Multi-Criteria Optimization in System Dynamics

Raimo Keloharju
Helsinki School of Economics

ABSTRACT

This paper begins by summarising some milestones in the expansion of the system dynamics methodology to give a background to multi-criteria optimization in system dynamics. The case of 'Inventory Control Policies' from Jarmain's (Editor) "Problems in Industrial Dynamics" is then used as an example to show how Wierzbicki's method in multi-criteria optimization can be adapted to system dynamics. The solution procedure transforms the model into a discrete trajectory in time.

METHODOLOGICAL BACKGROUND

When the methodology of SD was developed (Forrester 1961), computers were seen as fast calculating machines. A universal trend towards automation has since then changed completely the competitive environment of system dynamics and the framework of system dynamics modelling. In SD, the traditional way has been to build a model and then to simulate it. The current trend, however, is to change the emphasis from the model to one or more objective functions which guide the searching procedures for finding an acceptable model.

The philosophy behind the SD methodology is based on the assumption that the modeller has enough understanding of the problem under study to create the first version of a dynamic hypothesis. Ackoff has called this frame of mind preunderstanding as opposed to postunderstanding which results from a learning process (Ackoff 1973).

The idea of a computerized learning process was brought to the SD when the computer was given the possibility of searching for an acceptable model by using some objective function as the measure of performance. When it proceeds from one model version to the next, the modelling process is now automated (Keloharju 1976 and 1983). Fxperience and theoretical arguments support the idea that the search, which is based on heuristic optimization, should occur in parameter space. In the simplest case, the modeller chooses some one-dimensional objective function.

The optimization in parameter space is based on the idea that some combination of parameter values produces the ideal model behavior over the run-length. The minimization (or maximization) of such an average measure is the goal for the optimization process in its simplest form. The next logical step is to expand the idea to several objective functions.

An objective function can be formulated as a weighted combination of goals which produces the 'best' model behavior over the run-length after the optimization process has ended. The approach, which is similar to goal programming, has been called multi-objective optimization (Keloharju 1976). The choice of an acceptable combination of goals is based on the manual comparison of optimization results. In each case, a different objective function is used.

In adaptive optimization, the model changes during the 'real time' simulation. The objective function may be taken as given. The 'classical' solution procedure is based on a distinction between planning and action (Keloharju 1980). After each planning stage the model is revised automatically and then simulated for some chosen period of time (= action time). In this way, the model becomes discrete trajectory in the real time. Another possibility is to use control equations as a function of time. No use is then made of a separate planning stage (Kivijärvi and Tuominen 1986). This approach links SD to control theory.

After each planning stage automatic model revision makes it possible to simplify the model. Fven the feedback structure can be removed now (Keloharju 1987).

Figure 1 summarizes the short methodological review. It focuses on a distinction between the model and the objective function.

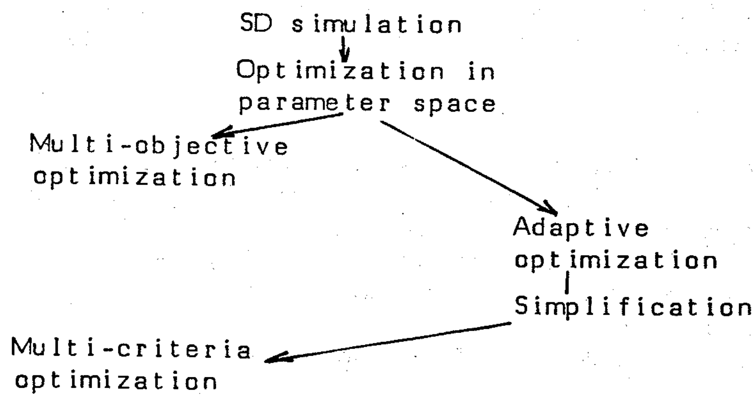
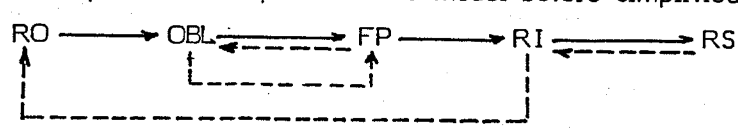


Figure 1. From simulation to multi-criteria optimization in SD

The inventory model from Jarmain's "Problems in Industrial Dynamics" (Jarmain 1963) is used as the demonstration example. The original purpose of the case was to show how some alternative policies create different model behavior. The current paper shows how the objective function can be changed automatically in an open model.

DESCRIPTION OF THE MODEL

Figure 2 shows a partial description of the model before simplification.



RO = Retail Orders
OBL = Order Backlog
FP = Factory Production
RI = Retail Inventory
RS = Retail Sales

Figure 2. Part of the original model

The feedback loop between OBL and FP is given; the effects of some given policies of retail ordering had to be estimated. Figure 2 shows that RO may depend, e.g., on RI. The alternative RO-policies of the case have been combined into a single policy equation which was then optimized by using one of alternative objective functions at a time (Keloharju 1976).

Figure 3 shows figure 2 after some modifications. Here decision making is based on information concerning the value of Average Retail Sales, ARS. Therefore the model itself is open. The information concerning ARS is multiplied by an estimation parameter (EP1, EP2), which is changed heuristically from time to time during the simulation (Keloharju 1987). This closes the model via some objective function.

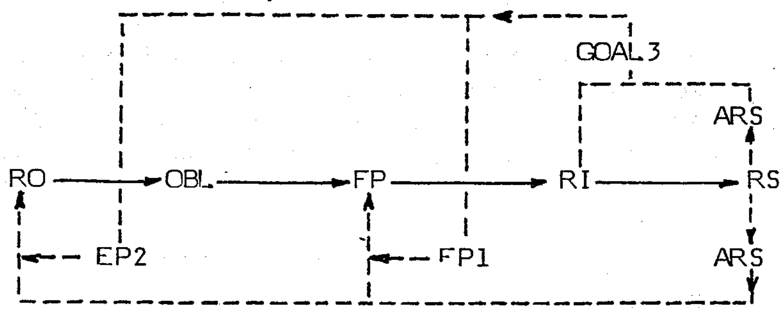


Figure 3. The simplified model makes use of feedforward

GOAL3, which is a possible objective function, is a function of RI and ARS. GOAL3 changes EP1 and EP2 the model is revised. Each time the model is revised the model has three objective functions to choose from (GOAL1, GOAL2, GOAL3). The choice is automatic. The model is now closed via feedforward paths from the objective function.

The model listing is in Appendix. In the reduced model version, used in his study, parameter SW on line 19 is zero. This directly affects the equation for FP on line 2 and RO on line 16. There are also indirect effects as many model equations were deleted from the information network.

GOAL1 is related to the cost of cumulative production change (FPCC). It is formulated as a weighted deviation from a target (Wierzbicki 1980):

A GOAL1.K=(PAR1*PAR1*TIME.K-FPCC.K)*(-WGHT1)
C PAR1=3
C WGHT1=300

PAR1 specifies the square root of the target value for average production change per time unit. GOAL1 is positive when the target value has not been attained.

The other goals are defined in the same way. GOAL2 relates to the cost of cumulative inventory changes, RICC. GOAL3 relates to cumulative inventory cost, RICOC.

The modeller searches for an acceptable solution by experimenting with PAR1, PAR2 and PAR3 since they relate to the goals of the study. The optimization problem is to minimize OBJ2 when

A OBJ1.K=MAX(GOAL1.K,GOAL2.K)
A OBJ2.K=MAX(GOAL3.K,OBJ1.K)

System dynamicists know that loop dominance may change when the model is being run. The same concerns objectives, too. The equation for OBJ2 takes this into account.

EXPERIMENTS WITH THE MODEL

Multi-criteria optimization can be seen as a very high order simulation. Ordinary optimization is based on repetitive simulation, i.e. optimization by simulation. In adaptive optimization, optimization by simulation and ordinary simulation alternate.

In multi-criteria optimization, adaptation concerns both the model and the objective function. Suppose, e.g., that the modeller gives the value of 20 from a terminal to PAR1, PAR2 and PAR3 in a multi-criteria optimization run (= run 1). Figure 4 shows that the model behavior is not acceptable because the output curve oscillates.

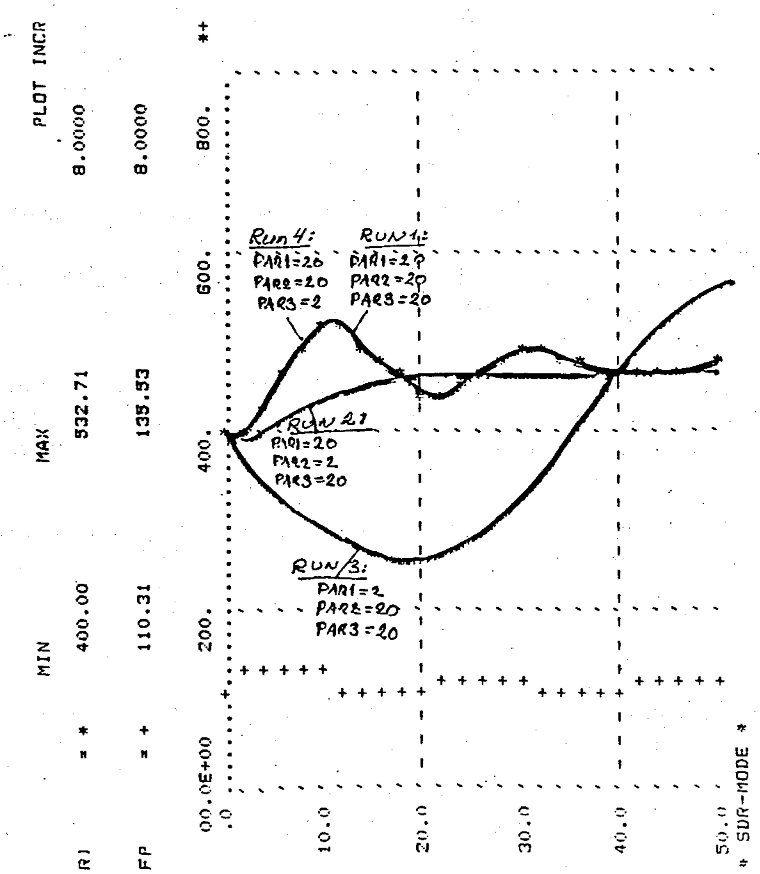


Figure 4. Some experiments in multi-criteria optimization

The modeller can search for an acceptable behavior by optimizing various combinations of parameter-values from a terminal. From the four experiments recorded in Figure 4, run 2 looks very good as a response to the step function used. The planning horizon and the action time are 1/5 of the run-length. These values were taken as given experimental parameters in the current study, an assumption which would be easy to remove.

The objective function chooses the constraining goal automatically. In run 2, the switching process occurred as follows:

Objective function	When in use (time in weeks)
GOAL2	0.5 - 1.5
GOAL3	2 - 20
GOAL2	20.5 - 30.5
GOAL3	31 - 50

Figure 5. Multi-criteria optimization is based on trade-off

CONCLUSIONS

System dynamicists can model and solve problems which other tool-kit-owners cannot. However I do not think that we as a scientific community can "grow" as long we isolate ourselves. In the long run, even survival is questionable without growth. Therefore I have tried to show here how SD can be made competitive with other Management Science tools. The bells are now tolling for system dynamicists.

REFERENCES

Ackoff, Russel L. (1973), The Future of Operational Research is Past. J.Opl.Res.Soc. 3(30):90-104

Forrester, Jay W. (1961), Industrial Dynamics . John Wiley and M.I.T. Press

Jarmain, W.Edvin (Editor) (1963), Problems in Industrial Dynamics . The M.I.T. Press

Keloharju, Raimo (1976), Multi-Objective Decision Models in System Dynamics. Dynamica 1(3)

Keloharju, Raimo (1980), General Frame of Resources, Structure and Trade-Off. Dynamica 1(6): 9-20

Keloharju, Raimo (1983), Relativity Dynamics . Acta Academiae Oeconomicae Helsingiensis. Series A:40

Keloharju, Raimo (1987), Simplification by Planning: The Third Way. Proceedings of the 1987 International Conference of the System Dynamics Society , Shanghai, China : 294-305

Kivijärvi Hannu and Markku Tuominen (1985), Solving economic optimal control problems with system dynamics. System Dynamics Review 2(2) : 93-104

Wierzbicki,A (1980), The Use of Reference Objectives in Multiobjective Optimization. Multiple Criteria Decision Making, Theory and Application. G. Fandel and T. Gal, Eds., Springer-Verlag, Berlin: pp 468-486

APPENDIX

```
* MULTI-CRITERIA OPTIMISATION ; NOVEMBER 2, 1987
L RI.K=RI.J+DT*(FP.JK-RS.JK)
R FP.KL=SW*PA.K+(1-SW)*EP1*ARS.K
L PA.K=PA.J+DT/TAP*(PI.J-PA.J)
A PI.K=FOB.K/WBD
L FOB.K=FOB.J+DT*(RO.JK-FP.JK)
L ARS.K=ARS.J+DT/TARS*(RS.JK-ARS.J)
A RID.K=WAS*ARS.K
A DDE.K=FOB.K/PA.K
L DDR.K=DDR.J+DT/TDDR*(DDE.J-DDR.J)
A PLD.K=(C1*DRO+C2*DDR.K)*ARS.K
R RS.KL=100+STEP(HGHT,STTM)+AMPL*SIN(6.28*TIME.K)/PERD
C HGHT=20
C STTM=0
C AMPL=0
C PERD=100
R RO.KL=(ARS.K+A1*(B1*RIDC+B2*RID.K-RI.K)/TAI
X +A2*((C1*DRO+C2*DDR.K)*ARS.K-FOB.K)/TAPL)*SW+
X (1-SW)*EP2*ARS.K
C SW=0
C EP1=1
C EP2=1
C A1=1
C A2=0
C B1=1
C B2=0
C C1=1
C C2=0
C TAP=4
C WBD=2
C TARS=1
C RIDC=400
C TAI=2
C DRO=2
C TAPL=2
C WAS=4
C TDDR=2
N RI=400
N FOB=200
N PA=100
N ARS=100
N DDR=DDE
L LFP.K=LFP.J+(DT)(FP.JK-FP1.JK)
N LFP=100
R FP1.KL=LFP.K
A FPCHA.K=FP.KL-FP1.KL
L FPCC.K=FPCC.J+DT*FPCHA.J*FPCHA.J
N FPCC=0
A GOAL1.K=(PAR1*PAR1*TIME.K-FPCC.K)*(-WGHT1)
C PAR1=20
C WGHT1=300
R RIR.KL=RI.K
L RI1.K=RI1.J+DT*(RIR.JK-RI1.J)
N RI1=400
A RICHA.K=RI.K-RI1.K
L RICC.K=RICC.J+DT*RICHA.J*RICHA.J
N RICC=0
A GOAL2.K=(PAR2*PAR2*TIME.K-RICC.K)*(-WGHT2)
C PAR2=2
C WGHT2=3
A RIDEV.K=WAS*ARS.K-RI.K
L RICOC.K=RICOC.J+DT*RIDEV.J*RIDEV.J
N RICOC=0
A GOAL3.K=(PAR3*PAR3*TIME.K-RICOC.K)*(-WGHT3)
C PAR3=20
C WGHT3=0.03
A OBJ1.K=MAX(GOAL1.K,GOAL2.K)
A OBJ2.K=MAX(GOAL3.K,OBJ1.K)
C DT=0.5
C LENGTH=50
A PRTPER.K=K1+STEP(K2,K3)-STEP(K2,K4)
C K1=0.5
C K2=0
C K3=100
C K4=100
C PLTPER=2
PRINT GOAL1,GOAL2,GOAL3,RI
PLOT RI=*,FP=+(0,800)
RUN
! +
```