Multi-Objective Decision Models in System Dynamics

Dr. Raimo Keloharju

The Helsinki School of Economics

PART II

ABSTRACT

This is a paper of a more technical nature, and it examines the effects of various planning horizons and of erroneous cost information on the model that was presented in the first part of the paper.

The study demonstrates perhaps counter-intuitively that mixed policies were more economical than more logical 'pure' policies of the traditional system dynamics type.

1. INTRODUCTION

In the first part of this paper a multi-objective decison model was introduced and some implied consequences of the approach used discussed. Let us take the same model again to explore two questions of a more technical nature:

(a) How does the cost structure affect the results obtained?

(b) What is the significance of a planning horizon in hybrid runs that combine simulation and optimization?

There is an analogy between a forecasting error and a cost function error to the extent that in both cases erroneous information might lead to unsatisfactory decisions. It has been shown in a recent study, however, that feedback models might be insensitive to forecasting errors when a simulation model is being optimized (6). Unreliable cost or profit information can be far more serious, however, because no feedback information will be available for corrective action. Actual forecasting error can be smoothed and a refined control policy might then prove effective but a company has to take the quality of cost or profit information as given, at least in the short run. This is way the cost structure topic deserves special attention.

2. COST STRUCTURE

The model was adopted from Jarmain's "Problems in Industrial Dynamics" (1). Initially the model had cost parameters PARA1, PARA2 and PARA3 with values of 300, 3, 0.03. The idea was to select these parameters in a way that approximately balances various cost components of the model. The cost parameters will now be changed in such a way that the range of parameter values changes symmetrically. OBJF was chosen as the objective function as it was the winner of objective function comparisons, described in part 1. Figure 1 shows the experimental design for cost parameter variation used and the results obtained.

Run number	(1)	(2)	(3)	(4)	(5)
Cost parameters					
PARA1	3000	300	60	30	15
PARA2	3	3	3	3	3
PARA3	0.003	0.03	0.15	0.3	0.6
Decision parameters					
A1	0.09	0.05	0.39	0.56	0.75
A2	0.83	1.0	1.0	1.0	1.0
B1	1.0	1.0	1.0	0.91	1.0
B2	0.03	0.5	0.08	0.12	0.13
C1	0.7	1.0	1.0	1.0	0.97
C2	0.02	0	0.08	0.13	0.08
Real cost	.245+06	.459+05	.757+05	.107+06	.149+06
Cost increase	434.9%	-	65.3%	133.6%	225.3%

Figure 1. Relationship between cost parameters and decision parameters with 150 iterations and with a simulation length of 50 periods

Figure 1 gives rise to the following conclusions:

(a) Estimation error caused by underestimation of the cost parameter range is less harmful than that caused by overestimation. Underestimation by a factor of ten increases costs only by a factor of 2.3. In a real multi-objective decison-making situation at least some objectives are incommensurable, however, and an estimation error is thus unavoidable. Still, an estimation error of a magnitude of ten times seems to be improbably high in a real life situation and this gives hope that workable model structures for multi-objective purposes will be found.

(b) The above-mentioned cost behaviour was primarily caused by parameter A1, which measures the ability of the model to make use of the inventory correction term. The high values of PARA3 (and thus low values of PARA1) mean that inventory discrepancy is a significant cost factor. SDR-algorithm reacted correspondingly and produced high values of A1. Relative cost parameter values were thus transferred to values of decision parameter A1. This indicates that without explicit use of cost information a system dynamics model might only give superficially valid results for decision-making purposes.

It might now be useful to examine cost estimation errors in even more detail, as this is a question of vital importance. There are at least three approaches available:

(a) <u>Maximize change in policy parameters due to changes in cost parameters</u>

To the original model were added sensitivity parameters PX11,...,PX32, which allow a maximum deviation of $\pm 10\%$ from the original, but this time 'unknown' cost parameter values. The initial values of sensitivity parameters were half-way between their upper and lower limits. Therefore, C PX11=15 and $0 \leq$ PX11 ≤ 30.

The final decision parameter values from Fig. 1, run (2), were given to the computer as the starting point, and at this time PX11,...,PX32 were the decision variables. The objective function OBJF used was maximized in order to find the least economical solution. The SDR-algorithm pushed cost-increase parameters PX11, PX21, and PX31 to their upper bounds and cost-decrease parameters PX21, PX22, PX32 to their lower bounds (=zero). This time the solution was trivial and as expected because all cost functions were convex. In a more complicated situation where convexity assumption is not valid, the final solution might be a trade-off between the cost functions. By starting from estimated maximum parameter errors, the experimenter could find a new value for the economic criterium (profit or cost) used, and the worst combination of sensitivity parameters. When the 'optimal'solution was found the first time (Fig. 1, run 2), sensitivity parameters had symmetrical values and cancelled each other out. By giving cost parameters new, corrected values the procedure earlier described might be repeated. It would be technically straightforward, therefore, to iterate in the following way :-

Iteration number	Optimizing criterium	Explorable parameters
1a	min OBJF	decision parameters
1b	max OBJF	sensitivity parameters
2a	min OBJF	decision parameters
etc.		

(b) Maximize change in policy parameters due to changes in cost parameters

To demonstrate this possibility a new objective function OBJP was formed in the following way :

$$\text{OBJP} = \max \sum_{i=1}^{6} \left| DP(i)_o - DP(i) \right| \quad , \text{ where}$$

OBJP = objective function for post-optimality analysis

DP(i) = decision parameter i ; i=1,...6

$DP(i)_o$ = initial value of decision parameter i (from run 2)

Both decision parameters DP(i) and sensitivity parameters were explorable parameters this time. The upper and lower bounds used where the same as before. The final solution obtained was as follows:

FINAL SOLUTION

A1	=	1.00
A2	=	.00
B1	=	.00
B2	=	1.00
C1	=	.00
C2	=	1.00
PX11	=	3.22
PX12	=	3.22
PX21	=	.03
PX22	=	.03
PX31	=	.00
PX32	=	.00

NO OF OBJ FCT EVALUATIONS	161
INITIAL VALUE OF OBJ FCT	.00000
FINAL VALUE OF OBJ FCT	.54500+01

This time, decision variables were pushed as far as possible from their optimal values, which were given in run 2. Decision parameter A1, for example, had received the value 0.05, and thus ended up in the upper limit 1. The new parameter combination found gave to OBJF a value of .186+10, which indicates that OBJP really did find an extremely poor solution.

As there was no cost component in the objective function, positive and negative sensitivity parameters received symmetrical treatment. Interpretation of their numerical value of about 1% error might be difficult. It could indicate, however, that the optimum solution was quite insensitive to cost parameter errors. Perhaps one should, after all not maximize OBJP but OBJP + OBJF.

(c) Investigate cost uncertainty by Monte-Carlo simulation

Monte-Carlo simulation has been used as a management consultant's tool to estimate uncertainty-based errors in company models that have been derived from historical company data (3). It is a relatively simple task to write a computer program which does required sampling until the reliability of results satisfies the experimenter. This idea could be applied to post-optimality cost parameter analysis by letting the computer

- automatically select random cost parameter values
- make a 'simulation' run based on optimal parameter values and on changed cost parameter values
- record wanted variable values like, for example, cumulative costs
- repeat the procedure until the experimenter is satisfied with the statistical risk remaining.

The procedure outlined above would require some changes in the present version of SDRDYN: A new iteration-oriented module that takes care of post-optimality analysis should be added. This is no longer, however, a fundamental problem, but a purely technical task. It has been proved that DYNAMO can be coupled succesfully with an optimizing tool, in our case with SDR. It follows from this that DYNAMO should be coupled with additional tools as long as the marginal utility of doing this remains positive. In an ideal situation, the linking of different tools would not be permanent, but could be done each time from a terminal.

3. RUN-LENGTH

An experiment was made by using a sine-wave (amplitude 50, period 100) as input information. The positive half of the sine-wave generated results summarized in Fig. 2. The total run-length of 50 periods was composed of shorter lengths in runs (7), (8), and (9). In run (7), for example, there were two runs of 25 periods covering periods 1-25 and 26-50 respectively.

Run number	(6)	(7)	(8)	(9)
composition of total run-length	50	25+25	17+17+16	13+12+13+12
cost	.99420+05	.55895+05 .38541+05	.50919+05 .11311+05 .31529+05	.29539+05 .55897+05 .60960+05 .34898+05
Total cost	.99420+05	.94436+05	.93759+05	.181129+06
Used CPU-time (Sec.)	6.773	8.322	9.849	11.301

Fig. 2. Relationship between cost and run-length

The results of Fig. 2 imply that optimum run-length is affected by several factors :-

(a) Model-generated cost and run-length decrease simultaneously, because the model can now better adapt itself to environmental changes. The decreasing cost trend of models (6) to (8) is caused by final parameter values as the comparison in Fig. 3 indicates. The values change considerably for the second half of the total run-length as the demand is then falling.

	Run (6)	Run (7)	
Parameters	Periods 1-50	Periods 1-25	Periods 26-50
A1	1.0	0.99	0.58
A2	1.0	1.0	0
B1	1.0	1.0	0.99
B2	0.14	0.16	0.15
C1	1.0	0.96	0.97
C2	0	0	0.31

Figure 3. Effect of run-length on parameter values

(b) Model-generated cost increases, however, when run-length decreases below some minimum length. Run (9), for example, gave costs that were about twice that in earlier versions. Why then did splitting of the run-length cause this strange result? Figure 4 collects some useful information from simulated data in order to understand what really happened. Parameter values are from periods 1-25 in run (7) and from periods 1-13 in run (9).

	Parameter A1 Run number		Parameter A2 Run number		Retail inventory		
						Run number	
Iteration	(7)	(9)	(7)	(9)	Period	(7)	(9)
1	1.0	1.0	0	0	0	400	400
5	0.9	0.9	0.1	0.1	5	395	369
10	0.68	0.68	0.32	0.32	10	416	312
15	0.68	0.68	0.32	0.32	15	438	266
20	0.616	0.216	0.784	0.784	20	454	229
25	0.616	0.216	0.384	0.784	25	472	200
30	0.739	0	0.461	1.0	30	494	195
35	0.539	0	0.661	1.0	35	512	224
40	0.467	0	0.993	1.0	40	503	264
45	0.467	0	0.993	0.996	45	494	327
50	0.579	0	0.681	1.0	50	511	414
60	0.633	0	0.728	1.0			
80	0.866	0	1.0	1.0			
150	0.99	0	1.0	1.0			

Figure 4. Comparison of runs (7) and (9)

Parameter A2 had a final value of one in both cases, but parameter A1 had a value of zero in version (9). When the 'planning horizon' of run-length was decreased to cover only the first 13 periods, the model visualized continuously increasing demand. In this situation inventory correction term to retail order rate was ineffective, and the ordering decision was based, therefore, on sales information. That is why the SDR-algorithm selected the following equation for periods 1-13 :-

OR.KL=ARS.K+(DRO+0.01*DDR.K)*ARS.K-FOB.K)/TAPL.

Also time series data for retain inventory show that the version (9) model was losing inventory as long as demand was increasing. This is a perfectly valid assumption if the world ended with the planning horizon. As it certainly does not, the problem of run-length selection is an important one, and similar to the problem of planning horizon selection. It is possible to reduce the planning horizon by using an artificial loss function that increases towards the end of the planning horizon. (4), (2). Formulation of such a constraint is related to the system then under study and requires, therefore, more art than science.

(c) CPU-time used increased with decreasing simulation length as expected. In the reported experiment each consecutive run started with the same parameter combination for A1,...,C2 instead of using the latest values. This procedure might have increased computer search time to some extent, but it reduced the risk of not finding a new solution.

4. FINAL REMARKS

This paper has shown that an optimizing system dynamics model was applicable for multi-objective decision-making purposes. All test runs had a fixed number of interations without any interruptions by the user. SDRDYN accepts any number of iterations and allows the model builder to check results from a terminal. With this information he may decide either to obtain more iterations or to change the objective function used to another one by selecting from an earlier defined group of potential functions.

A model builder might, for example, let the computer make a relatively small number of iterations and then might want to take the experimentation initiative from the computer. At any time the initiative can be returned to the computer again, all of which means a real interactive process in the full meaning of the expression. This possibility indicates one very fruitful research area, as well as hopefully improving system dynamics acceptability as a real world management tool. Also more sophisticated problem areas with genuine multi-objective character, different ways of formulating decision parameters and of giving them initial values, might be interesting subjects for further research efforts. Much more work should be done, however, before any firm conclusions can be drawn from the use of a tool which only strives for an optimum solution without any guarantee of really being able to reach it.

There are four pure policies that could be used when making decisions for the aggregate production and aggregate work force as demand fluctuates: (5)

(1) altering the size of the work force by hiring or laying off
(2) using overtime or idle time
(3) holding the production rate constant and allowing the fluctuations in demand to be absorbed by changes in the inventory level
(4) using subcontracting

In a real life production planning situation the most economical solution is a combination of the pure strategies listed above. Correspondingly, a combination of information use alternatives might prove to be most economical because information is just a mirror of what actually happens in a feedback model.

```
DYNAMO II F    *****A VERS. SDR *****
  PRECOMPILER   MOD SJ KKK-ATK
   DATE 76-09-25 TIME 09:05:21

>#add kelohar ju.multiob j
       * PRODUCTION-DISTRIBUTION SYSTEM
       NOTE  ORDINARY EQUATIONS
       L RI.K=RI.J+DT*(FP.JK-RS.JK) RETAIL INVENTORY
       R FP.KL=PA.K FACTORY PRODUCTION
       L PA.K=PA.J+DT*1/TAP*(PI.J-PA.J) PRODUCTION ABILITY
       A PI.K=FOB.K/WBD PRODUCTION INDICATED
       L FOB.K=FOB.J+DT*(RO.JK-FP.JK) FACTORY ORDER BACKLOG
       L ARS.K=ARS.J+DT*1/TARS*(RS.JK-ARS.J) AVERAGE RETAIL SALES
       A TRID.K=WAS*ARS.K TRIAL VALUE FOR RETAIL INV DESIRED
       A RID.K=MAX(TRID.K,TVRI.K)  RETAIL INVENTORY DESIRED
       A TVRI.K=STEP(450,TTC) TERMINAL VALUE OF RETAIL INVENTORY
       C TTC=1000 TIME FOR TERMINAL CONSTANT
       A DDE.K=FOB.K/PA.K DELIVERY DELAY ESTIMATE
       L DDR.K=DDR.J+DT*1/TDDR*(DDE.J-DDR.J) DELIVERY DELAY RECOGNIZED
       A PLD.K=(C1*DRO+C2*DDR.K)*ARS.K PIPELINE ORDERS DESIRD
       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
       NOTE EQUATION FOR RETAIL ORDERS
       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
       NOTE DECISION PARAMETERS
       C A1=1
       C A2=0
       C B1=1
       C B2=0
       C C1=1
       C C2=0
       NOTE SENSITIVITY PARAMETES
       C PX11=15
       C PX12=15
       C PX21=0.15
       C PX22=0.15
       C PX31=0.0015
       C PX32=0.0015
       C A1C=0.05
       C A2C=1.0
       C B1C=1.0
       C B2C=0.5
       C C1C=1.0
       C C2C=0
       NOTE ORDINARY PARAMETERS
       C TAP=4 TIME TO ADJUST PRODUCTION
       C WBD=2 WEEKS OF BACKLOG DESIRED
       C TARS=4 TIME TO AVERAGE RETAIL SALES
       C RIDC=400 RETAIL INVENTORY (IF A CONSTANT)
       C TAI=2 TIME TO ADJUST INVENTORY
       C DRO=2 DELAY IN RECEIVING ORDERS
       C TAPL=2 TIME TO ADJUST PIPELINE
       C WAS=4 WEEKS OF AVERAGE SALES
       C TDDR=2 TIME TO ADJUST DELIVERY DELAY RECOGNIZED
       NOTE INITIAL EQUATIONS
       N RI=400
       N FOB=200
       N PA=100
       N ARS=100
       N DDR=DDE
       NOTE ****MULTI-OBJECTIVE PART OF THE MODEL****
       NOTE  CFPCO IS AN OBJECTIVE FUNCTION CANDIDATE
       L LFP.K=LFP.J+(DT)(FP.JK-FP1.JK)
       N LFP=100
       R FP1.KL=LFP.K
       A FPCHA.K=FP.JK-FP1.JK
       A FPCOST.K=(PARA1+PX11-PX12)*FPCHA.K*FPCHA.K FACTORY PROD COST
       L CFPCO.K=CFPCO.J+DT*FPCOST.J CUMULATIVE FACTORY PRODUCTION COST
       N CFPCO=0
       NOTE  CRICH IS AN OBJECTIVE FUNCTION CANDIDATE
       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
       A RISCH.K=(PARA2+PX21-PX22)*RICHA.K*RICHA.K
       L CRICH.K=CRICH.J+DT*RISCH.J CUMULATIVE RETAIL INV COST
       N CRICH=0
       C PARA2=3 COST PARAMETER NUMBER TWO
       NOTE  CRICO IS AN OBJECTIVE FUNCTION CANDIDATE
       A RICOST.K=(PARA3+PX31-PX32)*(WAS*ARS.K-RI.K)*(WAS*ARS.K-RI.K)
       L CRICO.K=CRICO.J+DT*RICOST.J CUMULATIVE RETAIL INVENTORY COST
       N CRICO=0
       C PARA3=0.03 COST PARAMETER NUMBER THREE
       NOTE DERIVED OBJECTIVE FUNCTION CANDIDATES
       A OBJF.K=CFPCO.K+CRICH.K+CRICO.K
       A OBJ2.K=CFPCO.K+CRICH.K
       NOTE OBJECTIVE FUNCTION COMPONENTS FOR
       NOTE POST-OPTIMALITY ANALYSIS
       A A1D.K=A1C-A1
       A A2D.K=A2C-A2
       A B1D.K=B1C-B1
       A B2D.K=B2C-B2
       A C1D.K=C1C-C1
       A C2D.K=C2C-C2
       NOTE OBJECTIVE FUNCTION FOR POST-OPTIMALITY ANALYSIS
       A OBJP.K=MAX(A1D.K,-A1D.K)+MAX(A2D.K,-A2D.K)+
       X MAX(B1D.K,-B1D.K)+MAX(B2D.K,-B2D.K)+
       X MAX(C1D.K,-C1D.K)+MAX(C2D.K,-C2D.K)
       NOTE  OUTPUT SPECIFICATIONS
       PRINT 1)RS,RO/2)RI,CFPCO/3)CRICH,CRICO/
       X 4)OBJF,OBJ2/5)FOB,DDR/6)PA,ARS/7)LFP,OBJP
       SPEC DT=0.5/LENGTH=50/PRTPER=1/PLTPER=0
       RUN
```

References

(1) Jarmain, W.E. (Editor) (1964) — Problems in Industrial Dynamics (pp. 119-124), The M.I.T. Press

(2) Keloharju, R. (1975) — System Size-Methodological Trade-off in Aggregate Production Planning, The Finnish Journal of Business Economics, Vol. 24, No. 2

(3) Shegda, M. (1969) — Management Science Workshop of Lybrand, Ross Bros. & Montgomery, Chicago, III. Distributed material.

(4) Sikes, T.W. (1970) — The Search Decision Rule Applied to Aggregate Planning, Ph.D. thesis, UCLA

(5) Silver E.A. (1972) — Medium Range Aggregate Production Planning: State of the Art, Prod. & Inv. Management, 1st Qtr., 1972.

(6) Winch, G.W. (1976) — Optimization Experiments with Forecast Bias, Dynamica, Vol. 2, Part 3.