range values starting on a new record according to the following:

if the range given for a constraint is positive, then it is considered as a constraint range, 5 entries per record, decimal point must be provided;

the following structure is repeated n times, starting on a new record:

column head (separate record): column index, number of nonzero entries in the column, lower bound of the variable (0 must be specified as well), upper bound of the variable (infinity must be specified as 1.0E35),

column entries: as many (row-index,entry) pairs, as there are nonzero entries in the column, 5 pairs per record.

The provided sample problems, whose file extensions are empty, help in the understanding of the MILP input format.

There is few data checking when MILP input is used, thus the program run is terminated in case of a formal data error (i.e. the reading of a non-numeric character). For this reason it is advisable to use this facility primarily in the case of syntactically correct data systems generated by programs.

D. MPS Input Format

The MPS (Mathematical Programming System) input of the MILP system is essentially identical to the input of the IBM MPS system. The only differences occur at those points, where the IBM linear programming system requires specific input data that are only valid in that program system. MILP gives a message during the input in these special cases.

1. General Format of the Input Records

Two types of records are permitted:

1. Control records, which relate to the type of forthcoming data records.
2. Data records, which contain the data of the linear programming problem.

The control records (with the exception of NAME) consist of a single command word, which begins in the first position of the record.

The data records are divided into six fields, whose type is the same in every record. The maximal length of the records is 80 characters. The location of the data fields in the records is the following:

field	position	type
1.	2 - 3	code
2.	5 - 12	name
3.	15 - 22	name
4.	25 - 36	value
5.	40 - 47	name
6.	50 - 61	value

Remarks about the individual fields:

1. The code field (field 1) may be one or two characters long, and can only contain the characters corresponding to the data section. If the code consists of one character, then it can be placed on any position within the field.

2. The name fields (fields 2, 3 and 5) are eight characters long, and can contain arbitrary characters. Every character is significant, including the space (blank). The names are used by the program to identify the data of the problem.

3. The value fields (fields 4 and 6) are twelve characters long and can contain numerical values with decimal points. The numbers may be signed or unsigned. If there is no sign, the program assumes a + sign. The numbers can be given in exponential form as well (i.e. 1.73E-1), where the number after E is the exponent, which cannot be more than two characters long, not including the possible sign of the exponent. The first space character marks the end of the exponent. The number can be arbitrarily positioned within the field.

4. Comment records are introduced with a * character in the first position, which can be followed by an arbitrary character string. The comment records are completely ignored by the program during the input.

5. Every value field (fields 4 and 6) can have a $ character in the first position, which can be followed by an arbitrary character string as a comment. The program ignores the remaining part of a record after such a $ mark.

2. The Structure of the Input File

The first record on the input file has to be a NAME control record, and the last record has to be an ENDATA control record.

2.1. Problem Name, NAME

A NAME record makes it possible for the user to assign the desired denomination to the data set, to uniquely identify the problem to solve. The format of the NAME control record is the following:

position 1 - 4	15 - 22
NAME	Identifier of the problem

The identifier of the problem can be an arbitrary character string.

2.2. End of the data set, ENDATA

The ENDATA control record marks the end of the input data set on the data file. The input of the program is terminated as a result of ENDATA, and the input file is closed by the program.

All data and control records of the linear programming problem must be placed between these two control records. The input file containing the problem can be created with any editor, which can handle 80 character records in standard character representation. EDLIN, PE for example are of this type.

The input file is divided into data sections, and data types, some of which are optional, others are required. The control records of these data types , which introduce each data section, are the following:

ROWS	required
COLUMNS	required
RHS	required
RANGES	optional
BOUNDS	optional

The description and function of each data type are given in the following sections.

2.3. Constraints, Matrix Rows, ROWS

The ROWS control record introduces the specification of the row names (or constraint names) assigned to the rows of the constraint matrix of the linear programming problem. It also introduces the specification of the types of the constraints in the input file.

Description of the control record:

position 1 - 4
ROWS

Description of the data record:

Field 1	Field 2	Field 3 Field 4 Field 5 Field 6
Type of the constraint	name of the row	empty, or contains a comment ($ sign)

Description of the fields:

Field 1 - defines the type of the constraint row, which can be the following:

'N' unrestricted row (the objective function for instance);

'G' greater than or equal constraint,

```
 n
SUM a(i,j)*x(j)>=b(i);
j=1
```

'L' less than or equal constraint

```
 n
SUM a(i,j)*x(j)<=b(i);
j=1
```

'E' equality constraint,

```
 n
SUM a(i,j)*x(j)=b(i);
j=1
```

Field 2 - name of the constraint row.

Remarks:

1. The IBM MPS input system has a 'D' constraint type for rows obtained by linear combination, the MILP MPS input does not allow this option. If the program finds a 'D' type code in the input data set, it gives a warning (see the section Warnings), but assigns the 'N' (non-restrictive) type code to the row.

2. The IBM MPS input system has a SCALE command that can be used in field 3, MILP ignores this control command without any warning or error message.

2.4. Variables, Matrix Columns, COLUMNS

The COLUMNS control record introduces the specification of the column names of the constraint matrix, and the specification of the values of the matrix entries. The column names identify the structural variables of the LP. The matrix entries are identified by the column name and the rowname.

Description of the control record:

position 1 - 7
COLUMNS

Description of the data record:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
empty	column name	row name1	value1	row name2	value2

optional

Description of the fields:

Field 1 - ignored.
Field 2 - contains the column name, that is the name corresponding to the column of the matrix entry given in the record. If this field contains only space (blank) characters, then the previous column name is considered as column name by the program.
Field 3 - contains a row name: the name corresponding to the row of the matrix entry given in Field 4.
Field 4 - contains a matrix entry value; the value of the matrix entry defined in fields 2 and 3. (see the fifth remark in the first section regarding its format)
Field 5 - same as field 3, but optional.
Field 6 - same as field 4, but optional.

Remarks:

1. The matrix entries must be given at the same time as the columns. All non zero entries must be given for a column before the definition of the matrix can be continued with another column. The order of the entries within a column is arbitrary. Zero entries do not have to be given, since the program assumes zero if nothing is given as matrix entry.

2. The IBM MPS input has a SCALE option, which is not defined in the MILP input. The program gives an error message if one is encountered.

2.5. Right-hand-sides, RHS

The RHS control record introduces the data of the right-hand-side or right-hand-sides corresponding to the constraints in the data set. The right-hand-side(s) are to be considered as column(s), and everything described at COLUMNS is valid.

Description of the control record:

position 1 - 4	or	position 1 - 5
RHS		RHS's

The two formats are equivalent.

Description of the data record:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
empty	RHS name	row name1	value1	row name2	value2

optional

Description of the fields:

Field 1 - ignored.

Field 2 - contains the right-hand-side (RHS) column name, that is the name corresponding to the right-hand-side to which the matrix entry given in the record belongs. If this field contains only space (blank) characters, then the previous right-hand-side name is considered as right-hand-side name by the program.

Field 3 - contains a row name: the name corresponding to the row of the right-hand-side entry given in field 4.

Field 4 - contains a right-hand-side entry value; the value of the right-hand-side entry defined in fields 2, and 3. (see the fifth remark in the first section regarding its format)

Field 5 - same as field 3, but optional.

Field 6 - same as field 4, but optional.

Remarks:

1. The right-hand-side entries must be given at the same time as the right-hand-side names. All non zero entries must be given for a right-hand-side before the definition of the right-hand-sides can be continued with another right-hand-side. The order of the entries within a right-hand-side is arbitrary. Zero entries do not have to be given, since the program assumes zero if nothing is given as entry.

2. The IBM MPS input has a SCALE option, which is not defined in the MILP input. The program gives an error message if one is encountered.

2.6. Ranges, Bounds of the Constraints, RANGES

If a constraint row has both a lower and an upper bound, then the row is of the range type. In this case, the data of the range corresponding to the constraint must be given in this data section. If there is no range type constraint in the linear programming problem, then the RANGE input section can be omitted from the data set together with the control record.

There are three type codes that can be used at the definition of the rows, which make it possible to declare a row to be of the range type. The fact that some number is a lower bound or an upper bound of a constraint depends on the sign of the range value in addition to the type code.

The following table shows how the constraint is interpreted in function of the type code and the value of the range.

b - denotes the value of the right-hand-side
r - denotes the value of the range

type code	range value	lower bound of the row	upper bound of the row
G	r	b	b + abs(r)
L	r	b - abs(r)	b
E	r > 0	b	b + r
E	r < 0	b - abs(r)	b

The ranges as data are considered to be columns whose names must be given by the user, similarly to the right-hand-sides.

Description of the control record:

```
position  1 - 6  |
-----------------|
          RANGES |
```

Description of the data record:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
empty	range name	row name1	range value1	row name2	range value2

optional

Description of the fields:

Field 1 - ignored by the program.
Field 2 - range name, identifier of the range column;
Field 3 - row name. A range type row can only be G, L, E.
Field 4 - value of the range (r). If the value of the range is negative (r < 0), then the program sends a warning, and uses the value r=abs(r) in further calculations.
Field 5 - same as field 3, but optional.
Field 6 - same as field 4, but optional.

2.7. Bounds of the variables, BOUNDS

The type and domain of definition of the structural variables can be specified in this data section. If there is no other declaration, the program assumes that the domain of definition is the domain between 0 and +oo (infinity) ('PL' type). If there is no variable whose domain of definition is different from the latter, then the whole BOUNDS data section can be omitted from the data set together with the control record.

Description of the control record:

position 1 - 6
BOUNDS

Description of the data record:

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
type of the bound	name of the bound vector	column name	value of the bound	empty	empty

Description of the fields:

Field 1 - it can contain the following bound specifications:

LO - lower bound of the variable, l(i)

UP - upper bound of the variable, u(i)

FX - the variable has a fixed value, l(i) = u(i)

FR - the variable is not bounded (free), it can have values from minus infinity to plus infinity, l(i)=-oo, u(i)=+oo

MI - the lower bound of the variable is minus infinity, (its upper bound is zero if not specified), l(i)=-oo, u(i)=0

PL - the upper bound of the variable is plus infinity, (its lower bound is zero if not specified), l(i)=0, u(i)=+oo

Field 2 - identifies the bound vector.

Field 3 - column name, identifies the structural variable whose domain of definition is being bounded.

Field 4 - value; this field contains the lower bound value, or the upper bound value, or the fixed value depending on the type specification LO, UP or FX. In any other case, the value read here is ignored by the program.

Remarks:

1. If the program finds an MI type code for the type of a variable, then it sends a warning, and transforms the variable to PL type using transformations. (see the section on Warning messages)

2. The program checks if the values given are realistic, and if the lower bound for some variable is greater than the upper bound, then the program gives an error message during the input, and the optimization is stopped.

3. The user can specify any number of bound vectors in the input data set, but the bound rows cannot be intermixed during the specification. A new bound name can only occur when all the data belonging to the previous bound have been specified.

4. If a bound is specified several times within the same bound vector in the input data set, the program always takes into account the very last specification.

5. Since no more than one bound can be specified in one record, if we want to specify both a lower and an upper bound, then they must be specified in separate records, since the value found in the record modifies only the bound to which the code corresponds. The following domains of definition are specified with two records:

$$l(i) \le x(i) \le u(i) \quad \text{(LO and UP records)}$$
$$l(i) \le x(i) \le +\infty \quad \text{(LO and PL records)}$$
$$-\infty \le x(i) \le u(i) \quad \text{(MI and UP records)}$$

6. If the lower bound coincides with the upper bound, $l(i)=u(i)$, the input program transforms the variable to FX type, and fixes the value of the variable at $l(i)=u(i)$ value.

3. Example For a Linear Programming Problem

```
NAME           OHTEST53 (ORIGINAL)
*                TEST PROBLEM
ROWS
 N  OBJ.FUNCTION
 L  SEC.ROW
 L  THIRD
 L  FOUR
 L  FIVE        $ END OF NAME LIST
COLUMNS
    COL1        OBJ.FUNC    5.4          SEC.ROW    0.5
    COL1        THIRD       0.25         FOUR       -1.0
    COL1        FIVE        1.0
    COL2        OBJ.FUNC    7.3          THIRD      0.5
    COL2        FOUR        1.0
    COL3        OBJ.FUNC    12.96        SEC.ROW    0.6
                THIRD       0.6
    COL4        OBJ.FUNC    -6.0         SEC.ROW    -1.0
    COL5        OBJ.FUNC    -9.0         THIRD      -1.0
RHS
    RHS1        OBJ.FUNC    800.0        SEC.ROW    80.0
                THIRD       40.0         FIVE       150.0
RANGES
    RANGE1      FIVE        50.0
BOUNDS
 PL BOUND1      COL1                     $ POSITVE INFINITY
 PL BOUND1      COL2
 PL BOUND1      COL3
 UP BOUND1      COL4        20.0
 UP BOUND1      COL5        10.0
ENDATA
```

4. Warning Messages

Objective function name not found!
Assumed the first row as objective function.

The objective function name obtained from the keyboard is not present among the row names in the data set. In the following, the program considers the first row name as the objective function, and continues the input.

Record *****.
Incompatible characters in the relation code field!
Assumed 'N' as relation code.

The program writes out the serial number of the record where it could not interpret the relation code. The relation code in the record is not 'N','E','L','G'. The reading of the input data set is continued, but the program assumes the 'N' relation code for the relation code that could not be interpreted. (The MPS input system of the MILP package does not interpret the 'Dx' type relation codes of the IBM MPS input system, since linear combination of rows is not permitted here.)

Record *****.
Incompatible characters in the bound type field!
Assumed 'PL' as bound type.

The program writes out the serial number of the record, where it could not interpret the type code of the domain of definition of the variable. The bound type in the record is not 'LO','UP','FX','FR','MI','PL'. The reading of the input data set is continued, but the program assumes the 'PL' type code for the bound type code that could not be interpreted.

Record *****.
Bound type 'MI' found! Certain transformations are executed.

The program found an 'MI' type code during the reading of the bound type codes of the input data set. The program transforms the domain of definition of the variable to 'PL' type with the help of a problem transformation. The new bounds of the variable are the following:

LO <= x(i) <= +oo

The run continues with these bounds. (We note, that the optimization gives a meaningful result, the optimum will be the same as in the original problem, only the value of this variable will have an opposite sign at the output of the solution.)

Record *****.
Value of the range is negative!
The RHS value is transformed.

The program found a negative value during the reading of the ranges corresponding to rows. The program considers as value of the range the absolute value of the value of the range read, and continues the input.

Record *****.
Relation code 'Dx' not allowed!
Assumed 'N' as relation code.

The program found a 'Dx' type code among the relation codes, which is not permitted in this input system, since linear combinations of rows are not handled by the input system. The program assumes the 'N' relation code, and continues the input.

5. Error Messages, ERROR's

? Error! Right Hand Side name not found!

The right-hand-side name obtained from the keyboard does not occur in the data set. The checking of the input data set can continue, but the optimization cannot be started.

? Error! Range name not found!

The range name obtained from the keyboard does not occur in the data set. The checking of the input data set can continue, but the optimization cannot be started.

? Error! Bound name not found!

The bound name obtained from the keyboard does not occur in the data set. The checking of the input data set can continue, but the optimization cannot be started.

? Error! Incompatible characters in the numerical field!
Record no.: *****.

The program found a character string in a numerical field, that cannot be interpreted. The serial number of the faulty record is written out. The checking of the input data set can continue, but the optimization cannot be started.

? Error! Row name not found!
Record no.: *****.

During the input of the matrix data (COLUMNS), the right-hand-side (RHS), or the ranges (RANGE), there is reference to a row name which does not occur among the row names. The checking of the input data set can continue, but the optimization cannot be started.

? Error! Column name not found!
Record no.: *****.

During the input of the bounds (BOUNDS), there is reference to a column name which does not occur among the column names (COLUMNS). The checking of the input data set can continue, but the optimization cannot be started.

? Error! Upper bound < lower bound!
Record no.: *****.

During the input of the bounds (BOUNDS), the program found an upper bound value which is less than the lower bound. The checking of the input data set can continue, but the optimization cannot be started.

? Error! Record no.: *****. unrecognizable!

The character string in the record does not correspond to any control row. The record does not begin with the character string

```
'*'
'NAME    '
'ROWS    '
'COLUMNS '
'RHS     '
'RHS'S   '
'RANGES  '
'BOUNDS  '
'ENDATA  '
```

The checking of the input data set can continue, but the optimization cannot be started.

? Error! Too many errors in the input stream!

The program found more than 8 faulty data (errors) during the reading of the input data set. The input is terminated, and the control is returned to the keyboard with the FATAL ERROR ON INPUT FILE message.

6. Functioning of the Input Program

The MPS input program of the MILP program system is started interactively, by simply answering the questions.

The processing of the data set begins with the specification of the input data set.

The input program selects a linear programming problem from the data set which defines a complete family of linear programming problems (it can contain any number of right-hand-sides, bound vectors, and range vectors). An objective function name, a right-hand-side name, a bound name, and a range name must be given for the selection. These names are requested by the program on the screen at the beginning of the processing of the input. Its format is the following:

```
-----------------------------------------------------------
--- "IBM MPS" input format (MPSAD44/F1.01) ---
NAME OF INPUT FILE .......: name of the input data set
OBJECTIVE FUNCTION [FIRST]: name of the objective function
RIGHT HAND SIDE    [FIRST]: name of the right-hand-side
RANGE VECTOR       [FIRST]: name of the range vector
BOUND VECTOR       [FIRST]: name of the bound vector
-----------------------------------------------------------
```

In case we do not give any objective function name, that is we press the RETURN key instead of giving the name, then the first constraint row is considered as objective function.

If we do not give a right-hand-side name, range name, or bound name, then the program considers as right-hand-side vector, range, or bound to be the first right-hand-side vector, the first range vector, and the first bound vector.

7. Completion of the Input Program

After the input, the program writes out the following short statistics on the screen:

```
------------------------------------------------------------------
Number of rows    : *****  number of constraint rows
Number of columns: *****  number of structural variables
Non-zero entries : *****  number of non-zero entries
Input records....: *****  number of records in the data set
------------------------------------------------------------------
```

After this signal, the linear programming problem is ready for optimization, provided there were no fatal data errors in the input data set.

E. Result Reports

After the completion of the program, a short information appears on the screen about the results. The program asks at the same time if we want a detailed result report. If the response is affirmative, it asks the name of the file where to put the results. After the specification of the file name, it prepares the detailed results, which can be consulted using an editor, or can be printed out. The page breaks of the result file are such that the heading is repeated after every 18 lines, thus the names of the quantities can be appropriately followed even in case of consulting through an editor.

The first line gives coded information on the solution. The meaning of the codes is as follows:

1 optimal solution
2 solution unbounded
3 problem is infeasible
4 intermediate feasible solution
5 intermediate infeasible solution
6 intermediate feasible solution, with numerical troubles
7 intermediate infeasible solution, with numerical troubles

In the second line the name and the value of the objective function can be found.

The third line gives the number of variables at infeasible level.

Next, the so called column information appears, which means that a line of information appears about every structural variable with the following contents:

1. Index of the variable (sequence number).
2. Name of the variable (this is empty if the MILP input was used).
3. Status of the variable (B = basic, U = at its upper bound, 'space' = out of the basis, M = infeasible in the negative direction, P = infeasible in the positive direction).
4. Value of the variable.
5. Objective function coefficient.
6. Lower bound.
7. Upper bound.
8. Shadow price.

The column information is followed by the row information. In fact, this means information related to the logical variable belonging to the row, and its contents are the following:

1. Index of the row (preceded by a - sign).
2. Name of the row (empty in case of MILP input).
3. Status of the logical variable.
4. Value of the logical variable.
5. Relation sign of the constraint. (Z = objective function row).
6. Value of the original right-hand-side.
7. Value of the constraint range.
8. Shadow price of the logical variable.

F. Size Restrictions

The actual size restrictions of MILP depend on the parameters given at generation. Presently the maximal version is able to solve problems satisfying the following size restrictions:

number of collective constraints (m) ....<= 1500,
number of structural variables (n) ....<= 6000,
number of non-zero entries
in the constraint matrix plus
number of non-zero entries
in the non-triangular eta vectors ....<= 32000.

G. Use of the Program

The name of the file containing the executable program depends on the parameters used for generating the actual version of MILP.

The first three characters are always MIF.

The fourth character can be N or L. N stands for 'Not large memory', while L stands for 'Large memory'. If N is present than the size of the memory used for storing the matrix file and the "non-triangular" eta vectors does not exceed 64 KByte. In this case for the single precision (S) version nz<=16000, for the double precision (D) version nz<=8000. If the fourth character is L then the upper limit is determined by the available memory.

The fifth character of the name shows if it is a single precision (S) or double precision (D) version of MILP.

The sixth character of the name indicates that the program is not prepared to use the math coprocessor (6), or it is (7).

As a consequence the following program names can exist:

without coprocessor	with coprocessor
MIFNS6.EXE	MIFNS7.EXE
MIFND6.EXE	MIFND7.EXE
MIFLS6.EXE	MIFLS7.EXE
MIFLD6.EXE	MIFLD7.EXE

The EXE files must not be renamed.

The program can be used under [illegible]S operating system.

The memory requirement of the program depends to a large extent on the size of the problem for which it was generated, and on the use of simple or double precision arithmetic. It can be said for indication, that the memory requirement of a system consisting of 300 constraints, 600 variables, and 10000 non-zero entries in case of simple precision is approximately 210 KByte, while the memory requirement of a double precision system consisting of 1500 constraints, 3000 variables, and 32000 non-zero entries is approximately 540 KByte.

The optimization program does not use the disk during the run in general. An exception is the case when a dump is demanded. In this case the default disk drive is used for the time of the dump, and the dump file MILP.DMP is placed there.

The file names given as input for the program may be completely specified (drive, path, file).

The program is started by typing for instance:

```
-------------------------------------------------------
MIFLD7 <return>
-------------------------------------------------------
```

The following steps have already been described, or the program itself gives the necessary information.

A short result report appears after the run, which simultaneously qualifies the result:

OPTIMAL SOLUTION
NO FEASIBLE SOLUTION
SOLUTION UNBOUNDED
PROBLEM UNSOLVABLE WITH PRESENT PARAMETERS
ETA SPACE OVERFLOW

After this, it is possible to write the usual detailed results in a file, from where they can be printed, or consulted using an editor. A dump can be made of the optimal solution if needed. This is done by answering Y to the question

```
------------------------------------------------------------
DO YOU WANT A DUMP OF THE SOLUTION? (Y/N)
------------------------------------------------------------
```

then the name of the file where the dump is to be written must be given as answer to another question.