WORKING PAPER

USER GUIDE TO A MATHEMATICAL PROGRAMMING PACKAGE FOR MULTI-CRITERIA DYNAMIC LINEAR PROBLEMS HYBRID VERSION 3.1

M. Makowski
J. Sosnowski

December 1988
WP-88-111

IIASA
International Institute
for Applied Systems Analysis

USER GUIDE TO A MATHEMATICAL PROGRAMMING PACKAGE FOR MULTI-CRITERIA DYNAMIC LINEAR PROBLEMS HYBRID VERSION 3.1

M. Makowski
J. Sosnowski

December 1988
WP-88-111

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Foreword

This paper is one of the series of 11 Working Papers presenting the software for interactive decision support and software tools for developing decision support systems. These products constitute the outcome of the contracted study agreement between the System and Decision Sciences Program at IIASA and several Polish scientific institutions. The theoretical part of these results is presented in the IIASA Working Paper WP-88-071 entitled *Theory, Software and Testing Examples in Decision Support Systems* which contains the theoretical and methodological bacgrounds of the software systems developed within the project.

This paper presents the HYBRID system for solving linear multiple criteria optimization problems, utilizing the reference point technique. This method, originally developed by the authors, is the non-simplex one and is based on the augmented lagrangian technique with conjugate gradient optimization. Due to special properties of the method it was possible to make the implementation especially efficient for solving dynamic problems. In such cases the HYBRID outperforms such known packages, like MINOS. Since the method does not require to store large amount of information (like basis matrix in standard Simplex formulation) it is especially valuable for microcomputer applications, allowing to solve problems a magnitude bigger dimensionality then with standard methods.

Alexander B. Kurzhanski
Chairman
System and Decision Sciences Program.

Contents

User Guide to a Mathematical Programming Package for Multicriteria Dynamic Linear Problems HYBRID version 3.1

Marek Makowski and Janusz Sosnowski

1. INTRODUCTION

The purpose of this report is to provide sufficient information for using the package. Section 1 contains executive summary, general remarks on solution techniques as well as guidelines for usage of the HYBRID package. Section 2 contains mathematical formulation of various types of problems that can be solved by HYBRID. Section 3 provides information on structure of the software and data. Section 4 gives detailed description of the way in which a user may choose different options provided by the package. This section contains also detailed specification of the format for input data. Section 5 discusses the way in which HYBRID provides diagnostics and results. Section 6 contains short tutorial example. In the Appendix the specification of the MPS standard for input data and an example of the MPS format input file is provided.

This paper does not include information necessary for understanding the mathematical, methodological and theoretical foundations of the HYBRID package. A reader who is interested in those questions should consult the Methodological Guide to HYBRID [1]. The the following topics are disscussed there:

- more detailed discussion of mathematical formulation of various types of problems that can be solved by HYBRID.
- methodological problems related to solution techniques.
- foundations of the chosen solution technique and discussion the computational algorithm.
- short discussion of testing examples.

This report contains updated information provided in the previous version of the User Guide [2] and essential part of the introduction to the Methodological Guide [1]. The latter part is provided for users who are not interested in methodological and theoretical foundations therefore it contains only minimum theoretical background required for usage of the package.

1.1. Executive summary

HYBRID is a mathematical programming package which includes all the functions necessary for the solution of linear programming problems. The current version of HYBRID is referred to further on as HYBRID 3.1. HYBRID may be used for solving both static and dynamic LP problems (in fact also for problems with structure more general then the classical formulation of dynamic linear problems). HYBRID may be used for both single- and multi-criteria problems. HYBRID can be used for solving any linear programming problem but it is specially useful for dynamic problems; this covers a wide area of applications of operation researches. HYBRID has been designed more for real-world problems that require scenario analysis than for academic (e.g., randomly generated) problems. Thus HYBRID is oriented towards an interactive mode of operation in which a sequence of problems is to be solved under varying conditions (e.g., different objective functions, reference points, values of constraints or bounds). It offers also many options useful for diagnostic and verification of a problem being solved.

HYBRID is a member of a decision analysis and support system DIDAS family which is designed to support usage of multicriteria optimization tools. HYBRID can be used by an analyst or by a team composed of a decision maker and an analyst or - on last stage of application - by a decision maker alone. In any case we will speak further on about a user of a HYBRID package.

HYBRID can serve as a tool which helps to choose a decision in a complex situation in which many options may and should be examined. Such problems occur in many situations, such as problems of economic planning and analysis, many technological or engineering design problems, problems of environmental control. To illustrate possible range of applications, let us list problems for which the proposed approach either has been or may be applied: planning of agriculture production policy in a decentralized economy (both for governmental agency and for production units), flood control in a watershed, planning formation and utilization of water resources in an agricultural region, scheduling irrigation, planning and design of purification plant system for water or air pollution. Also many optimization problems in economic planning over time, production scheduling, inventory, transportation, control dynamic systems can be formulated as linear dynamic problems.

To avoid a possible misleading conclusion that the usage of HYBRID may replace a real decision maker, we should stress that HYBRID is designed to help a decision maker to concentrate on real decision making while HYBRID takes care on cumbersome computations and provides information that serves for analysis of consequences of different options or alternatives. A user is expected to define various alternatives or scenarios, changing his preferences and priorities when learning about consequences of possible decisions.

The binary files with HYBRID 3.1 are available from IIASA in two versions: one for VAX 11/780 with Berkeley UNIX 4.2 and one for a PC compatible with IBM/AT.

1.2. Recommended way of usage of the HYBRID package.

1.2.1. Preparation of a problem formulation

A problem to be solved should be defined as a mathematical programming model. Formulation of a mathematical programming problem is a complex task and this paper is not devoted to discuss this question in details. Therefore this section is aimed at providing only a short summary of a recommended approach.

Firstly, a set of variables that sufficiently describe the problem - for the sake of the desired analysis - should be selected. It is desired - however not necessary - to define the problem in such a way as to possibly exploit the problem structure (further on referred to as a dynamic problem). Secondly, a set of constraints which defines a set of admissible (i.e. acceptable or recognized as feasible by a decision maker) solutions should be defined. Finally a set of criteria which could serve for a selection of a solution should be defined.

The formal definition of criteria can be performed in HYBRID in an easy way. However, it should be stressed that any definition of a complex problem usually requires cooperation of a specialist - who knowns the nature and background of the problem to be solved - with a system analyst who can advise on a suitable way of formal definition. It should be clearly pointed out that a proper definition can substantially improve the use of any computational technique. For small problems used for illustration of the method, it is fairly easy to define a problem. But for real life problems, this stage requires a close cooperation between a decision maker and a team of analysts as well as a substantial amount of time and resources.

For real life problems, the following steps are recommended:

1. Mathematical formulation of the problem being solved should be defined.
2. A data base for the problem should be created. This may be done on PC with a help of a suitable commercial product (such as Framework, dBase, Paradox, Oracle). Original data should be placed in this data base. A user need not worry about possible range of quantities (which usually has an impact on computational problems) because HYBRID provides automatic scaling of the problem.
3. Verification of the data base and of the model formal definition should be performed.
4. The corresponding MPS standard file should be created. This may be done by a specialized problem generator (easily written by a system analyst), or an general purpose problem generator or by any appropriate utility program of data base software. We strongly discourage the user from creating the MPS file with help of a standard text editor.

1.2.2. Problem verification

This stage serves for the verification of model definition which is crucial for real application of any mathematical programming approach.

First stage consists of preprocessing the MPS file by HYBRID, which offers many options helpful for that task. HYBRID points to possible sources of inconsistency in model definition. Since this information is self-explaining, details are not discussed here. It is also advisable to examine the model printout by rows and by columns, which helps to verify model specification and may help in tracing possible errors in MPS file generation.

Second stage consists of solving optimization problems for selected criteria which helps in the analysis of consistency of solutions. For larger problems a design and application of a problem oriented report writer is recommended. HYBRID generates a "user_file" for that purpose which contains all information necessary for the analysis of a solution.

After an analysis of a solution, a user may change any of the following parameters: values of coefficients, values of constraints and also any parameters discussed in next section. This may be done with help of the interactive procedure which instead of MPS file uses "communication region" that contains problem formulation processed by HYBRID. Therefore, a user needs no longer to care about original MPS file which has the backup function only.

1.2.3. Problem analysis

Problem analysis consists of consecutive stages:

- analysis of obtained solution
- modification of the problem
- solution of modified problem.

Analysis of a solution consists of following steps (some of which are optional):

1. The user should examine of values of selected criteria. Since the solution obtained in HYBRID is Pareto optimal, the user should not expect improvement in any criteria without worsening some other criteria. But values of each criterion can be mutually compared. It is also possible to compute the best solutions for each criterion separately. A point (in criteria space) composed of best solutions is called the "utopia" point (since usually it is not attainable). HYBRID provides also a point composed of worst values for each criterion. This point is called "nadir" point. Such information help to define a reference point (desired values of criteria) because it is reasonable to expect values of each criterion to lie between utopia and nadir point.
2. The user may also at this stage make modifications to the original problem without involving the MPS file.
3. For dynamic problems, HYBRID allows also for easy examination of trajectories (referred to by so called generic name of a variable).

Modification of the problem may be done in two ways:

1. At this stage, the user can modify the formulation of the original problem. But main activity in this stage is expected after the model is well defined and verified and no longer requires changes in parameters that define the set of admissible (acceptable) solutions. It should be stressed, that each change of this set usually results in change of the set of Pareto-optimal solutions and both utopia and nadir points should be computed again.
2. If the values of all constraints and coefficients that define the admissible set of solutions are accepted, the user should start with computations of utopia point. This can be easily done in an interactive way. After utopia and corresponding nadir points

are obtained (which requires n solutions of the problem, where n is the number of criteria defined) the user can also interactively change any number of the following parameters that define the selection of an efficient solution to the multicriteria problem:

- Reference point (i.e. desired values for each criterion) might be changed. This point may be attainable or non-attainable.
- Weights attached to each criterion can be modified.
- Reference trajectories in dynamic case can be changed as reference points.
- Regularization parameters in selection function can be adjusted.

3. Additionally, the user can temporarily remove a criterion (or a number of criteria) from analysis. This option results in the computation of a Pareto optimal point in respect to remaining "active" criteria but values of criteria that are not active are also available for review.

Solution of a problem. The problem defined by a user (after possible modification) is transformed by HYBRID to an equivalent LP problem which is solved without interaction of a user (an experienced user may however have an access to the information that characterizes the optimization run).

1.2.4. Remarks relevant to dynamic problems.

HYBRID allows for solving both static and dynamic LP problems. Static problems can be interpreted as problems for which a specific structure is not recognized nor exploited. But many real life problems have specific structure which - if exploited - can result not only in much faster execution of optimization runs but also remarkably help in problem definition and interpretation of results.

Numerous problems have dynamic nature and it is natural to take advantage of its proper definition. HYBRID offers many options for dynamic problems, such as:

1. In many situations, the user may deal with generic names of variables. A generic name consists of 6 first characters of a name while 2 last characters corresponds to the period of time. Therefore, the user may for example refer to the entire trajectory (by generic name) or to value of a variable for a specific time period (by full name). Such approach corresponds to a widely used practice of generating trajectories for dynamic problems.
2. The user may select any of 4 types of criteria that correspond to practical applications. Those can be defined for each time period (together with additional "global" conditions), but this requires rather large effort. Therefore, for dynamic problems, criteria are specified just by the type of criterion and the generic name of the corresponding variable. Types of criteria are discussed in details later.
3. A problem can be declared as a dynamic one by the definition of periods of time. For a dynamic problem, additional rules must be observed. These rules correspond to the way in which the MPS file has to be sorted and to the way in which names for rows and columns are selected. These rules follow a widely accepted standard of generation of dynamic problems. The formulation of a dynamic problem, which is accepted

by HYBRID is actually an extension of the classical formulation of dynamic problem (cf Section 2.2.). In this formulation a model may contain also a group of constraints that do not follow the standard of state equations.

1.3. Outline of the solution technique

HYBRID uses a particular implementation of the Lagrange multiplier method for solving linear programming problems. General linear constraints are included within an augmented Lagrangian function. The LP problem is solved by minimizing a sequence of quadratic functions subject to simple constraints (lower and upper bounds). This minimization is achieved by the use of a method which combines the conjugate gradient method and an active constraints strategy.

In recent years many methods oriented for solving dynamic linear problems (DLP) have been developed. Most of those methods consists of adaptation of the simplex method for problems with a special structure of constraints. In HYBRID, a different approach is applied. A DLP, which should be defined together with a state equation, is solved through the use of adjoint equations and by reduction of gradients to control subspaces (more exactly, to a subspace of independent variables). The method exploits the sparseness of the matrix structure. The simple constraints (lower and upper bounds for non–slack variables) for control variables are not violated during optimization and the resulting sequence of multipliers is feasible for the dual problem. The global constraints (i.e constraints other then those defined as simple constraints) may be violated, however, and therefore the algorithm can be started from any point that satisfies the simple constraints.

2. STATEMENT OF OPTIMIZATION PROBLEMS

HYBRID is designed for solving multicriteria dynamic linear optimization problems. However it can be used also for solving both single– and multicriteria problems of either static or dynamic linear problems as well as single criterion linear–quadratic problems. To provide formal definition of any type of problems that can be solved by HYBRID the following approach is adopted. Firstly three types of single criterion problems is summarized – namely static linear programming problem, dynamic linear programming problem and linear quadratic dynamic problem. Secondly different types of criteria that can be applied to both static and dynamic LP problems are discussed.

2.1. Formulation of a single criterion LP problem

We will consider a linear programming problem (P) in the following standard form (see, e.g., [3]):

$$\min \quad cx \tag{2.1}$$

$$b - r \leq Ax \leq b \tag{2.2}$$

$$l \leq x \leq u \tag{2.3}$$

where $x,c,l,u \in R^n$, $b,r \in R^m$ and A is an $m \times n$ matrix.

The constraints are divided into two groups: general constraints (2.2) and simple constraints (2.3). In the input data file (MPS file) the vectors b is called RHS and the vector r – RANGES. The vector l and u are called LOWER and UPPER BOUNDS, respectively. Obviously, some of bounds and/or ranges may have an infinite value. Therefore HYBRID may be used for solving any LP problem formulated in the way accepted by most of commercial packages.

2.2. Formulation of a single criterion Dynamic Linear Problem (DLP)

HYBRID 3.1 is capable to solve Dynamic Linear Optimization Problems (DLP) of more general class then formulated e.g. in [3]. One of main generalizations – from a practical point of view – is that a problem with delays for control variables may be solved by HYBRID. In fact, HYBRID accepts also problems with delays for both state and control variables, provided that state variables for periods "before" initial state do not enter state equations.

All variables are divided into two groups: decision variables u and state variables x_t, the latter are specified for each period of time.

A single criteria DLP problem may be formulated as follows:

Find a trajectory x_t and decision variables u such that both:

state equations:

$$-H_t x_t + \sum_{i=0}^{t-1} A_{t-1,i} x_i + B_t u = c_t, \qquad t=1,...,T \tag{2.9}$$

with given initial condition x_0

and constraints:

$$d - r \leq \sum_{t=0}^{T} F_t x_t + Du \leq d \tag{2.10}$$

$$e \leq u \leq f \tag{2.11}$$

are satisfied and the following function is minimized:

$$\sum_{t=1}^{T} a_t x_t + bu \tag{2.12}$$

Components of vector u are called decision variables for historical reasons. Actually a vector u may be composed of any variables, some of them may be specified for each time period and enter criteria defined for a dynamic case. But some components of vector u may not be specified for any time period. An example of such variable is "..dummy.", a variable generated by HYBRID for a multicriteria problem. A user may also specify variables independent of time. For the sake of keeping the formulation of the problem as simple as possible we have not introduced a separate name for such variables.

The following two symbols can be used in the specification file for definition of DLP:

NT – number of periods (stands for T in the above formulation)

NSTV – number of state variables in each period (the dimension of vectors x_t)

The user can define state inequalities instead of state equations (2.9). The slack variables for such inequalities are generated by HYBRID. Therefore, for the sake of the presentation simplicity, only the state equation will be considered further on.

The structure of an DLP problem may be illustrated by the following diagram:

u	x_0	x_1	x_2	x_3	rhs	var.
B_1	A_{00}	$-H_1$	0	0	c_1	state eq.
B_2	A_{10}	A_{11}	$-H_2$	0	c_2	state eq.
B_3	A_{20}	A_{21}	A_{22}	$-H_3$	c_3	state eq.
D	F_0	F_1	F_2	F_3	d	constr.
b	0	a_1	a_2	a_3	—	goal

where H_t is diagonal matrix and 0 is a matrix composed of zero elements

2.3. Formulation of single criterion linear-quadratic problems.

HYBRID 3.1 allows for solution of a single-criterion linear-quadratic problem with a simple quadratic term. For a problem which does not have recognized structure (as discussed in Sec. 2.2) the formulation takes the following form:

$$\min \quad cx + \gamma/2 \parallel x - \bar{x} \parallel^2$$

subject (2.2) and (2.3), where $\bar{x}$ is a given point in the solution space and $\gamma > 0$ is a given parameter.

Similarly, for a dynamic problem one may formulate the problem in the following way:

$$\min \sum_{t=1}^{T} a_t x_t + bu + \gamma/2 \parallel u - \bar{u} \parallel^2$$

subject (2.9) and (2.11), where $\bar{u}$ is a given point in the space of independent variables.

2.4. Multicriteria optimization

2.4.1. General remarks

The specification of a single-objective function, which adequately reflects preferences of a model user is perhaps the major unresolved difficulty in solving many practical problems as a relevant optimization problem. This issue is even more difficult in the case of collective decision making. Multiobjective optimization approaches make this problem less difficult, particularly if they allow for an interactive redefinition of the problem.

The method adopted in HYBRID 3.1 is the reference point approach introduced by Wierzbicki [4]. Since the method has been described in a series of papers and reports and has been applied to DIDAS (cf [5],[6]), we give only general outline of the approach applied. This approach may be summarized in form of following stages:

1. The user of the model (referred to further as the decision maker – DM) specifies a number of criteria (objectives). For static LP problem a criterion is a linear combination of variables. For DLP problems one may also use other types of criteria (cf sec. 2.4.2). The definition of criteria in HYBRID can be performed in an easy way described in the Section 4.3.2.
2. The DM specifies an aspiration level $\bar{q} = \{\bar{q}_1,....,\bar{q}_{NC}\}$, where $\bar{q}_i$ are desired values for each criterion and NC is a number of criteria. Aspiration level is called also a reference point.
3. The problem is transformed into an auxiliary parametric LP (or DLP) problem. Its solution gives a Pareto-optimal point. If specified aspiration level $\bar{q}$ is not attainable, then the Pareto-optimal point is the nearest (in the sense of a Chebyshev weighted norm) to the aspiration level. If the aspiration level is attainable, then the Pareto-optimal point is uniformly better then $\bar{q}$. Properties of the Pareto-optimal point depend on the localization of the reference point (aspiration level) and on weights associated with criteria.
4. The DM explores various Pareto-optimal points by changing either the aspiration level $\bar{q}$ or/and weights attached to criteria or/and other parameters related to the definition of the multicriteria problem.
5. The procedure described in points 3 and 4 is repeated until satisfactory solution is found.

2.4.2. Types of criteria

A user may define any number of criteria. To facilitate the definition 6 types of criteria are available and a user is requested to declare chosen types of criteria before their actual definition. Two types of criteria are simple linear combination of variables and those criteria may be used for both static and dynamic problems. Four other types of criteria correspond to various possible performance indices often used for dynamic problems. Since the latter criteria implicitly relate to the dynamic nature of the problem, they may be used only for variables that are defined for each time period. The only exception is the type DER of criteria, which may be defined by state variables only.

For the sake of simplicity, only the variables of the type x_i (which otherwise is used in this paper to distinguish a state variable in DLP) are used in the following formulae. Note that $x_i = \{x_{it}\}, \quad t=1,...T$.

An k-th criterion q_k is defined in one of following ways, for static and dynamic LP:

Type MIN

$$q_k = \sum_{t=1}^{T} \sum_{i=1}^{n} a_{it} x_{it} \rightarrow \min$$

where n is number of (state and control) variables, T is number of periods; $T=1$ is assumed for static LP.

Type MAX

$$q_k = \sum_{t=1}^{T} \sum_{i=1}^{n} a_{it} x_{it} \rightarrow \max$$

and exclusively for dynamic LP (where x_i denotes a selected state or control variable, $\bar{x}_i$ - its reference trajectory):

Type SUP

$$q_k = \max_{t=1,..T} (x_{it} - \bar{x}_{it}) \rightarrow \min$$

Type INF

$$q_k = \min_{t=1,..T} (x_{it} - \bar{x}_{it}) \rightarrow \max$$

Type FOL

$$q_k = \max_{t=1,..T} (abs(x_{it} - \bar{x}_{it})) \rightarrow \min$$

Type DER (which applies only to state variables)

$$q_k = \max_{t=1,..T} (abs(x_{it} - x_{it-1})) \rightarrow \min$$

3. STRUCTURE OF THE SOFTWARE AND DATA

3.1. General description of the package

The package is constructed in modules to provide a reasonably high level of flexibility and efficiency. The package consists of five subpackages:

- Two preprocessors that serve to process data, enable a modification of the problem, perform diagnostics and may supply information useful for verification of a problem. The first preprocessor is used for processing of initial formulation and diagnostics of the problem. It also transforms a multicriteria problem to a parametric single criteria optimization problem. The second preprocessor allows for analysis of a solution and for the interactive change of various parameters that may correspond to choice of some option, change of parameters in definition of multicriteria problem, change of matrix coefficients, right hand sides of constraints etc.
- Optimization package called solver of a relevant optimization problem (either static or dynamic)
- Postprocessor that can provide results in the standard MPS format and can also generate the "user file" which contains all information needed for the analysis of a solution; the later option makes it easier to link HYBRID to a specialized report-writer or a graphic package.
- Driver, which ease a usage of all subpackages. The driver of the PC version of provides a context sensitive help which helps an unexperienced user in efficient usage of the package.

All five subpackages use a binary file, that contains all data that define the problem being solved. A second binary file contains a solution obtained by last run of the solver. From the user point of view, HYBRID 3.1 is still one package that may be easily used for different purposes chosen via specification file.

The chosen method of allocating storage in the memory takes maximal advantage of the available computer memory and of the features of typical real-world problems. In general, the matrix of constraints is large and sparse, while the number of all non-zero coefficients that take different numerical values is much smaller than the number of all non-zero coefficients. A super-sparse-matrix technique is therefore applied to store the data that define the problem to be solved. This involves the construction of a table of coefficients which take different numerical values. The memory management is handled by a flexible way. HYBRID is coded partly in C and partly in an extention of Fortran (the latter part is processed by preprocessor to generate a code which conforms to Fortran 77 standard).

3.2. Installation and usage of the package

The mainframe version of HYBRID has been installated on IIASA/VAX. There exists computer readable manual page which provides summary on usage of package. This manual page may also be printed by command *man -tq -h0 hybrid*. To use HYBRID just type *hybrid*. There are few options that may be used for different purposes. To find the meaning of the currently available options type *hybrid -h*.

The PC version of HYBRID may be installed in the following way. Copy all *.exe files from both diskettes to a directory which is included in your path. The following *.exe files should be accesible: hybrid.exe (driver for all HYBRID modules), h0.exe (initial preprocessor), h1.exe (second preprocessor), h2.exe (solver), h3.exe (postprocessor). To use HYBRID package just type: hybrid. The PC version of HYBRID provides context sensitive help (via F1 key). Files _specs _mps and _crit contain a simple example. To run the example place those files into a current directory and invoke hybrid.

In the current directory (which should not be write protected) the input files should be located. There should be enough space left on current disk for writing files (to avoid possible file management problems at least 200kB is required). The driver of PC version checks if enough space on disk is provided and if disk is writeable. For some distribution versions (compiled with in-line instruction for 8087/80287 and linked wtih 8087/80287 library) driver checks if math coprocessor is installed. If those requirements are not fulfilled, the program is aborted.

3.3. General description of the options provided by HYBRID

In order to provide general information about capabilities of HYBRID, the main options are listed below. HYBRID offers the following features:

- Input of data and the formulation of an LP problem follow the MPS standard. Additional rules (that concern only sequencing of some rows and columns) should be observed in order to take advantage of the structure of a dynamic problem. An experienced user may speed up computations by setting certain options and/or parameters.
- The problem can be modified at any stage of its solution (i.e., by changing the matrix of coefficients, introducing or altering right-hand sides, ranges or bounds).
- The multicriteria problem is formulated and solved as a sequence of parametric optimization problems modified in interactive way upon analysis of previous results.
- The solution technique can be chosen. First choice is done by definition of a static or a dynamic problem. Some specialized techniques may be used for problems that may cause numerical problems. This includes one of two regularization techniques and/or possibility of using preconditioned conjugate gradient method (cf the Methodological Guide [1]). For a badly scaled problem an implementation of scaling (as described by the authors in [7]) algorithm is available.
- A comprehensive diagnostics is implemented, including the checking of parallel rows, the detection of columns and rows which are empty or contain only one entry, the splitting of columns, the recognition of inconsistencies in right-hand sides, ranges and bounds, and various other features that are useful in debugging the problem formulation. The package supports a display of a matrix by rows (printing the nonzero elements and names of the corresponding columns, right-hand sides and ranges), as well as a display of a matrix by columns (analogous to displaying by rows). A check of the feasibility of a problem prior to its optimization is optionally performed. More detailed information for an infeasible or unbounded problem is optionally provided by the package.

- All data that correspond to the formulation of the problem being solved are stored in a binary file. An other binary file contains all other information corresponding to a current run. The latter file is stored on disk in certain situations to allow continuation of computations from failed (or interrupted) runs or to run a modified problem while using previously obtained information. Therefore the MPS input file is read and processed only by first preprocessor, which serves for initial formulation of the problem. Such approach allows also for efficient storing of many solutions that may be later used for more detailed analysis, comparisons and modifications.
- Any solution is also available in the standard MPS format and optionally in a binary file which contains all data that might be useful for postoptimal analysis and reports.

3.4. Structure of the data

All necessary input files should be placed in a default directory. In addition to the files provided by a user (for initial run) HYBRID uses files that are generated during previous runs. Therefore a user is adviced to select a directory in which he will work using HYBRID and to back-up generated files that contain valuable information. PC version of HYBRID provides an option which allows for storing the status of the problem for later modification (a user may attach a comment to this file which eases the problem of later choice out of possibly many files which contain different modification of the problem.

3.4.1. Input files

_specs	Specification of a problem to be solved
_mps	MPS data file (needed for initial run only)
_modif	Inputs data for modification of the problem (for mainframe version only)
_crit	Definition of criteria for multicriteria optimization (needed for initial run only)
stdin	Standard input is used for interaction with the package

3.4.2. Files generated by HYBRID

stdout	Standard output is used for interaction with a user
_log.ini	File that contains initial diagnostics and statistics of the problem.
_log	File that contains diagnostics of the latest run of the problem.
_comm	File that contains constant part of the communication region (generated by a previous run of the problem being solved)

__sol	File that contains a variable communication region (generated by a previous run of the problem being solved). Contents of this part depends on the current status of the problem. A user of the PC version of HYBRID may attach a comment and store this content at any stage of modification process. The stored file may be later used (the driver provides relevant information about content of the file); names of those files are composed in the following way: root name is equal to the problem name and extention is given by a user;
__back	A file used interchangeably (with file __sol) to secure the contents of the communication region
__solpr	Results of optimization in MPS standard format
__userf	Results of optimization in a binary file of random access (cf sec. 5.4)
__byrows	Display a matrix by rows
__bycols	Display a matrix by columns
__mpso	output of MPS–format file (after transformation of a multicriteria problem to a corresponding single criterion one and after possible modification).

4. USER-SUPPLIED INFORMATION

4.1. Overview

The user can supply information of three types:

- the problem specification
- the formulation of the problem
- modifications to the problem being solved

Problem specification is optional. If the specification file is empty all of the control statements take their default values. Problem specification is discussed in more detail in Section 4.2.

The formulation of the problem is necessary for the initial run (cold start) but not for subsequent or modification runs. Problem formulation consists of two parts. Firstly, one defines a problem as an LP (additional requirements apply for DLP) without multicriteria part. Secondly, one defines (if needed) multicriteria part. Problem formulation is discussed in more detail in Section 4.3.

The problem may be modified on either an initial run or a recovery run (after finding an optimal solution, in the case of an infeasible/unbounded problem or following an interrupted run). The way in which the problem may be modified is discussed in more detail in Section 4.4.

4.2. Problem specification

The user specifies the problem and may control some of the operations performed by HYBRID with the help of the specification file containing the control statements. The definitions and default values of these statements are given in the following subsections.

Statements may be given in any order. Note that each new value for a given control statement will overwrite the previous one (either the default value or the value restored from a recovery file or previously defined in the same specification file) without any specific warning.

A statement in the specification file is recognized by the first four characters of the keyword and – if required – by a parameter following the keyword. The keywords are given in full in Sections 4.1 through 4.6 and may be used in this form for the sake of clarity. Each statement should be specified in free format on a separate line. Only the first 30 characters are processed. Blank(s) are used to separate the keyword and its parameter, and therefore blanks cannot be embedded in either the keyword or the parameters. The last column (i.e., the 30th) must contain a blank.

The specification file is read from the file *__specs* until a star (*) is encountered in the first column or EOF (end of file) is reached. The user may control printing of input stream by placing the PRINT or NOPRINT statement in the specification file. Each statement is checked for validity and error messages are printed if a statement is incorrect. If the number of errors occurring during the processing of the specification file reaches 30 the run is terminated. Any error detected during processing causes termination of the run after the specification file has been processed. The diagnostics are printed on file *__log.ini* or *__log*. More severe errors are reported on *stderr*. If no error occurs and

the PRINT directive is not in effect, no information is issued. In any case, the current values of all control statements are listed in the diagnostics file.

A line that contains the character "c" in the first column followed by a blank in the second column is treated as a comment and ignored. There is no restriction on the contents of the remaining columns.

A control parameter is therefore defined by a default value, by a value restored from a recovery file, or by a statement in the specification file. The values of the parameters can also be overwritten in this sequence, i.e., a default value is overwritten by a value from a recovery file, which may itself be overwritten by a statement in the specification file.

4.2.1. General description of control commands

The sequence of operations executed by HYBRID is controlled by the user through commands provided in a specification file. Some of these control commands are listed below. It is recommended that only the commands mentioned here should be redefined by the user: as yet there is insufficient information on the effects of changing the values of other parameters or options. The authors of the package hope to formulate guidelines governing the modification of these additional parameters/options in due course.

A control statement activates or deactivates a certain option, defines or redefines the logical number of an input/output unit, or sets the value of a parameter. Each statement has a default value which is initialized prior to starting the run. These default values are also given below.

The control commands are divided into five groups:

1. Commands without parameters
2. Commands with character string parameters
3. Commands with integer parameters
4. Commands with real parameters
5. Commands with a single parameter which may be either a character string or a real number

Each group of commands is discussed separately below.

4.2.2. Commands without parameters

Each statement of this type activates or deactivates a certain action, and therefore they are listed in pairs. The first of each pair of listed options is the default one.

MAXIMIZE MINIMIZE	Defines type of optimization of the objective function. This option is overwritten if multicriteria optimization is performed
NOMULTI MULTI	To activate the subpackage for multicriteria optimization (definition and modification)

SCALE NOSCALE	To activate the scaling routine
NOMODIFY MODIFY	To activate the routine for problem modification
NOBROWS BYROWS	To display the matrix by rows
NOBCOLS BYCOLS	To display the matrix by columns
NOMPS MPSOUT	To output the problem being solved in MPS–format
NOACCEPT ACCEPT	To allow minor errors in the MPS file (such as zero elements, duplicated elements, etc.). Such errors are reported but do not cause termination of the run prior to optimization if the ACCEPT option is set
NOREGUL REGZERO	To regularize the problem (see Methodological Guide).
NOGETFEAS GETFEAS	To check feasibility prior to optimization. This action should be avoided for problems likely to have a feasible solution because its use increases the total number of iterations.
NOPARALLEL PARALLEL	To check for parallel rows. This option is time- consuming but helps to identify dominating rows and pairs of constraints that may be replaced by a single constraint with appropriate range.
COMMULT ZERMULT	Sets the initial value of the Lagrange multipliers to zero. The alternative is to compute the initial multipliers before the first iteration,but this requires activation of the SPIRIT routine.
NOSPIR SPIRIT	Activates a specialized routine for computation of a starting point.
NOPRE PREC	Vector preconditioning
NOPRE PREM	Matrix preconditioning

Declaration of a dynamic problem is implicitly done by NT parameter (cf sec. 4.2.4).

4.2.3. Commands with character string parameters

GOAL	Name of the neutral row taken as the objective function. If absent, the neutral row encountered first is assumed to be the objective function. The name is overwritten if multicriteria optimization is performed.
RHS	Name of the set of right-hand sides and ranges. If absent, the first such name encountered is taken
BOUNDS	Name of the bounds set. If absent, the first such name encountered is taken.
NAME	Name of the problem. If absent, the name found in the MPS file is assumed.

4.2.4. Commands with integer parameters

MROWS	100	Maximum number of rows
MCOLS	5*MROWS	Maximum number of columns
MELEM	5*MCOLS	Maximum number of nonzero matrix elements
MDIFF	MELEM	Maximum number of different quantities defining the problem
MTIME	10	Number of CPU minutes allocated for the run
MITER	-	Maximum number of iterations (cf Section 4.5)
MERRORS	50	Maximum number of errors allowed on the MPS file before processing is terminated)
ITSCAL	30	Maximum number of iterations during scaling
ISECURE	500	Number of iterations after which the communication region is stored (in addition to secure action after each update of multipliers and termination of the run)
ITLOG	0	Level of information detail issued during optimization: 0 causes information after each update of multiplier, positive value n gives information every n steps in minimization of augmented Lagrangian function
INORM	1	L_∞ norm is assumed in the stopping criterion. For L_2 norm, 0 should be specified
IFEAS	2	Feasibility is checked if doubtful (i.e. some conditions hold - cf description of multiplier method sec. 3.4.). To check feasibility after a first update of multiplier IFEAS 1 should be stated. To force check of feasibility before entering optimization - GETF option (cf sec. 4.2.) may be used.
NT	0	Number of periods for a dynamic problem
NSTV	0	Number of state variables for a dynamic problem
MAXCRIT	10	Maximum number of criteria

Numbers or expressions given above correspond to the default values.

Note that NT=0 implies a static problem. For a dynamic problem NT should be greater then 1. For a static problem NSTV should be equal to 0.

4.2.5. Commands with real parameters

BIGN	1.e+30	Any number greater then BIGN is treated as infinite
TZERO	2.22e-16	Any number of absolute value less then TZERO is replaced by 0
SMALL	1.e-6	Any number in the result file with absolute value less then SMALL is replaced by 0.
FEAS	1.e-5	Feasibility tolerance
SETA	.5	Parameters for scaling
SBETA	.5	
SEPS	.985	
SEP1	.01	
RO	1.	Penalty parameters for Lagrangian
ROST	2.	
ROS2	4.	
ROMX	512.	
RETA	0.	Regularization parameters
RMET	0.	
RSET	0.	
EPS	0.	Stopping-criteria parameters
EPSD	FEAS	
EPSS	0.	
EPSM	1.5e-8	Maximum accuracy for minimizing of the augmented Lagrangian penalty function
MEPS	.01	Parameter of achievement scalarizing function (cf Methological Guide)
QCF	0.0	Weight parameter in definition of linear quadratic problem (cf Section 2.3 - QCF corresponds to the coefficient γ).

The parameters with values equal to zero can be changed by a user but this is not recommended. If a user does not change them, they will be computed according to the rules given in sec. 4.2.7. Some of the above listed parameters define tolerance which in turn should be consistent with computer machine precision (e.g. for IBM PC with math coprocessor the precision of double precision real number is equal to 2^{-52}.

4.2.6. Commands with mixed parameters

Both of these commands take either a real-valued parameter or the word NONE. The commands are used to define the lower and upper bounds for variables. This may be changed for selected variables through appropriate definitions in the BOUNDS section of the MPS file. The default values are the following:

LBOUND	0.
UBOUND	NONE

4.2.7. Remarks on resetting default parameters

Various parameters occur in the algorithms presented in the preceding two sections. Most of them play an important role and have to be chosen very carefully. Moreover, the values of some of these parameters are (or should be) interrelated.

The values of any of the parameters may be reset by an advanced user. If this is done, the PARAM procedure checks only whether the parameter meets certain general requirements, e.g., that it is positive. Thus the user should be very careful when making changes in parameters that affect tolerances.

Some parameters have a non-zero default value. This is generally the case for parameters that do not depend on the problem being solved. If the user specifies an unacceptable value for such a parameter, the default value is restored.

Other parameters default to an initial value of zero; the parameter values are then recomputed according to the rules given below as the program proceeds. If a user specifies a non-zero initial value which becomes unacceptable during the course of the calculation, the values computed from the following rules are restored:

RETA = 1./abs(AMXMAT), where
RETA is the initial regularization parameter,
AMXMAT is the largest of the matrix elements.

RSETA = ROST**2, where
RSETA is the coefficient for increasing RETA.

RMETA = 1.e+4/sqrt(FEAS), where
RMETA is the maximum value of the regularization parameter.

MITER = 2*(N+M),where
MITER is the maximum number of multiplier iterations,
N is the number of variables,
M is the number of constraints.

MSITER = (M+N)*N, where
MSITER is the maximum number of iterations during minimization of the augmented Lagrangian.

4.3. Formulation of the problem

4.3.1. Preparation of input data file

At present a problem to be solved has to be presented in standard MPS format (cf Appendix); this may be done using a commercial problem generator or a generator tailored specifically to the problem. This does not apply however to specification of its multicriteria part which may be done in an easy way (cf sec. 4.3.2.). We shall therefore make only a few general suggestions and comments on this part of the system. Additional requirements for structure of MPS file apply for DLP (cf sec. 2.2).

The names of rows and columns should start with a letter or a number to avoid possible confusion with names generated by the package for multiobjective optimization problems. Names should not include a blank because of the syntax rules used in the modification routine.

Any line in the MPS file may contain a star (*) in the first column. Any such line is treated as a comment and there are no restrictions on the contents of the remaining columns.

We recommend that lower and upper bounds should be specified for all these variables and in these constraints whenever sensible values are known. This is useful in defining the admissible region over which optimization is to be performed and usually results in a decrease in computation time.

Since the computer code (if used for a DLP problem) for the algorithm applied in HYBRID is based on the structure of the problem, the proper formulation of a dynamic problem being solved is critical. Therefore, there are some restrictions concerning the form of MPS file for DLP. Those restriction which enable also diagnostic of a problem formulation, and are as follows:

1. All names for rows that correspond to state equations and columns that correspond to control and state variables should have exactly 8 characters. First six characters define so called generic name of a variable, whereas last two characters identify the period. Each name should start with a letter or with a number. Use of embedded blanks is not allowed because it would be in conflict with modification routine. By control variables we understand any variable that is defined for each time period and is not a state variable. Therefore a variable that is not defined for each time period may enter also a state equation and may have any name but such a variable should not be used for definition of a dynamic type criterion.
2. A name of a state equation row should be the same as a name of a state variable defined by that row.
3. State equations should be first in ROWS section and they should be sorted by periods. Arrangement of rows defining state equations should be the same for all periods.
4. Columns are divided into two groups. First control (decision) variables should be specified in any arrangement. Secondly, the state variables should be specified, sorted by periods (i.e. first all state variables for first period). The arrangement of state variables in all periods should be the same.

5. Initial conditions for state equations should be specified in BOUNDS section (by fixing state variable corresponding to period number 0). Other constraints for state variables (if any) should be specified (preferably with use of ranges) in column section. The package removes any constraints for state variables (for all periods except initial) from BOUNDS section. If a constraint is needed for a state variable it should be specified as a general–type constraint (cf sec.2.2. and 2.3.).

A reasonable scaling of a problem is very important for numerical reliability. It is generally recommended that data and variables values should be as close to 1.0 as possible. However, since an automatic scaling is performed by HYBRID the user needs not to be very careful when scaling. The only requirement is that care should be taken in the formulation of the problem to ensure that only significant variables are included in the constraints formulation. This requirement is easily fulfilled for real-world problems. Since HYBRID provides the scaling option the user need not worry about differences in the magnitude of the coefficients.

4.3.2. Specification of multicriteria problem

A specification of a multicriteria problem has to be done in two parts. First part consists of a declarations of all criteria. The second on gives definitions of criteria.

A criterion is declared by specification of its name (four characters) and a type acronym given above with each criterion type. For MIN and MAX types additionally an overestimate of a number of linear combination components should also be given (if absent, the default value 10 is assumed). Each criterion should be declared on separate line (card image), * character in first column finishes declaration of criteria. Only 37 columns (for each card image) are processed. Should a criterion name be shorter then 4 characters, periods are added up to four characters. There are no restrictions on mixing various types of criteria.

Definition of criteria should follow declaration. Criteria may be defined in any order. Input stream (80 columns cards images) is processed until * character in first column or EOF is reached.

Each card image is assumed to contain at least two fields. Fields are separated by at least one blank.

First field should contain a criterion name or / (a continuation mark). Note that continuation may be required only for types MIN or MAX.

For types MIN and MAX second (and possibly further) field(s) contain a real number followed by * and by a name of variable (no embedded blanks allowed). A number corresponds to a coefficient associated with the specified variable in linear combination that defines given criterion.

For types SUP, INF, FOL and DER a second field contains a name, which is the name of variable that is associated with a criterion being defined. The name required is so called generic name (cf sec.1.2.4.) therefore should be composed of exactly 6 characters.

Reference trajectories (if required) should be specified in a way described in the following sections.

Note that both declaration and definition of criteria should be made in so called cold start (initial), whereas parameters (described further on) may be changed in both cold start and subsequent computations.

4.4. Modification of the problem

The PC version of HYBRID provides means for user friendly interfaces that serves the modification of the problem. A context sensitive help is provided, therefore there is no need to describe the way of performing modification. The following subsections provides information on modification options for the batch version of HYBRID.

4.4.1. Modification of the matrix, RHS, RANGES, BOUNDS

A user may interactively modify the problem being solved by activating the modification routine. This is activated by inserting the keyword MODIFY in the specification file (either during first or subsequent runs).

The modification lines should follow the MPS standard with the following exceptions:

1. Data are read in free format, and therefore there is no need to worry about placing data in the fields prescribed by the MPS standard.
2. Sections may occur in any order, and may also be subdivided.
3. Only 37 columns (in each card image) are processed.
4. Due to the problem of repacking the data (which has not yet been completely overcome), reclassifying a row or introducing new non-zero elements in the matrix is not allowed.
5. To remove a range the names of the rows affected should be specified with value 0. in the ranges section. Negative values are however illegal.

The data which are to be modified are read from a file *__modif* until a star (*) is found in the first column or EOF (end of file) is reached. The user may specify PRINT and NOPRINT commands which cause echoing of modification cards to standard output and suppress echoing, respectively.

Any user who does not want to follow the format restrictions imposed by the MPS standard should instead observe the following syntax rules:

1. Section names should follow MPS format.
2. Lines (with the exception of section names, comments, PRINT and NOPRINT commands and the star character that serves as an optional EOF mark) should have a blank in the first and 37th columns.
3. Fields are separated by at least one blank.
4. The number and contents of all fields must correspond to the information required by the modified section.
5. A line is treated as a comment if it contains the character "c" in the first column followed by a blank in the second column.

Since it is assumed that the sets of bounds and right-hand sides have been chosen, no associated name is needed. Thus, in the modification lines the corresponding fields should contain blanks (if prepared according to MPS format) or be absent (if in free format).

In addition to possible error diagnostics, other information is printed during modification.

Processing may be terminated if the number of errors detected during modification exceeds MERRORS (see Section 4.2.4).

4.4.2. Modification of multicriteria problem parameters

A user may change parameters of multicriteria problem in interactive way. To facilitate this task the same routine displays also information about last solution (cf sec. 5.3.).

The routine recognizes following commands (only first character from terminal input is processed)

HELP	displays list of commands
VERB	set verbose mode of interaction; results in comments for supposed action of a user; this is the default mode
NVERB	set non-verbose mode, which result is replacing comments by acronyms
LIST	lists of information about obtained solution.
STATUS	a user may temporary remove a criterion from achievement function; the status is stored for subsequent runs, but criterion may be restored by the same command; i.e. specification of a criterion name change status from active to non-active or vice versa
UTOPIA	a user may look for a utopia point for selected criterion; this results in setting status for all other criteria as non-active
RFP	change of reference point (aspiration level)
WEIGHT	change of weight coefficients; weight coefficients are normalized in such a way, that sum of weights is equal to number of active criteria; default values for weights are 1.
MEPS	change of value of ϵ_m coefficient
TRAJ	change of reference trajectories; one may also display values of the respective trajectory. To display a trajectory that does not enter any criterion, instead of a criterion name .... (four periods) should be entered and, next, a trajectory name should be specified.
COEF	change of coefficients in linear combination defining criteria MAX and MIN
END	exit modification status

All options that allow change of respective parameters, offer also a possibility of list of values of those parameters. An interactive way of changing parameters is fairly easy and - since a user may be guided in verbose mode - there is no need to include more details in this report.

5. HYBRID-GENERATED INFORMATION

5.1. Initial information and problem diagnostics

The information generated may be divided into the following classes:

1. If the recovery option is activated, information about the recovery file (name of problem, date and time of creation, status of solution, size, etc.) is printed.
2. A summary of the current values of all control statements and parameters is printed.
3. On the occasion of a cold start, the input of the MPS file is reported. Error diagnostics and warnings are also issued, if applicable. This information should be self-explanatory, and therefore is not included in the examples presented in the Appendices.
4. For a multicriteria problem parameters of multicriteria definition and solution are reported. Interactive process of changing of parameters is also reported. Additionally, for a cold start only, definitions of criteria are printed.
5. If the modification option is activated the relevant information and possibly some diagnostics are provided.
6. A summary of input data and problem statistics is printed.
7. If a user overestimates the core required, a reallocation procedure is called and a report is printed.
8. If scaling is performed, this is reported.
9. The values of the parameters set by the PARAM procedure are reported.

The storage allocation information issued after the problem has been set up refers to two parts of the communication region:

1. The fixed part (for a given version of HYBRID), which contains the values of all the control statements.
2. The working area, which contains the rest of the information and the data for the problem being solved.

Additional information may be placed in different files.

5.2. Information generated during optimization

Information may be provided at two levels of detail. The user may change the level by ITLOG option. The default setting (ITLOG 0) causes issuing information every time the multipliers are updated. The alternative (ITLOG n) is to print information every n steps in the augmented Lagrangian minimization algorithm are executed; this produces vast amount of printout and should be used only for specific purposes.

The abbreviations used in printouts are explained below.

ITER	Number of iterations
RINF	Norm of gradient (L_∞)
C	Norm of gradient (L_2)
GOAL	Value of goal function
NINF	Number of infeasibilities

SINF	Sum of absolute values of infeasibilities
MAXINF	Maximum value of infeasibilities (the name of the row concerned follows)
SITC	Small iteration (i.e., number of conjugate gradient iterations)
RO	Value of penalty parameter
EPSRO	Value of EPS/RO
COM.TIME	Computation time
GDUAL	Value of the gradient of the dual function
MULT. NORM	Value of the norm of the multipliers
FDUAL	Value of the dual function
ACT. ROWS	Number of active rows
BASIC COLUMNS	Number of columns that are not equal to a given bound
ALF	Step length

In addition, a report is issued each time the communication region is stored.

Finally, exit from the optimization routine and the status of the solution is reported.

5.3. Multicriteria optimization

The following information relates to the batch version of HYBRID since the PC version provides user friendly interface with context sensitive help.

A user may display (by LIST command - cf sec. 4.4.2.) following information about a solution of multicriteria problem. The meaning of displayed information is as follows:

WEIGTS	value of a weight coefficient (note that weight coefficients are normalized as discussed in the Methodological Guide [1]).
RFP	component of reference point for a criterion (aspiration level for a criterion)
VALUE	value of a criterion obtained in the last run for which an optimal solution has been found
WORST	worst value of a criterion (obtained during all modified runs). This is true nadir component, if an utopia point has been calculated (see following information).
BEST	best value for a criterion (with same reservations as for WORST)
U	logical variable indicating weather utopia point for a criterion has been already calculated (t) or has been not (f)
A	status of a criterion (t for active, f for nonactive)

Information about reference trajectories and components a criteria of types MIN and MAX may be displayed (also by LIST command) after command TRAJ or COEF, respectively. Those commands enables also change of respective values.

5.4. Results

Results are reported in standard MPS format with an additional column that contains (in the appropriate sections) scaling coefficients for each row and column. The definitions of additional rows and columns generated during transformation of multicriteria problem to an auxiliary single-criteria one, is given in the Methodological Guide [1].

Information provided in standard MPS format file is given also in a random access binary file which may be used for problem specific report writer. The structure of that file is illustrated by the following program which produces an ASCII file that contain a solution. The latter file is in format similar to the standard MPS output format, the only difference is due to replacement of the word "none" for an upper bound by the value defined as BIGN (cf sec. 4.2.5.) and by -BIGN for a "none" in lower bound column.

```
c
c This program reads random access binary file _userf generated by
c HYBRID and outputs results in MPS format file
c Meaning of variables is as follows:
c     pname   name of the solved problem
c     status  status of solution
c     goalv   value of objective function
c     itc     number of iterations
c     m       number of rows
c     n       number of columns
c     rown    name of a row
c     coln    name of a column
c     at      status of a variable
c     val     value of a variable
c     slacv   value of a corresponding slack variable
c     vll     value of lower bound for a row or variable
c     ull     value of upper bound for a row or variable
c     vdual   value of a corresponding dual variable
c
      character*2 at
      character*8 rown, coln,pname
      character*16 status
      real*8 goalv,val,slacv,vll,vul,vdual
      integer m,n,itc
c
      open(3,file = '_userf',access='DIRECT',recl=50)
      read(3, rec=1) m,n,pname,status,goalv, itc
      write(6,1000) pname,status,goalv,itc
      write(6,1001)
      irec=2
      do 10 i=1,m
      read(3, rec=irec)rown,at,val,slacv,vll,vul,vdual
      write(6,1003) i,rown,at,val,slacv,vll,vul,vdual
10    irec = irec+1
      write(6,1002)
      do 11 i=1,n
      read(3, rec=irec)coln,at,val,slacv,vll,vul,vdual
      write(6,1003) i,coln,at,val,slacv,vll,vul,vdual
11    irec = irec+1
c
 1000 format(  '1problem name       ',a8/
     *         ' status             ',a16/
     *         ' objective value    ',g14.5/
     *         ' iteration count    ',i6/)
 1001 format(//' section 1  -  rows'//'  number  ...row..  at  '
     *,'...activity...  slack activity  .lower limit..  '
     *,'..upper limit.  dual activity. '/)
 1002 format(//' section 2  -  columns'//'  number  .column.  at  '
```

```
     *,'...activity...  obj. gradient.  .lower limit..  '
     *,'..upper limit.  reduced cost.. '/)
1003 format(i8,2x,a8,2x,a2,5(2x,g14.5))
     end
```

6. TUTORIAL EXAMPLE

6.1. Sample of data

For a first run a user has to prepare the following three files:

- Specification file *__specs* specifies chosen options.
- File *__crit* that contains declaration and definition of criteria.
- MPS input file *__mps*, which contains the model to be solved (however without its multicriteria part). This file is reproduced in the Appendix.

The *__specs* file may contain the following statements:

```
multi
nstv 1
nt 4
accept
byrows
*
```

First four statements are necessary to declare the multicriteria problem (first one), to specify that the problem is dynamic (next two) and to allow for minor errors (like one entry in a row or duplicated elements). Last statement is optional and is for producing printouts of the resulting (after transformation of multicriteria problem to a corresponding single criteria LP) matrix by rows.

Declaration and specification of criteria may be done in the following way (below a contents of a *__crit* file is shown):

```
cons max
traj fol
endc max
*
cons .95*consu.01 +.9*consu.02 +.86*consu.03 +.82*consu.04
traj consu.
endc 1.*state.04
*
```

The above example is for generation of three criteria. Firstly, all criteria are declared by specification of each name followed by the criterion type. First and third criteria are of type MAX, whereas the second one is of type FOL. Secondly, after a declaration (which is ended by * in first column), definition of criteria follows. For the definition of the MAX–type criteria a linear combination of variables for chosen periods are used, while for the FOL–type criterion the 6–character generic name of a variable is used. The latter simple definition is expanded for all periods.

6.2. An example of an initial run

The following output was obtained for the specification file and definition of criteria presented in sec. 6.1.

```
                h y b r i d  -  version 3.1-PC August 1988

executed on  Sun Dec 04 13:05:20 1988


options:
initial run.  yes     dynamic pr..  yes     multicr....  yes
getfeasible.   no     modification   no     accept.....  yes
lower bound.  yes     upper bound.   no     scaling....  yes
mpsout.....   no      byrows......  yes     bycolumns..   no
spirit.....   no      comp.multp..   no     regul..zero   no
regul..ref..  no      quadratic.pr   no     precond.c.g   no
matrix prec.  no      paral. rows.   no


input files:
communicat..   none     mps input...   _mps     modificat..    none
multicriter.  _crit


output files:
communicat..  _comm     back-up.....  _back     solution...  _solpr
mps output..   none     byrows out..  _byrows   bycols out.    none
user file...  _userf


names:
problem name              objective....dummy..     (minimize)
rhs,ranges..              bounds......


dimensions:
rows.....  100     columns.....  300     elements...  1500
diff. elem..  1500     maxcr.....   10
integer parameters:
nt.....      4     nstv.....      1     max. errors    50
iter. log...    0     inorm.....     1     infeas.....     2
miter.....      0     nbc1.....      0     mtime.....     30


real parameters:
big number.    .1000E+31  zero  tol..    .2200E-15  small number    .1000E-05
lower boun.    .0000E+00  upper boun.    .1000E+31  feas. tol...    .1000E-03
tpara.....     .1000E-05  eps.....       .0000E+00  epsm.....      .1000E-06
epss.....      .7500E+00  epsd.....      .0000E+00  ro.....        .1000E+01
rost.....      .2000E+01  ro max.....    .5120E+03  ro step2....   .4000E+01
sbeta.....     .5000E+00  seta.....      .5000E+00  seps.....      .9850E+00
seps1.....     .1000E-01  meps.....      .1000E-01  qcf.....       .0000E+00
reta.....      .0000E+00  rmeta.....     .0000E+00  rseta.....     .0000E+00
em.....        .1000E+00  ems.....       .7600E+00  emmin.....     .1000E-05



input of mps format file: _mps
 card    section
    1    name          mann04
    2    rows
                         input of criteria definition from file _crit
                         generate   11 rows for multicrit. problem
                         updated max. number of rows:    24
```

```
                        updated max. number of criteria:     4
   16    columns
                        generate    9 columns for multicrit. problem
                        generate   45 elements for multicrit. problem
   49    rhs
   54    bounds
   64    endata
                        updated max. number of crit. comp.:     6
                        updated max. number of columns:    22
                        updated max. number of elements:    78

indices for time periods :

period no    0   1   2   3   4
index       00  01  02  03  04
variable state. for period 01 - set free
variable state. for period 02 - set free
variable state. for period 03 - set free
variable state. for period 04 - set free


definition of criteria
name  type       components
cons  max        .95000*consu.01       .90000*consu.02       .86000*consu.03
                 .82000*consu.04
traj  fol         consu.
endc  max       1.00000*state.04

 input summary
 =============
 objective       .dummy..   (min)
 rhs and ranges
 bounds

number of
 rows                24   (   8 eq,  10 le,   4 ge,   2 n)
 columns             22   (max. spec.     22)
 matrix elements     77   (max. spec.     78)
 total dif. magn.    27   (max. spec.   1500)
 rhs                  4
 ranges               0
 bounds               9

                  matrix only     matrix, rhs, ranges and bounds
 density          14.583%              -
 v                7.7600                 8.0361
 mean             .78306               -
 var              .38364               -
 min. elem.       .82000E-02             .82000E-02
 max. elem.       1.0000                 3.1600


scaling begins
                                  matrix only     matrix,rhs,ranges,bounds
iter     gn        v       min.coef   max.coef   min.coef   max.coef
 0  79.26       .9190       .4177      74.62      .1700      74.62
 1  19.44       .7218       .4293      32.03      .2185      32.03      g
 2  5.688       .6693       .5399      37.86      .2692      37.86      ccd
 3  1.426       .6622       .5344      34.05      .2798      34.05      ccd
optimal scaling   iter=   3 stop=  .98934
in-core swap of data

after scaling
                    matrix only     matrix, rhs, ranges and bounds
```

```
 v                  .7003                  .7843
 mean               1.403                -
 var                3.539                -
 min. elem.         .5000                  .3400
 max. elem.         32.00                  32.00


scaling coef.    minimal     maximal
rows             .50000      64.000
columns          .50000      2.0000
 problem printed by rows on file: __byrows

end with formulation of  the problem (     .05 min. execution time )
finished on Sun Dec 04 13:05:23 1988
```

The user may check the formulation of the problem by analysing the file *__byrows* which contains listing of the problem in the following form:

```
 problem summary
 ===============
 name               mann04
 objective        .dummy..   (min)
 rhs and ranges
 bounds

number of
 rows                 24   (   8 eq,  10 le,   4 ge,   2 n)
 columns              22   (max. spec.     22)
 matrix elements      77   (max. spec.     78)
 total dif. magn.     32   (max. spec.   1500)
 rhs                   4
 ranges                0
 bounds                9

                  matrix only     matrix, rhs, ranges and bounds
 density          14.583%               -
 v                .70032                  .78433
 mean             1.4032                -
 var              3.5389                -
 min. elem.       .50000                  .34000
 max. elem.       32.000                  32.000


rows

     1   state.01  (eq)    rhs =  .00000
     1.00000inves.01     1.00000state.00     -1.00000state.01

     2   state.02  (eq)    rhs =  .00000
     1.00000inves.02     1.00000state.01     -1.00000state.02

     3   state.03  (eq)    rhs =  .00000
     1.00000inves.03     1.00000state.02     -1.00000state.03

     4   state.04  (eq)    rhs =  .00000
     1.00000inves.04     1.00000state.03     -1.00000state.04

     5   =traj.01  (eq)    rhs =  .00000
     1.00000+traj.01    -1.00000-traj.01     -1.00000consu.01

     6   =traj.02  (eq)    rhs =  .00000
```

```
 1.00000+traj.02     -1.00000-traj.02     -1.00000consu.02

 7   =traj.03  (eq)    rhs =  .00000
 1.00000+traj.03     -1.00000-traj.03     -1.00000consu.03

 8   =traj.04  (eq)    rhs =  .00000
 1.00000+traj.04     -1.00000-traj.04     -1.00000consu.04

 9   cashf.01  (le)    rhs =  .00000      no range
 1.00000consu.01      1.00000inves.01      -.27000state.00

10   cashf.02  (le)    rhs =  .00000      no range
 1.00000consu.02      1.00000inves.02      -.28000state.01

11   cashf.03  (le)    rhs =  .00000      no range
 1.00000consu.03      1.00000inves.03      -.28000state.02

12   cashf.04  (le)    rhs =  .00000      no range
 1.00000consu.04      1.00000inves.04      -.29000state.03

13   <cons...  (le)    rhs =  .00000      no range
-1.00000..dummy.      -.95000consu.01      -.90000consu.02      -.86000consu.03
-.82000consu.04

14   <traj.01  (le)    rhs =  .00000      no range
-1.00000..dummy.      1.00000+traj.01      1.00000-traj.01

15   <traj.02  (le)    rhs =  .00000      no range
-1.00000..dummy.      1.00000+traj.02      1.00000-traj.02

16   <traj.03  (le)    rhs =  .00000      no range
-1.00000..dummy.      1.00000+traj.03      1.00000-traj.03

17   <traj.04  (le)    rhs =  .00000      no range
-1.00000..dummy.      1.00000+traj.04      1.00000-traj.04

18   <endc...  (le)    rhs =  .00000      no range
-1.00000..dummy.     -1.00000state.04

19   minca.01  (ge)    rhs =  3.1600      no range
 1.00000state.01

20   minca.02  (ge)    rhs =  3.1600      no range
 1.00000state.02

21   minca.03  (ge)    rhs =  3.1600      no range
 1.00000state.03

22   minca.04  (ge)    rhs =  3.1600      no range
 1.00000state.04

23   goal      (ne)
  .95000consu.01       .90000consu.02       .86000consu.03       .81000consu.04

24   .dummy..  (ne)
 1.00000..dummy.   .10000D-01+traj.01   .10000D-01+traj.02   .10000D-01+traj.03
  .10000D-01+traj.04   .10000D-01-traj.01
.10000D-01-traj.02   .10000D-01-traj.03   .10000D-01-traj.04  -.95000D-02consu.01
-.90000D-02consu.02  -.86000D-02consu.03  -.82000D-02consu.04  -.10000D-01state.04


bounds
          (fr)..dummy.        .65000(lo)consu.01        .65000(lo)consu.02        .65000(lo)consu.03
     .65000(lo)consu.04        .17000(up)inves.01        .17000(up)inves.02        .18000(up)inves.03
    .19000(up)inves.04       3.00000(fx)state.00             (fr)state.01             (fr)state.02
```

(fr)state.03 (fr)state.04

Note that in the listing the additional rows and columns (generated by HYBRID during transformation of the multicriteria problem into the equivalent single–criteria problem) are included. The meaning of those rows and columns is specified in the Methodological Guide [1].

6.3. Consecutive runs

In a first phase (cf sec. 1.2) a user may use information provided by HYBRID to verify the problem formulation. This may lead to introduction of minor changes in a matrix, which may be done with the help of ***modify*** option. After verification of a problem one may start actual optimization stage.

Having obtained first optimal solution a user may run the problem again for new values of the matrix elements and/or constraints and/or parameters for multicriteria optimization. Now a user of a batch version of HYBRID may use (for any next run of the problem) new options in *__specs* file. In the PC version change of options is provided in an interactive way. A user may also in an interactive way modify the problem by setting different values for a reference point, weights etc. It is also possible to look for "utopia point" or to temporally remove a criterion and so on.

7. REFERENCES

1 Makowski M., Sosnowski J., A Mathematical Programming Package for Multicriteria Dynamic Linear Problems HYBRID. Methodological Guide to Version 3.1, In: WP-88-071, IIASA, Laxenburg, July 1988.

2 Makowski M., Sosnowski J., A Mathematical Programming Package for Multicriteria Dynamic Linear Problems HYBRID. Methodological and User Guide to Version 3.03, WP-88-002, IIASA, Laxenburg, January 1988.

3 A. Propoi, Problems of Dynamic Linear Programming, IIASA, RM-76-78

4 A.Wierzbicki, A mathematical basis for satisficing decision making, WP-80-90, IIASA, 1980).

5. M. Kallio, A. Lewandowski and W. Orchard-Hays. An implementation of the reference point approach for multiobjective optimization. WP-80-35. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1980.

6 A. Lewandowski and Grauer M., The reference point optimization approach - methods of efficient implementation. CP-8-S12, IIASA Collaborative Proceedings Series: Multiobjective and Stochastic Optimization Proceedings of an IIASA Task Force Meeting, 1982

7. M. Makowski and J. Sosnowski. Implementation of an algorithm for scaling matrices and other programs useful in linear programming. CP-81-37. International Institute for Applied Systems Analysis, Laxenburg, Austria, 1981.

8 B.A. Murtagh, Advanced Linear Programming: Computation and Practice, Mc Graw–Hill, New York, 1981.

APPENDIX

MPS standard for input of data

A linear programming problem is usually defined in two parts. Firstly, a specification of problem is given. The is no standard for a problem specification, therefore we have adopted for HYBRID a very permissive way for problem specification which ease the specification and modification of the problem.

Secondly, the nonzero elements of matrix have to be entered. Most codes designed for solving linear programming problems follows the data format originally designed to the MPS series of codes developed for IBM computers. This format has become de facto standard adopted by most codes designed There are slight variations between codes, therefore we present the following specification which is accepted by most codes and do not restrict possibilities offered by the original standard developed for IBM computers.

The data that correspond to a linear programming problem is grouped in the following sections:
NAME
ROWS
COLUMNS
RHS
RANGES (optional)
BOUNDS (optional)
ENDATA

A section name must be entered starting with first column. The above listed order is compulsory. Section name should be entered in all either lower or upper cases (this depends on installation). Data in each section should be entered in a card image which has the following fields and corresponding contents:

	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
Columns	2-3	5-12	15-22	25-36	40-47	50-61
Contents	Indicator	Name	Name	Value	Name	Value

The indicators are listed below (in sections ROWS and BOUNDS). Indicators should be entered in all either lower or upper cases (this depends on installation). A value should be entered in floating point format.

We will examine each of the section in turn:

NAME — This section consists of just the section name card with a title of the problem placed in columns 15-22.

ROWS — In this section all the rows names are defined together with the row type. The row type in entered in Field 1 (in column 2 or 3) and the indicators have the following meaning :

- E equality
- L less than or equal

G greater than or equal

N free (no restriction); the first free-type row encountered is regarded as the objective row, unless the objective is explicitly identified in the specification file or by a code itself (the latter case applies for multiobjective optimization).

COLUMNS This section defines the variables and the coefficients of the constraints matrix (including the objective row). Only nonzero coefficients are entered. The data are entered column by column and all data for nonzero entries in each column must be grouped together contiguously. The card image contains the column name in Field 2, the row name in Field 3 and the value of coefficient in Field 4. One may enter two coefficients for the same column on one card by placing the name of the second row in Field 5 and the value of coefficient in Field 6. Columns for slack variables are taken care of by a code.

RHS This section contains the non-zero elements of the right-hand sides of constraints. The data format corresponds to COLUMNS section, the only difference is due to replacement of a column name by a label (with may also be blank). More than one right-hand side set may be specified in this section (the one to be used for the current run is specified in the specification file by its label). Therefore one card image contains the optional label in Field 3, the row name in Field 4 and the value of right-hand side in Field 5.

RANGES This section contains entries for inequalities rows for which both lower and upper bound exist. The data format is the same as for RHS section. The value of range is interpreted as the difference between the upper and the lower bound for the respective row. One of those bounds (if nonzero) is entered in the RHS section, the type of the row indicates whether lower or upper bound in defined in the RHS section.

BOUNDS This section contains changes for bounds for variables initially set to default values. The default bounds are usually defined as 0. for the lower bound and no constraint for the upper bound. The default bounds (for all variables) may be usually changed in the specification file. More than one set of bounds may be specified in this section (the one to be used for the current run is specified in the specification file by its label). More than one bound for a particular variable may be entered. The bound indicators have the following meaning :

LO lower bound

UP upper bound

FX fixed value for the variable

FR free variable (no bounds)

MI no lower bound, upper bound equal to 0.

PL no upper bound, lower bound equal to 0.

The card image has the following data format: Field 1 contains indicator of the type of bound, Field 2 contains optional label, Field 3 specifies the column name and Field 4 specifies value of bound, if applicable.

ENDATA — This section consists of just the section name card, signalling the end of data.

An example of MPS standard input file

The following example contains the input data file in MPS standard for the problem presented in sec. 5.1 of the Methodological Guide [1]. The problem has been generated for two periods of time. The meaning of variables is as follows: conn.. - consumption, inv... - investment, kap... - capital. Note that two last characters in a variable name correspond to a period number. The line that contains only numbers and underscores serves as a ruler and is not a part of the MPS format data file.

```
name            mann02
rows
 e  kap...01
 e  kap...02
 l  mon...01
 l  mon...02
 g  cka...01
 g  cka...02
 n  goal
columns
123456789_123456789_123456789_123456789_
    con...01  goal              0.95
    con...01  mon...01          1.00
    con...02  goal              0.90
    con...02  mon...02          1.00
    inv...01  kap...01          1.00
    inv...01  mon...01          1.00
    inv...02  kap...02          1.00
    inv...02  mon...02          1.00
    kap...00  kap...01          1.00
    kap...00  mon...01         -0.27
    kap...01  kap...01         -1.00
    kap...01  kap...02          1.00
    kap...01  cka...01          1.00
    kap...01  mon...02         -0.28
    kap...02  kap...02         -1.00
    kap...02  cka...02          1.00
rhs
    test1     cka...01          3.16
    test1     cka...02          3.16
    test2     cka...01          4.11
    test2     cka...02          4.11
bounds
 up bnd1      inv...01          0.17
 up bnd1      inv...02          0.17
 lo bnd1      con...01          0.65
 lo bnd1      con...02          0.65
 fx bnd1      kap...00          3.00
 fr bnd1      kap...01
 fr bnd1      kap...02
endata
```