

Interactive Multicriteria Optimization in System Dynamics Models

M. Toro‡ A. Ollero‡

J. Aracil‡

‡Escuela Técnica Superior de Ingenieros Industriales,
Avda. Reina Mercedes s/n, 41012-Sevilla, Spain

†Centro Superior de Tecnologías de la Información,
Univ. Málaga.

Abstract

In this paper we present a program package which combines *System Dynamics Simulation* with programs for *Interactive Multicriteria Optimization* (IMO).

The program package incorporates conventional well tested routines for nonlinear optimization, that do not require previous computations of derivatives, and methods to optimize a set of objective functions by progressive articulation of the user preferences between different criteria.

To facilitate the user interaction, a special purpose man-machine interface have been included in the package. By means of this interface, the user can impose the required preference structure by only expressing, in a linguistic way, his/her opinion about each objective in the current solution of the Interactive Multicriteria Optimization algorithm.

The program package can be used to optimize a set of objective functions both in problems concerning the estimation of model parameters from historical data, and problems related with the search of optimal policies.

The man-machine interface and optimization programs have been written in C and linked with the DYNAMO continuous systems simulation language to configure the program package. The package can be used in IBM PC (or compatible) with a hard disk.

1 Introduction

The trial-and-error method has usually been used in System Dynamic models to determine good policies by the heuristic modification of control parameters defining policies. On the other hand, the identification of these models normally involve the heuristic search for appropriate values of models parameters. However, computer optimization searching seems to be an attractive tool for eliminating from further consideration large numbers of otherwise possible solutions.

From our point of view, the use of optimization methods in System Dynamic Models is restricted due to the difficulties in the formulation of the optimization problem. The optimization of System Dynamic Models is typically a Multicriteria Optimization Problem involving non-commensurable and conflicting criteria.

In this paper we consider methods in which the multicriteria search for the best policy is viewed as a Multiobjective Optimization Problem. A common way to solve this problem is by the combination of a scalar (one objective) optimization method with a strategy to articulate preferences about tradeoffs between objectives (see for example Zeleny, 1980; Osyczka, 1984).

Particularly, we deal with interactive multiobjective methods which are based on a progressive articulation of the preferences to generate a sequence of solutions which must converge to the preferred one.

Moreover, when considering the optimization of System Dynamic Models it is also necessary to cope with the inherent imprecision of the users of these models. Thus, the application of interactive multiobjective methods based on the user's precise estimation of tradeoffs between criteria in each interaction is a very unrealistic approach.

To overcome this problem we use techniques to deal with qualitative linguistic information in the man-machine interaction. Particularly, we have designed and implemented an interface based on previous results on approximate reasoning methods (Baptistella and Ollero, 1980) and its applications to System Dynamics (Ollero, Aracil and Camacho, 1984; 1986).

By using this interface the only informations required from the user are qualitative diagnosis concerning the degree of satisfaction of each objective in the current (non-inferior) solution of the multiobjective problem. From

this information the system deduces the appropriate modifications in the formulation of the next scalar objective function to be optimized.

2 Multiobjective Optimization of System Dynamics Models

In System Dynamics, after analyzing a problem the next step is to build an influence diagram containing the set of causes used to explain a phenomenon. Using this influence diagram a set of equations can be written and after being processed by a computer they must be able to represent the behaviour the real system presents. Validation of the model is the next step, and after it the model can be used as an instrument to obtain optimal policies using some reference as to optimal behaviour.

Simulation of the model may be done in DYNAMO language. This would be followed by an election of parameters so as to adjust the simulation output to a specific behaviour. Likewise in the final phase different parameters must be chosen so as to optimize some objective. If there are several objectives to be optimized that are incompatible between themselves this optimization may be difficult and an additional problem is introduced.

This problem may be studied systematically with routines that produce optimization. This brings us to another problem, using these routines we miss out on the simplicity of use of professional DYNAMO, one of its main highlights. In most cases we will have to rewrite the equations in FORTRAN or C languages.

So as to resolve both these problems a set of modules has been designed so as to use professional DYNAMO's simplicity and that translates to C language the original DYNAMO code. The most important object of these modules is to resolve the problem of multiobjective optimization.

Three modules have been built:

- Minimization.
- Deduction.
- Building the matrix of protocols.

These modules and professional DYNAMO can resolve efficiently the problem of optimization.

3 Minimization module

This module is designed to optimize a one dimension function. The Powell method is used so the jacobian matrix will not have to be entered. Only the C routine will be used with the -xf option of professional DYNAMO compiler.

Let the system we are studying be:

$$\frac{dx}{dt} = f(x, p)$$

with $x(t_0) = x_0$, and it is to be integrated from $t = t_0$ to $t = t_1$ with $x(t_1) = x_1$.

The module function is to minimize a set of objectives nz_i , $i = 1, m$. It will be assumed that z_i are the values of the system's variables nz_i at the time instant t_1 .

Considers that the optimization criteria of the model users (or builders) can be interpreted by using the objetive functions

$$z_1(x, p, t), \dots, z_m(x, p, t)$$

where

$$p = (p_1, \dots, p_r)$$

are a initial value for the parameters np_i , $i = 1, \dots, r$ that must be chosen to minimize the objetive functions.

The interactive multiobjctive problem is solved be means of a sequence of minimizations problems with a scalar function

$$\Phi = \phi(z_1, \dots, z_m)$$

The minimization module solve these scalar problems using the definition of Φ provided by the deduction module.

In this paper we use the well known funtion

$$\Phi = \sum w_i z_i$$

where w_i are normalized positive weights obtained in each interaction from the deduction module. We notice that all the solutions obtained by minimizing Φ are non inferior ones; that is, if p^* solves the scalar optimization problem, then there does not exist other solution \bar{p} such that

$$z_i(\bar{p}) \leq z_i(p^*)$$

with at least one of the inequalities being strict.

The information to the module is given by the file with the following configuration:

```

m
nz1
nz2
...
r
np1 ta1 p1
np2 ta2 p2
...
tol
```

where nz_i is the name of objective i , np_i the name of parameter i , ta_i its size (magnitude's order), p_i its initial value, tol the tolerance, m the number of objectives and r the number of parameters. As output the module gives a file of the same kind. Only p_i is actualized to the value calculated so that Φ is minimum.

4 Protocol building module

An algorithm must be implemented so as to adjust the weights to the objectives we obtain.

The evaluation of the order of importance between objectives will be done by the expert on the subject. These evaluations will generally not be very precise at least formally:

good
acceptable
neutral
poor
unacceptable

Figure 1: Atributes of the objectives

objective nz_1 is good
 objective nz_2 is acceptable
 ...

In Figs. 1 and 2 the atributtes of the objectives and weights are shown. Using these evaluations done by the expert a set of factors that will change the weights, fc_i , must be found. These change factors will modify the weights w_i for the next minimization of the function Φ . To do this we will use a simple technique. We will build in this module a protocol, that is a set of rules, such as:

if objective nz_1 is good and objective nz_2 is bad then small.increase weight 2.

if objective nz_1 is bad and objective nz_2 is good then small.decrease weight 2.

...

In these rules appear expresions of the kind: objective nz_1 is good, objective nz_2 is bad, small.increase weight 2, etc. This expresions are represented by usind fuzzy sets.

The set of rules must be defined for each problem. The deduction module built a protocol matrix to represent the rules. This matrix is computed by using deduction techniques based on approximate reasoning (Zadeh, 1973; Gupta and others, 1985).

To facilitate the definition of rules by the user, tabular representations can be used. In Fig. 3 we show seven trivial rules to modify the weight 2 from the evaluation of two objectives in the current solution.

large_increase
small_increase
nil
small_decrease
large_decrease

Figure 2: Atributtes of the weight

	acceptable	neutral	poor
acceptable	nil	small_increase	large_increase
neutral	small_decrease	nil	small_increase
poor	large_decrease	small_decrease	nil

Figure 3: Rules of the protocol

5 Deduction Module

This module contains the expert's opinion on the outcome of the objectives and has the form:

objective nz_1 is bad

objective nz_2 is good

...

Using these evaluations and the protocol matrix built previously it deduces a set of change factors of the ponderation weights so as to actualize them.

The way of doing this is by simulating the model in DYNAMO with the parameters values of the last minimization and observe through a report the fullfilment of the objectives.

6 Application

Consider the dynamical system given by the equations

$$\begin{aligned}\frac{d}{dt}var1 &= -k * (var1 - eq1) \\ \frac{d}{dt}var2 &= -k * (var2 - eq2) \\ var1(0) &= var1i \\ var2(0) &= var2i\end{aligned}$$

where $k, eq1, eq2, var1i$ and $var2i$ are constants.

The objectives are to fit the output of the system to the historical series generated by the equations

$$\begin{aligned}hist1 &= hist10 * exp(-kn1 * (t - t_i)) + hist11 \\ hist2 &= hist20 * exp(-kn2 * (t - t_i)) + hist21\end{aligned}$$

where appear the following constants:

$$hist10, hist11, kn1, hist20, hist21, kn2$$

The equations of the dynamical system are coupled through the parameter k , so the problem of fitting variables $var1(t)$ and $var2(t)$ to $hist1(t)$ and $hist2(t)$ is a multicriteria optimization one.

The criteria of performance can be written

$$\begin{aligned}dif1 &= (var1 - hist1)^2 \\ dif2 &= (var2 - hist2)^2 \\ \frac{d}{dt}er1 &= dif1 \\ \frac{d}{dt}er2 &= dif2\end{aligned}$$

with $er1(0) = er2(0) = 0$. Then the multicriteria optimization problem can be applied choosing the objectives $er1, er2, dif1$ and $dif2$.

The two first objectives measure the areas between the curves to be fitted, and with the latter the difference between the final states.

The steps are:

1. Optimize.
2. Evaluate every one of the four objectives.
3. Simulate to check the responses.

This sequence is followed until a reasonable level of satisfaction is reached.

With this set of modules the parameters of a model can be adjusted to the results of a simulation and a set of reference forms. The DINAMO equations, the objectives, parameters, ..., the resultants weights and parameters values, ..., of the example have been tested with the following model:

```
* MODEL
l var1.k=var1.j+(dt)(-k*(var1.j-eq1))
c k=3
n var1=vari1
c vari1=12
c eq1=15
l var2.k=var2.j+(dt)(-k*(var2.j-eq2))
n var2=var2i
c var2i=12
c eq2=20
l er1.k=er1.j+(dt)(dif1.jk)
a hist1.k=hist10*exp(-kn1*(time.k-timei))+hist11
r dif1.kl=(var1.k-hist1.k)*(var1.k-hist1.k)
c hist10=30
c hist11=25
n er1=0
c kn1=5.
l er2.k=er2.j+(dt)(dif2.jk)
a hist2.k=hist20*exp(-kn2*(time.k-timei))+hist21
r dif2.kl=(var2.k-hist2.k)*(var2.k-hist2.k)
c hist20=25
c hist21=20
n er2=0
```

```
c kn2=1.  
n time=timei  
c timei=0  
save var1,var2,hist1,hist2  
spec dt=0.1,length=4,savper=0.1
```

OBJETIVES

```
4  
ER1  
ER2  
DIF1  
DIF2
```

PARAMETERS

```
5  
EQ1 1 24.1484  
EQ2 1 21.6372  
K 1 1.57286  
VARI1 1 43.5066  
VAR2I 1 49.1276
```

TOLERANCE

```
1e-007
```

WEIGHTS

```
0.370100  
0.370100  
0.602516  
0.602516
```

Other possibilities might be optimization of specific policies for some model variables.

7 References

Baptistella L.F. and A. Ollero, 1980, *Fuzzy methodologies for iterative multicriteria optimization*, IEEE SMC-10, 155-165.

Gupta and others, 1985, *Aproximate Reasoning Expert Systems*, Nort Holland.

Osyczka A., 1984 *Multicriterion Optimization in Engineering*, Ellis Horwood.

Ollero, A. J. Aracil and E.F. Camacho, 1984, "Optimization of dynamic models: An interactive multiobjective approach", *Large Scale Systems*, 6, 1, 1-12.

Ollero, A. J. Aracil and E.F. Camacho, 1986, "Optimization of dynamic regional models", *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, 25, 1, 39-49.

Zadeh, L.A., 1973, *Outline of a new approach to the analysis of complex systems and decision processes*, IEEE Transactions on System, Man and Cibernetics, 3, 28-44.

Zeleny M., 1980, *Multiple Criteria Decision Making*, McGraw Hill.