SUPERIMPOSING DIRECT SEARCH METHODS FOR PARAMETER OPTIMIZATION ONTO DYNAMIC SIMULATION MODELS

Rainer Heckler
Hans-Paul Schwefel

ABSTRACT

An integrated modular software package has been developed by the Programme Group of Systems Analysis and Technological Development (STE) of the Nuclear Research Centre at Jülich (KFA) to provide automatic optimization of a set of user defined decision variables. This optimization module containing different procedures for direct search algorithms can be added to our FORTRAN based Data-Model-Interface (DMI) for dynamic simulation (1).
The latter is formulated independently from an actual simulation model as well as from the actual parameters, the objective, and the constraints which are chosen. Once initiated it controls all user specified input values and the linkage between the optimization algorithm and the simulation model. Taking the KFA-energy model (2) as an example one application for finding a specific energy policy is demonstrated.

I. INTRODUCTION

As a decision aid in energy systems analysis dynamic simulation models have shown to be a useful tool. In a consistent way they enable to investigate possible consequences of alternative strategies. For deterministic simulation models it is not only possible to compute the results for a given parameter setting but also to ask for the values of parameters producing a desired result.

Thus it is possible to invert the question "What will happen, if..." to "What should be done, in order to achieve a desirable result". Techniques for solving this task are generally known as optimization methods. For instance a lot of linear optimization models are in use, but their application is generally restricted by two characteristics:

- Linear or at least linearized relations between decision variables and criteria of goodness (objectives) are required.
- The results are very sensitive to small changes in the assumptions either concerning strategic parameters or numerical values for the constraints.

To at least partially overcome such difficulties it is necessary to choose optimization methods which do not require linear model structures but permit a formulation close to reality of the system to be simulated. There are, indeed, optimization routines that are applicable to such a broad extent. Although direct search techniques cannot guarantee the exact solution, as is the case in linear and non-linear programming, is it not better to apply a rather good method to a realistic model than to apply an exact method to a simplified one?

We want to show, how such optimization methods may be used in connection with a large dynamic simulation model for the energy system of the Federal Republic of Germany. This software package for superposing optimization procedures to our energy model we called GOLEM which stands for goal-oriented long-term energy model.

II. THE OPTIMIZATION MODULE

Combining simulation and optimization may be done in two principally different ways.

- optimization within simulation
- simulation within optimization

Both features are possible with our optimization module.

For preassigned time steps within a simulation run the optimization algorithm may be called. In this case the optimization algorithm can be incorporated as a subroutine of the simulation model.

But for a dynamic model it is not always sufficient to optimize the system for one single moment and even a sequence of optimizations for consecutive time points will usually not lead to an overall optimal solution. The path of a dynamic process within a definite system will be determined by system parameters, i.e. initial values and coefficients of differential equations. To achieve overall optimization it is necessary to run the model over the whole period for each parameter setting.

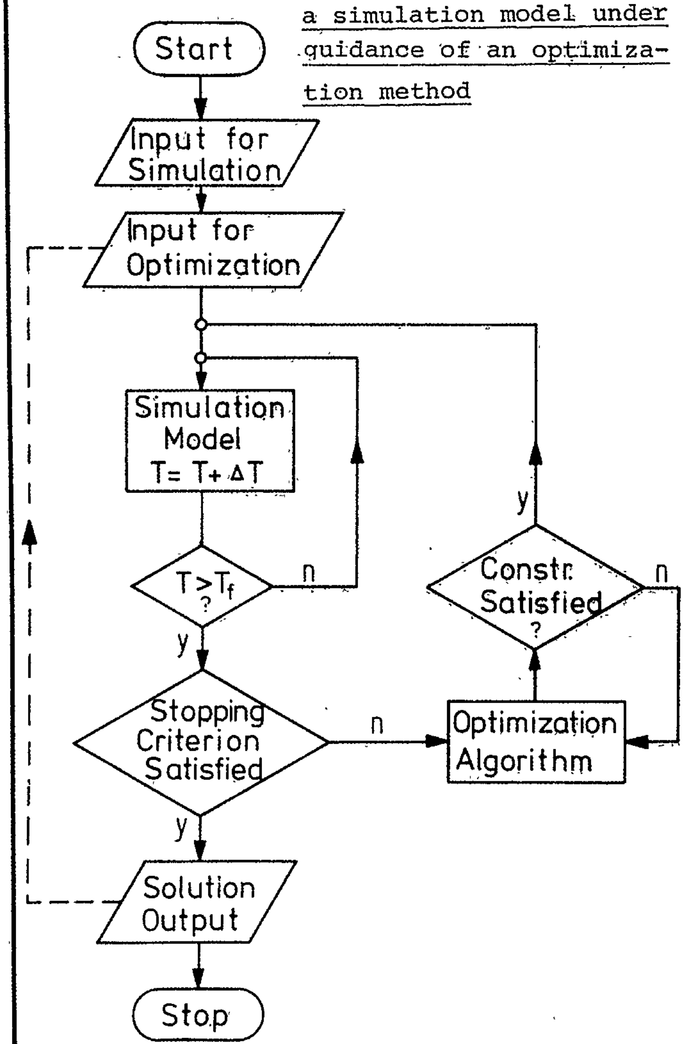
In principle the optimum-seeking technique handles the simulation program as a "black box". It generates consecutive parameter settings $p=\{p_i;\ i=1(1)n\}$ as input and receives output values $F(p)$ depending on the objective chosen. Instead of a series of optimizations within one model run, a series of model runs within one optimization task is performed (illustration 1).

The optimization module is constructed in such a way that there is a minimum of linkage between the simulation model and the optimum seeking program.

All variables corresponding to the different modules of the model, i.e. input and output variables, as well as the names of the objective and the constraints are defined as global variables. Their sequence is determined by position in a blank common which is used throughout all modules of the simulation model as well as the DMI subroutines, which represent the Data-Model-Interface (1), presented at this conference, too. From the point of view of the modules there is access by name whereas the optimization module accesses the variables by their position index in the common block.

ILLUSTRATION 1

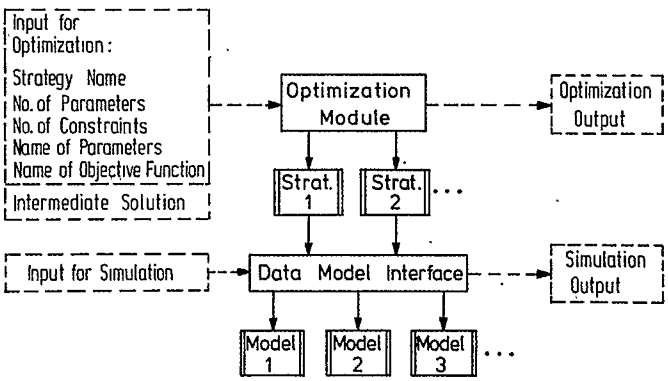
Flow diagram for running a simulation model under guidance of an optimization method



The optimization module takes it's starting values for the parameters from the data base of the simulation model by means of the blank common block just mentioned and replaces them by those values calculated during the iterations (illustration 2).

ILLUSTRATION 2

Input and output data handling for combined simulation and optimization



A short dataset contains the essential specifications for using optimization techniques for an optimization run with the user's simulation model, that may consist of several modules (subroutines). These specifications concern:

- The optimization strategy chosen denoted by a four letter abbreviation.
- A time limit for execution as termination criterion in addition to the normal convergence criterion.
- Accuracy parameters for the direct search method chosen.
- List of names (and in the case of arrays the indexes) of the parameters to be varied. Optional are starting values for the search step sizes and for the parameters. The latter overwrite the equivalent values given in the data base of the simulation model.
- The name of the objective function including information whether a minimum or a maximum is searched for.
- Name of items to be used for evaluating constraints.

There are three types of constraints which are treated in different ways:

- Constraints on parameters, for which it can be tested before starting simulation, whether or not they are respected.
- Constraints on any global variable defined in the simulation model, which can only be tested by running the simulation model. As soon as such a constraint is violated the simulation will be terminated. The difference between this termination time T_s and the final time T_f serves as restriction value, so that this type of constraint may be written as:

  $CSTR=T_f-T_s-\varepsilon\cdot\Delta T\leq 0;\ 0<\varepsilon<1$

  where ΔT is the time step size of the model.

  If upper and/or lower bounds for the parameters to be varied exist there is a more effective method, i.e. always transforming the parameters mathematically when they are transferred between optimization algorithm and simulation model.

Whereas the parameters to be optimized are exogeneous to the simulation model the objective function and constraints of the second type have to be defined and evaluated in one of the modules of the simulation model. If more objective functions and constraints are formulated in advance, the user is able to switch between different objectives and/or constraints only by changing the correspondent name(s) in the input file.

As soon as the termination or convergence criterion is satisfied, the computed parameter values, the respective actual search step sizes, the objective function value and some other useful information can be stored as an intermediate solution. Now it may be used as input for another optimization task, e.g. with an other optimization strategy or an other set of objective function and constraints.

Up until now, 15 different direct search routines are incorporated within our optimization module for finding best or at least improved solutions of a problem, and it is very easy to incorporate further optimization algorithms. The list of strategies incorporated up until now is shown in illustration 3.

ILLUSTRATION 3

List of strategies incorporated

CODE	DESCRIPTION OF STRATEGY OR VARIANT
	UNIVARIATE STRATEGY
FIBO	-WITH FIBONACCI SEARCH
GOLD	-WITH GOLDEN- SECTION SEARCH
LAGR	-WITH LAGRANGE INTERPOLATION
HOJE	PATTERN SEARCH (HOOKE AND JEEVES)
	ROTATING COORDINATES SEARCH
ROSE	-WITHOUT LINEAR SEARCH (ROSENBROCK)
	WITH LINEAR SEARCH
DSCG	-WITH GRAM- SCHIMDT ORTHONORMALIZATION
DSCP	-WITH PALMER ORTHONORMALIZATION
POWE	CONJUGATE DIRECTIONS SEARCH (POWELL)
	VARIABLE METRIC METHOD (DAVIDON, FLETCHER, AND POWELL)
DFPS	MODIFIED BY STEWART
SIMP	SIMPLEX SEARCH (NELDER AND MEAD)
COMP	COMPLEX SEARCH (BOX)
	EVOLUTION STRATEGY
EVOL	-TWO- MEMBERED (RECHENBERG)
GRUP	-MULTI- MEMBERED (SCHWEFEL)

Here we only want to describe a rather new one, which has proved to be the most reliable one in a very large test series (3), the so-called evolution strategy.

It is based upon a simple imitation of the basic rules of biological evolution: mutation and selection, population, recombination, and some others. It is not a Monte-Carlo method, though it contains some stochastic elements. The evolution strategy is a family of algorithms, the simplest one being similar to the adaptive step size random search of Rastrigin (4) and Schumer and Steiglitz (5), the most sophisticated one going far beyond Bremermann's 'search by evolution' (6).

Mutations are not pure random settings of the parameters but changes of the variables from one iteration (generation) to the other which belong to a Gaussian distribution. The parameters of that distribution, variances and co-variances, are attributes of each individual, just like the object parameters of the function to be extremalized. And they are changed from one generation to the other, too.

By selection of the fittest, the population not only creeps towards the optimum, but also adapts the parameters of the random mutability and thus accelerates the convergence, for example on ridges or in narrow valleys. Moreover, if the population is large enough, this method gives a rather good chance of finding a global out of several local optima, and there are nearly no restrictions to the type of objective functions. This method has proven to be the most reliable one out of all known direct search methods, especially when the number of variables is large. The computing times - they depend on the type of objective functions of course - often increase less rapidly with the number of free parameters, than is the case with other search algorithms.

III. THE KFA ENERGY MODEL

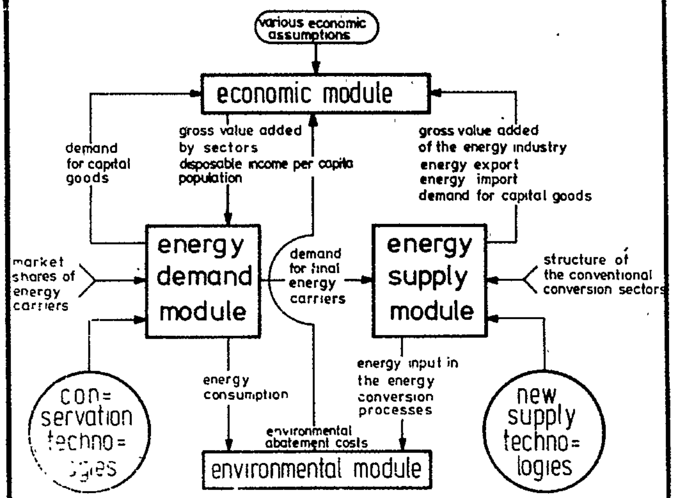
As a decision aid for planning the energy system of the Federal Republic of Germany a dynamic simulation model has been developed at the KFA Jülich. It was presented at the 1977 WSC (2). Only a summarizing overview shall be given here to understand the following example of an optimization run with it. To enable the model to be as flexible as possible, a modular structure was chosen. The complete model consists of four modules:

a. the macroeconomic module;
b. the energy demand module;
c. the energy supply module;
d. the environmental module.

Illustration 4 gives an overall view of the links between these modules, which form an interaction structure of closed-loop type.

ILLUSTRATION 4

Model structure



The aim of the macroeconomic module is to provide the necessary inputs for the energy demand module, i.e. the population size, the average disposable income per capita and the gross value added by branches. It may be influenced by a wanted or expected economic growth rate. Thus, different scenarios may be produced with respect to the development of the economic system.

The energy demand module calculates the energy demand by fuel and sector via correlation functions, which are gained from past data e.g. between the average disposable income per capita and the individual persons transport volume, together with specific energy consumption values and market allocations of different fuels.

In order to meet the energy demand, which is taken as the actual consumption in the supply module, a flow model was constructed. This is described mathematically by a set of simultaneous equations, which is solved for every year. For each of 14 energy carriers there must be a balance between the primary energy consumption and the indigenous production, the import, the export, the bunkering and the net stock changes on the one hand side and the final energy consumption, the non-energetic fuel consumption, the distribution losses and the resultant of the conversion balance (inputs minus outputs plus own consumption) on the other hand side.

The total emissions due to energy production, conversion and consumption are calculated in the environmental module, separately for the sectors and energy carriers which cause them and for the destinations air and water.

Within the energy demand module as well as the supply module new technologies for conservation and conversion may be activated by setting appropriate decision variable values.

IV. AN EXAMPLE

One example shall demonstrate the capability of GOLEM, our system of simulation and optimization tools. This example should not be taken as a forecasting result - this usually is wrong in connection with a simulation model -, nor even as a statement about an optimal development for the real energy system of the Federal Republic of Germany.

Finding optimal states or developments of a real system cannot be done with a single optimization run. One has to experiment with different criteria, alternative parameter sets and sometimes even with different models giving emphasis to different aspects of the same system.

What will be shown here is, that direct search techniques enable the user to find those parameters or time series within a dynamic simulation model which maximize or minimize an integral criterion under restrictions given to other resulting variables or derivatives of them. In principle a solution by hand is possible, too, but would cost even more simulation runs and give no certainty of having arrived at the desired solution.

Using the dynamic simulation model for the long-term energy model of the F.R.G. the following objective function was chosen.

$$\int_{t_i}^{t_f} (MPO(t)+MPM(t)) \cdot (t-t_i)\,dt \rightarrow \min$$

$t_i = 1985; \quad t_f = 2000$

This is the integral over the mineral oil (crude MPO and refined MPM) imports weighted with the time. As free parameters two times series were chosen:

FCTX(t) the quota of methanol added to motor spirit,

CATNL(t) the capacity of high temperature reactors used for production of process heat to gasify lignite,

each of which was given by base points at the years 1990, 1995 and 2000. The values for 1985 were set to zero.

Constraints were given to

$MPN(t) \leq RMPN(t)$ the imports of natural gas

$MPC(t) \leq RMPC(t)$ the imports of hard coal

$MGB(t) \leq MCB(t)$ the indigenous mining of lignite

according to exogeneous time series.

Methanol production as a new conversion technology uses gas which could be imported as natural gas or produced as synthetic natural gas by nuclear lignite gasification. Other possible options were not used in this case. Lignite now mainly is used for producing electricity. The indigenous mining being limited (imports are negligible) lignite gasification reduces lignite electrification which has to be compensated by other fuels. In this case hard coal had to fill the gap, but mining and imports of hard coal were restricted, too. On the other hand lignite gasification by means of nuclear process heat produces electricity and coke (to be used in blast furnaces e.g.) as byproducts, thus changing the balances for other energy carriers. An additional constraint had to be added in order to ensure that the remaining amounts of lignite for production of electricity would always be positive. There is not enough space here to explain all other relations within the energy supply module being affected by a combined methanol production and lignite gasification strategy.

Illustration 5 shows the printout of an optimization session at the timesharing computer (IBM 370/168 under TSS) of the KFA.

At first there are some DMI messages concerning the number of modules being incorporated, their time steps, the start and the end year of the simulation and the number of variables.

Then GOLEM, the optimization module, begins working and informs the user about the relevant control items and, at the end of the task, about the final state reached. In this case the capacity of high temperature reactors CATNL (in MW) and the methanol fraction of motor spirit at the distinct base years 1990, 1995 and 2000 are displayed. During the following output dialogue four illustrations were produced.

ILLUSTRATION 5

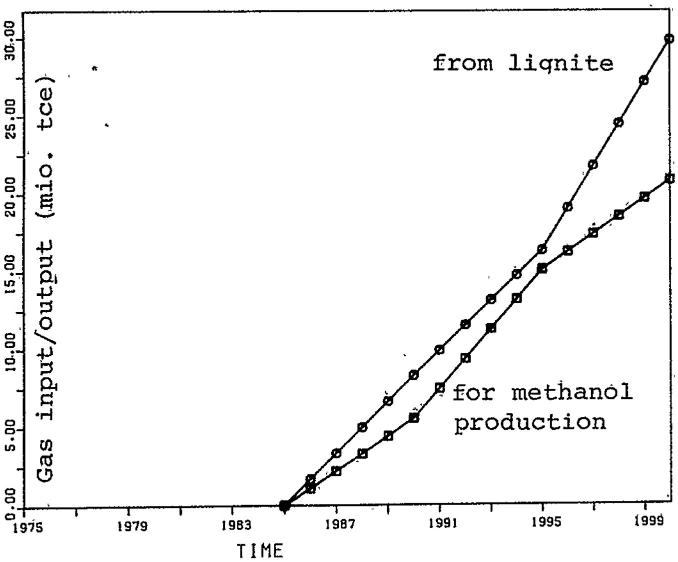
Printout of an interactive optimization session

```
-->
DMIDD EG,O
-->
RUNG EG,2,O

***************   D M I   *********  VERSION  1.0  ********* 16/08/78 ****

******  MOSQ =  1  2  3  4
**M2**  MODT =  1  8  8  8

******  TINI  = 1960.00       TEND  = 2000.00       DT     =  0.1250

**M3**  NR. OF GLOBAL   VAR.S  =  1731      NR. OF EXOGEN.  VAR.S  =   680

***************   GOLEM   ***** TSS  VERSION  1.1  ********* 18/08/78 ****

**M3**  MAX. EXECUTION TIME  SEC.  :                              900.00

**M3**          TWO  - MEMBERED EVOLUTION STRATEGY

**M3**  NO. OF PARAMETERS          :                                   6
**M3**  NO. OF CONSTRAINTS         :                                   1

**M2**  ACCURACY  EA = 0.1E-02  EB = 0.1E-02  EC = 0.1E-10  ED = 0.5E-05


**M3**  INITIAL OBJECTIVE VALUE    :                      0.0000000000E 00
**M2**  STARTING RANDOM NUMBER GENERATOR  :                          11685
                                        *
                                      * G *
                                        *
**M3**  CONVERGENCE CRITERION SATISFIED
**M3**  FINAL OBJECTIVE VALUE      :                     -0.1055559719E 10
**M3**  NO. OF SUCCESSES/MUTATIONS :                           21      120
**M3**  EXECUTION TIME        SEC. :                                651.70
**M3**  CALLS OF OBJECTIVE FUNCTION:                                    81
**M3**  CALLS OF CONSTRAINTS       :                                   121

                 TABLE OF FINAL   VALUES
  NAME    IEL     IP  TYP              SL           FINVAL           INIVAL
 CATNL     2   25246   10   0.1862959E-01   0.7498469D 01       0.0000000
 CATNL     3   25247   10   0.2624106E-01   0.1494613D 02       0.0000000
 CATNL     4   25248   10   0.3558256E-01   0.2744244D 02       0.0000000
 FCTX      2   25242   20   0.1176469E-02   0.8259789D-01       0.0000000
 FCTX      3   25243   20   0.1176469E-02   0.2242535D 00       0.0000000
 FCTX      4   25244   20   0.1176469E-02   0.3028294D 00       0.0000000
 GOAL      1    1730    1                  -0.1055560E 10  0.0000000E 00
 CSTR                   2                   0.6250000E-01

STORE?  (O/Y/S)
Y
```

Illustration 6 shows the development of the gas input for methanol production and of the amount of gas produced by lignite gasification.

ILLUSTRATION 6

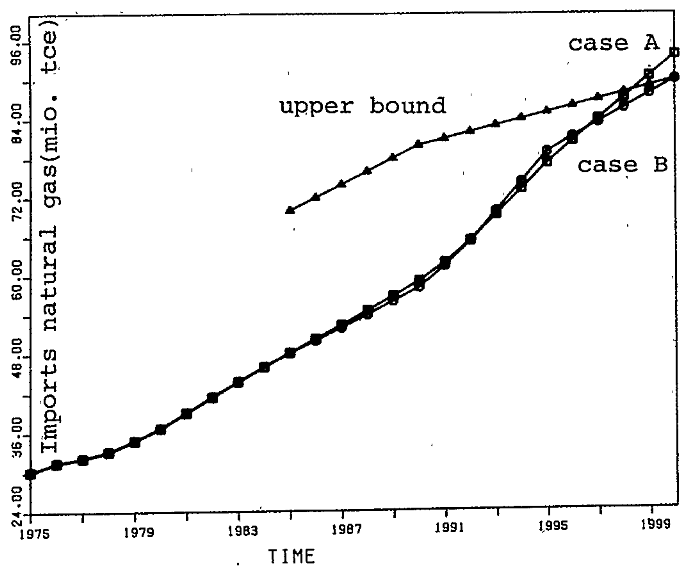
Gas input for methanol production
Gas output by lignite gasification



The latter being higher, especially towards the end of the time period is due to the restriction to natural gas imports.

ILLUSTRATION 7

Imports of natural gas

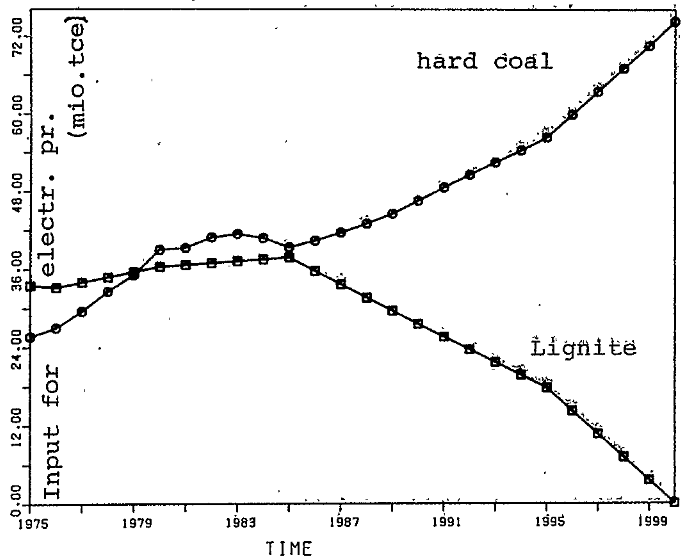


From illustration 7 you can see that this constraint is violated in the case A (no methanol and no gasification, which is the initial state). That means that the optimization had to start from a non-feasible point.

Illustration 8 demonstrates how the missing lignite for electricity production has to be replaced by a corresponding amount of hard coal.

ILLUSTRATION 8

Input for electricity production

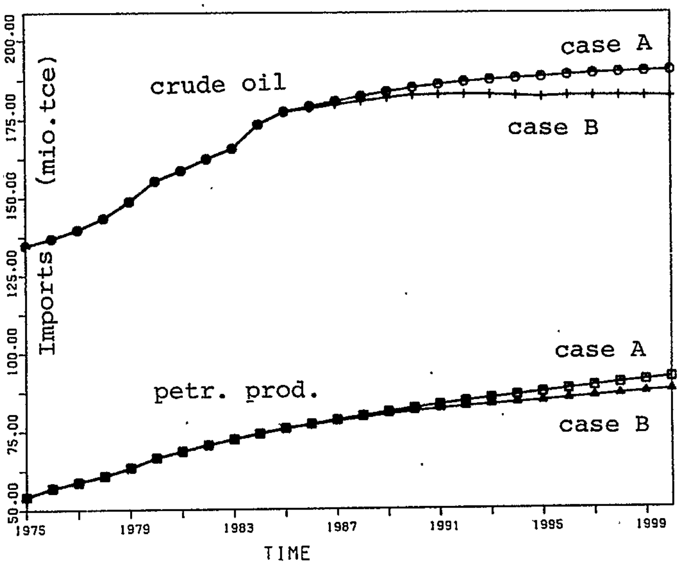


Finally the imports of crude oil and petroleum products which were minimized are shown in illustration 9 for the initial (case A) and the final state (case B) of the optimization task.

One important concluding remark has to be added here. Using GOLEM to find extremal solutions we have learned a lot more about the behavior of the energy model. Thus we are able to locate and diminish deficiencies which otherwise we would not have seen so clearly. The use of optimization algorithms does not only lead to better solutions for decision variables but also to a better model.

ILLUSTRATION 9

Import of crude oil and petroleum products



V. SUMMARY

The FORTRAN based optimization module described here presents the possibility to search for optimal parameters (even time series) within a dynamic simulation model. It is easy to handle even for the unexperienced user in a timesharing environment. The main features arise from its large flexibility with respect to

- the optimization procedure
- the objective and the constraints
- the parameters to be varied

which may be activated by merely filling in their names in a special input data set.

Together with the KFA simulation model of the national energy system this optimization module supports the search for desired developments of the energy system. That's why we call this tool GOLEM or goal oriented long-term energy model.

BIBLIOGRAPHY

1. Drepper, F. A Data - Model Interface for Modular Dynamic Simulation, presented at this conference.

2. Schmitz, K. and Schwefel, H.P. An Integrated Energy Simulation Model of the Federal Republic of Germany as a Decision Aid for Analyzing and Planning the Energy System. 1977 Winter Simulation Conference, Gaithersburg, Maryland.

3. Schwefel, H.P. Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Mit einer vergleichenden Einführung in die Hill-Climbing- und Zufallsstrategien. (in German: Numerical Optimization of Computer Models by Means of the Evolutionary Strategy. With a Comparative Introduction to the Hill-Climbing and to the Heuristic Strategies). Interdisciplinary Systems Research Nr. 26, Birkhäuser-Verlag, Basel, 1976

4. Rastrigin, L.A. The convergence of the random search method in the extremal control of a many-parameter system, Automation and Remote Control 24 (1963), pp. 1337-1342.

5. Schumer, M.A. and Steiglitz, K. Adaptive step size random search, IEEE Transactions on Automatic Control 13 (1968), pp. 270-276

6. Bremermann, H.J. Numerical optimization procedures derived from biological evolution processes, pp. 597-616 in Cybernetic Problems in Bionics, Gordon and Breach, New York, 1968