A Case Study in System Dynamics Optimisation

R. Keloharju
(Helsinki School of Economics)

E. F. Wolstenholme
(University of Bradford Management Centre

Abstract

This paper presents the use of optimisation as a tool for policy analysis and design in system dynamics models and presents a demonstration of its use on the 'project model' developed by G. P. Richardson and A. L. Pugh III in their book "Introduction to System Dynamics Modelling with DYNAMO". The use of optimisation to design parameters, table functions and new model structure is shown to produce a very significantly improved performance for this model compared to conventional approaches.

INTRODUCTION

The purpose of this paper is to present a case study to demonstrate the merits of using optimisation for the purpose of policy design in system dynamics models.

Traditionally, system dynamics has relied very extensively on the use of intuition and experience by system owners and analysts to help design policies for improving system behaviour over time. This situation is now changing and much effort is being expounded in the development of policy design methods. Basically, two schools of thought are emerging. The first of these concerns the application of control theoretic methods such as eigenvalue analysis, linear control theory, Routh stability criterion, model control theory and optional control theory. (Sharma (1985), Mohapatra and Sharma (1985). These approaches can be powerful but do required a level of assumption and analytical ability out of keeping with the original aims of system dynamics, which was to facilitate the exploration of systems by as wide a range of practitioners as possible. Intensive use of these methods is not anticipated until computer software is developed to improve their ease of application.

The second major aproach to policy design which has emerged in recent years is that of simulation by optimisation (Keloharju (1983), Gustafson and Wiechowski (1986)). This approach also relies fundementally on computer software but is not inhibiting in it's dependence on sophisticated analytical techniques. The software to be described and applied for optimisation in this paper was originally developed in the late 1970's as an appendage to DYSMAP (Dynamic Simulation Model Application Programme) (Cavana and Coyle 1982) and known as DYSMOD (Dynamic Simulation Model Optimiser and Developer). This software is currently under further development by Bradford and Salford Universities in the United Kingdom and will be released shortly as a specific version of the redeveloped and restructured DYSMAP2.

The DYSMOD Optimiser

Although the concept of optimisation is not new in system dynamics, the DYSMOD approach to model development and analysis provides a new dimension to system dynamics. The software uses a hill climbing routine to heuristically determine the optimum values for any number of model parameters relative to predefined objective functions or performance measures. Essentially, the method assumes a system dynamics model as a starting point. However, experience uses might formulate their models somewhat unconventionally to give the software maximum scope to assist with the task of model development.

Optimisation in parameter space is achieved by interleafing simulation and optimisation. One iteration of the procedure consists, firstly, of a DYSMAP simulation run, in which the value of the objective function is recorded, and secondly in a run of the optimiser to choose parameter values which might improve the objective function. Subsequent iterations consist of rerunning DYSMAP to test out the resultant improvement in the objective function under the new parameters and further refinement of them by optimisation. Any one experiment with the software might take a 100 or more iterations. This procedure presents few problems, however, given efficient software and the current downward trend in computer hardware costs.

One of the more basic and but perhaps more trivial uses of such an optimisation technique (which is often wrongly considered as it's only use) is in fitting models to past data. Whilst the method has an important role in the area of validation, it's much more influential role is in parameter and table function policy design and, less obviously, in structural policy design. The latter is achieved by combining alternative policy equations using pseudo parameters to achieve 'mixed' rather than 'pure' policy analysis. Further, by combining real and pseudo parameters in an experiment, parameter and structural policy design can be carrried out simultaneously and in a process which subsumes sensitivity analysis. In conventional system dynamics practice these activities must be carried out separately and sequentially and in a very limited way. In optimisation the model structure can be considered as a continuum and the process considered as one of choosing or synthesising a final model from an infinite number of possible models, comprised of all the parameter and structural permutations offered (a variety set).

In large scale models the policy design process is further facilitated in DYSMOD by the use of a base vector convention similar to those used elsewhere; for example in the simplex algorithms of mathematical programming. That is, only a limited number of parameters are considered 'free' or in the base at any time; but that the candidate parameters for the base can be changed as optimisation proceeds. Other sophistications available, but outside the scope of this writing are the concepts of model simplification (that is, the driving of psuedo perameter variables to zero to eliminate model structure) and optimisation over variable time horizons within the simulation.

Figure 1 gives an overview of the structure of, and interactive imputs to DYSMOD. During optimisation/simulation only the final values of perameters and objective functions are printed. The rerun facility allows the final 'model' to be run conventionally to examine other model variables and their behaviour over time. Subsequently, further optimisation can be undertaken with revised paramters, objective functions, numbers of iterations, etc.

A Case Study

1. <u>A Specimen Model.</u> Optimisation as a policy design tool is best demonstrated by using a system dynamics model as its starting point. In order to avoid the devotion of time and space here to introducing a new model it was decided to choose a well known model as a candidate for optimisation. The model chosen for this purpose was the excellent 'project model' (Richardson and Pugh (1981)), originally developed to explain the process of applying system dynamics and to demonstrate the power of the method for exploring the merits of alternative system operating poicies. It should be stressed that this choice is not in any way meant to imply criticism of the policy design method used by Richardson and Pugh. Indeed the opposite is the case and the work here should be seen as a way of extending the analysis provided by these authors.

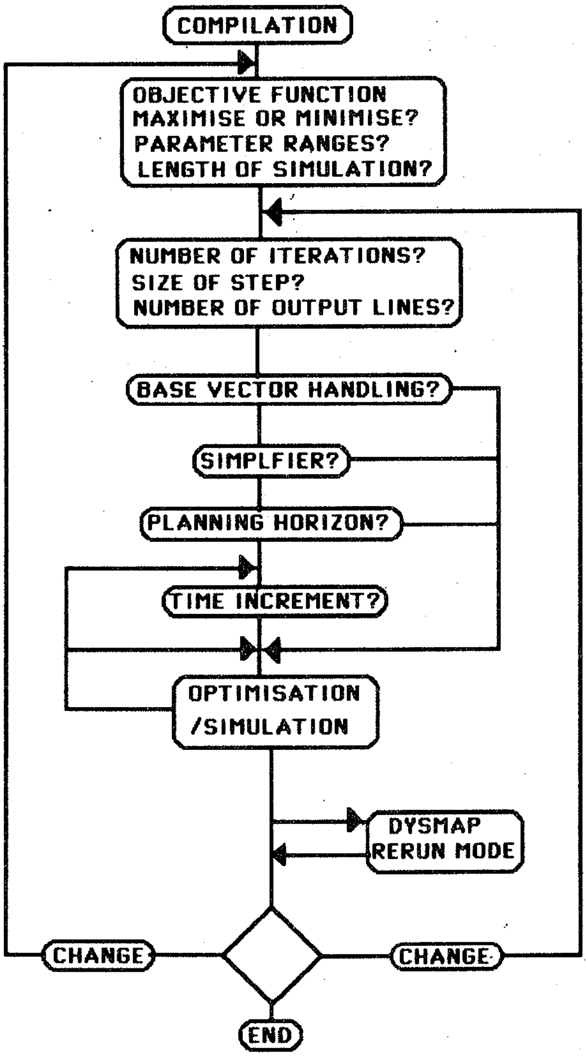

FIGURE 1 THE STRUCTURE OF DYSMOD

The purpose of the 'project model' was to explore the evolution over time of a project involving a set of tasks. The basic version of the model foçused on factors affecting project overruns, and subsequent developments dealt with the introduction of more realism into the model (for example disaggregating the workforce) and in examining the redesign of parameter and structural based polices for control of the project. Performance measures of total project time and labour costs were introduced as a basis by which to compare alternative policies.

An influence diagram of the model tailored to this presentation is given in Figure 2 and a full listing of the model, including additional equations for the purpose of optimisation, is provided in Appendix I. Figure 2 is somewhat self explanatory and no further discussion of the existing model will be given here. Instead, the optimisation modifications to the model will be described together with some policy experiments and results for comparison with those achieved on the basic model. Comments will also be made on these results where they provide evidence of the general insights which can be gain from the optimisation procedures.

2. <u>Amendments to the Model.</u> It will be seen from Appendix I that a supplementary set of equations are introduced into the model (lines 98 - 111). Since the optimisation process consists of many simulation runs, it is firstly very necessary to have precisely calculated performance measures rather than to rely on extracting such information from output graphs. AUX1, ..., AUX4 therefore store for later calculation the exact time of project completion and variables AUXC1 and AUC2 register the total cost from the project. Secondly, it is necessary to define a suitable objective function. In the case of the project model an appopriate measure is the trade off between the cost and the completion time for the project as given by the equation for OBJ1.

It is stated by Richardson and Pugh that (for such projects of the type described in the model); "whether or not a policy is an improvement depends on how one weighs the additional cost against the saving in completion time. A system dynamics model does not set or evaluate the criteria for improved system behaviour ....people make the value judgements." However, in optimisation it is perfectly feasible to explore such judgements in the model. It will be seen that the equation for OBJ1 incorporates a parameter 'WEIGHT to achieve this. Varying WEIGHT between model experiments allows the strength of the trade-off between cost and time to be investigated. AUX5 in the equation for OBJ1 is included in order to penalise early completion and avoid trival results.

(Equations 105, 106, 107 and 110 given in Appendix I facilitate global sensitivity analysis and are not used in the experiments described here.)

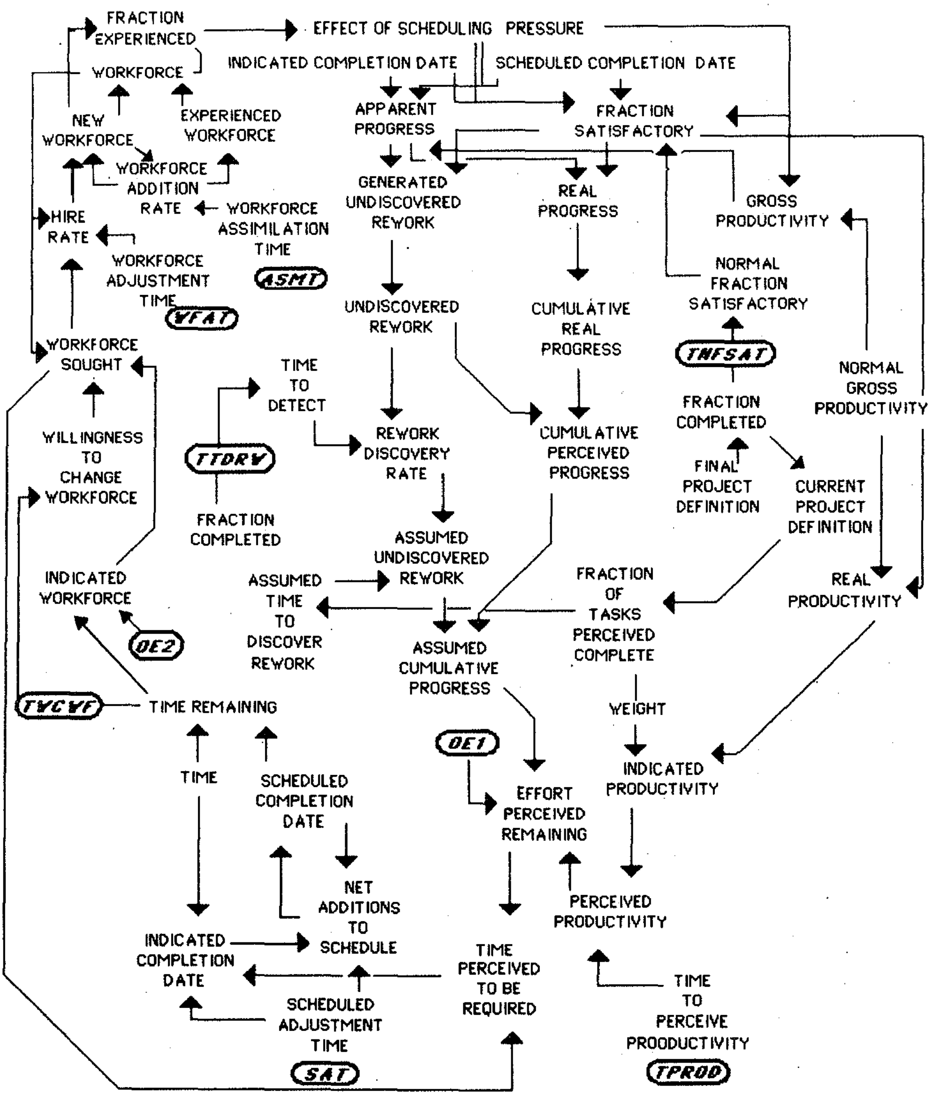
FRACTION EXPERIENCED
EFFECT OF SCHEDULING PRESSURE
WORKFORCE
INDICATED COMPLETION DATE
SCHEDULED COMPLETION DATE
NEW WORKFORCE
EXPERIENCED WORKFORCE
APPARENT PROGRESS
FRACTION SATISFACTORY
WORKFORCE ADDITION RATE
GENERATED UNDISCOVERED REWORK
REAL PROGRESS
GROSS PRODUCTIVITY
HIRE RATE
WORKFORCE ASSIMILATION TIME
ASMT
WORKFORCE ADJUSTMENT TIME
WFAT
NORMAL FRACTION SATISFACTORY
UNDISCOVERED REWORK
CUMULATIVE REAL PROGRESS
TNFSAT
WORKFORCE SOUGHT
TIME TO DETECT
NORMAL GROSS PRODUCTIVITY
FRACTION COMPLETED
WILLINGNESS TO CHANGE WORKFORCE
TTDRW
REWORK DISCOVERY RATE
CUMULATIVE PERCEIVED PROGRESS
FINAL PROJECT DEFINITION
CURRENT PROJECT DEFINITION
FRACTION COMPLETED
ASSUMED UNDISCOVERED REWORK
INDICATED WORKFORCE
ASSUMED TIME TO DISCOVER REWORK
FRACTION OF TASKS PERCEIVED COMPLETE
REAL PRODUCTIVITY
OE2
ASSUMED CUMULATIVE PROGRESS
TWCWF
TIME REMAINING
WEIGHT
OE1
INDICATED PRODUCTIVITY
TIME
SCHEDULED COMPLETION DATE
EFFORT PERCEIVED REMAINING
NET ADDITIONS TO SCHEDULE
PERCEIVED PRODUCTIVITY
INDICATED COMPLETION DATE
TIME PERCEIVED TO BE REQUIRED
SCHEDULED ADJUSTMENT TIME
TIME TO PERCEIVE PROODUCTIVITY
SAT
TPROD

FIGURE 2.
INFLUENCE DIAGRAM FOR PROJECT MODEL

3. Direct Parameter Policy Design. Richardson and Pugh defined some parameters which were used in their parameter based policy design experiments. These are listed in Table 1 (and highlighted on Figure 2) together with the values allocated to them by Richardson and Pugh in their base model run. This table also sets ranges for these variables which were used in the optimisation experiments. The conventional approach of system dynamics is to vary these paramters either one a a time or in combination to establish their effect. Examples of the conventional results obtained by Richardson and Pugh for project cost and time are shown in Table 2. Run 1 on table 2 represents the results from the base run of their model and the second line represents the effect of a new parameter set. Lines 3 - 8 in Table 2 give results from the optimisation experiments. In each line the objective function weight was changed and the parameter values given are those chosen by the optimisation software.

Table 1. Parameters Defined for Parameter Policy Design Experiments.

		Value used by R and P Base Model run (weeks)	Range set for Optimisation Experiments (weeks)
TPPROD	Time to Percieve Progress	6	3 - 6
WFAT	Workforce Adjustment Time	3	1 - 10
SAT	Schedule Admustment Time	6	1 - 10
ASMT	Workforce Assimilation Time	6	1 - 10

Table 2 Results of Conventional and Optimisation Experiments for Parameter Policy Design.

	RUN	WEIGHT	ASMT	SAT	WFAT	TPPROD	TIME	COST
Richarson)	1	-	6	6	3	6	61.500	6.247E+6
and Pugh)	2	-	3	4	2	3	60.250	5.861E+6
)	3	0.50	1.0	3.673	1.008	4.893	52.750	5.4898+6
)	4	0.60	1.0	1.792	1.032	6.000	53.250	5.398E+6
Optimiser)	5	0.65	1.0	5.249	1.005	6.000	52.750	5.578E+6
)	6	0.70	1.0	4.951	1.001	6.000	52.750	5.567E+6
)	7	0.80	1.0	4.951	1.001	6.000	52.750	5.567E+6
)	8	1.10	1.0	4.950	1.000	3.270	52.50	5.659E+6

Some interesting factors emerge from these results. Firstly, the optimiser results indicate that it is possible to reduce both project time and cost significantly from the base run; and from the position reached by the individual parameter changes suggested by Richardson and Pugh, by manipulating the four basic parameters. Secondly, time and cost do not appear to be sensitive to the value of the weight attached to these factors in the objective function. Thirdly, that the approximate same time and cost can be achieved by many different permutations of the perameters. This result might be considered to contribute to the proof of the view that there is no generally right combination, of parameters for a model, which is often claimed in conventional simulation.

It should also be noted here that WFAT and ASMT are chosen to be small. The former implies that workforce imbalances should be eliminated as quickly as possible and the latter that the workforce should be assimilated as quickly as possible. This may of course incur additional training costs and highlights the need for this factor to be included in the model. In fact, it is interesting to reflect that such low values of these parameters might never have been tested out in conventional system dynamics policy analysis, since conventional wisdom would have expected instabilities to have been created. Conversely, the results imply that the scheduled adjustment time (SAT) and the time to percieve progress (TPPROD) need not be low.

Finally, it should be noted that identical results are obtained with weights of 6.7 and 6.8. This results implies that model behaviour is not necessarily always a function of policies and that what is really important in system dynamics is the point along each continuous parameter trajectory at which behaviour does change.

4. Table Function Policy Design. As an alternative to changing specific parameters in the model, policy design can also be carried out by varying table functions. Three of the table functions suggested by Richardson and Pugh for this purpose are given in Table 3. These are also highlighted in Figure 2. Table 3 defines each of these functions and gives the ranges of the tables used in the optimisation experiments. In each case the first boundary given is that used in the base run of the model by Richardson and Pugh.

Run 1 in Table 4 shows an example of the results obtained by Richardson and Pugh from changing TWCWF alone relative to the base model run and setting this at the alternative boundary defined in Table 3. Runs 2, 3 and 4 in Table 4 show the results from optimisation when the previous parameters, plus each table function in turn, are chosen by the software. Run 5 allows a pair of table functions to change together. All optimisation experiments were carried out a weight value of 0.6.

Table		Range Set for Optimisation Experiments
TTDRW	Table for time to detect rework	boundary I 12/12/12/10/5/^{0}.5 boundary II 6/6/6/5/3/0.4
TWCWF	Table for willingness to change workforce	boundary I 0/0/0/.1/.3/.7/.9/1 boundary II 0/0/.1/.9/1/1/1/1
TNFSAT	Table for normal fraction satisfactory	boundary I .5/.55/.63/.75/.9/1 boundary II .6/.63/.7/.8/.92/1

TABLE 3. Table function defined for Policy Design Experiments.

	RUN	ASSMT	SAT	WFAT	TPPROD	TABLE FUNCTIONS	COST($)	TIME (WEEKS)
Richard-son and Pugh	1	6	6.000	3.000	6.000	TWCWF 0/0/.1/.9/1/1/1/1	6.5076E+6	55.75
(	2	1	5.500	1.000	3.018	TTDRW 6.9/8.3/6.2/5.0/5/.5	5.1951E+6	49.50
Opti- (	3	1	4.240	1.054	6.000	TWCWF 0/0/.1/.8/.8/1.0/.9	5.566E+6	48.50
miser (	4	1	4.946	1.000	6.000	TNFSAT .6/.6/.7/.8/.5	5.052E+6	51.25
(	5	1	7.418	1.001	6.000	(TTDRW 12/12/5/12/10/5/.5) (TWCWF .1/.7/.6/.7/.9)	5.104E+6	47.75

TABLE 4. Results of Conventional and Optimisation Experiments for Table Function Design.

In general the results in Table 4 again show significant improvement in both time and cost for the project relative both to the base run and the Table 2 results. Run 2 suggests an interesting policy alternative which changes loop polarities. That is that the time to discover rework should be a concave function of the project progress rather than a convex function. This function is chosen to be to close to its lower boundary which suggests that the faster rework is discovered the better for the project. This, in turn, suggests better quality control is needed and indicates, realistically, that the cost of this should be taken into account in the model. In runs 3 and 4 the optimisation also chose the alternative boundaries of TWCWF and TNFSAT for which the same conclusions can be drawn as for run 2. In run 5 the model developed in run 4 for TNFS was taken as a starting point and the other two table functions optimised. The results are the best of this group.

5. Structural Policy Design. Richardson and Pugh also introduced structural policy design experiments into their model. An overestimation parameter was introduced for this purpose and considerable discussion was presented by these authors as to whether overestimation should be applied to effort perceived remaining on the project or to the indicated workforce. The final choice for locating this parameter was recommended to be in the equation for the effort perceived remaining. In order to investigate this issue using optimisation two overstimation parameters were defined. As will be seen in the appendix OE1 was defined in the equation for effort perceived remaining and OE2 was defined in the equation for the indicated workforce. An experiement was then conducted allowing OE1 and OE2 to vary between 1 and 3 and allowing ASMT, SAT, WFAT and TPPROD to vary as defined in Table 1. Table 5 shows the results of this experiment with weight values from 0.4 to 0.65 (runs 2-7) together with the Richardson and Pugh result from the base model with OED inserted in the effort perceived remaining equation at a value of 1.5.

In all cases of the optimisation results in Table 5, the project schedule of 40 weeks could be kept, and at a reasonable cost. (This cost could possibly be improved even further by incorporating the table function related policies defined earlier). The results also shed considerable light on the choice between OE1 and OE2. In the project model OE1 can be considered to represent an overestimation of demand and OE2 an overestimation of supply. When cost considerations are more important that time time considerations (WEIGHT is small) there is no need to push supply. Therefore, OE2=1 and OE≥ 1. When time is more important than cost (WEIGHT is large) there is no need to push demand. Therefore, OE1 = 1 and OE2 ≥ 1. In other words the two overestimation parameters were never chosen to be effective together but a clear point emerged at which it was preferable to switch from one to the other. This situation is somewhat analogous to the situation of monetary policy in economics and Table 5 gives and answer to the question as to when to switch from 'restrictive monetory policy' to 'expansionist monetory policy' and vice-versa.

	RUN	WEIGHT	ASSMT	SAT	WFAT	TIPPROD	OE1	OE2	TIME	COST
Richardson and Pugh	1	-	6.0	6.000	3.000	6.000	1.500	-	45.75	6.563E+6
(	2	0.4	1.0	1.216	5.489	6.000	2.138	1.000	40.00	5.462E+6
(	3	0.45	1.0	1.000	5.055	6.000	2.271	1.000	40.00	5.416E+6
(	4	0.5	1.0	1.000	5.433	6.000	2.320	1.000	40.00	5.454E+6
Optimiser (	5	0.55	1.0	1.000	5.430	6.000	2.320	1.000	40.00	5.454E+6
(	6	0.6	1.0	1.000	10.000	3.000	1.000	2.116	40.00	5.569E+6
(	7	0.65	1.0	1.000	10.000	3.000	1.000	2.116	40.00	5.469E+6

TABLE 5. Results of Conventional and Optimisation Experiments for Structural Policy Design.

CONCLUSIONS

This paper has presented the rationale for, and a case study of the merits of, optimisation for policy design in system dynamics models. When implemented through a good computer software interface the procedure is entirely straightforward to perform. The results produced are, as in conventional system dynamics, totally explainable in terms of the underlying feedback structure of the model. However, the saving in conputational effort required by the analyst in producing policy insights is enormous. This is not, however, to say that the level of thinking is reduced. Rather this is increased, since considerable skill is necessary in formulating the model so as to provide the computer with the sufficient scope to generate the maximum amount of model exploration.

REFERENCES

Cavana, R. Y. and R. G. Coyle (1982). DYSMAP User Manual, University of Bradford.

Gustafsson, L. and M. Wiechowski (1986). Compiling DYNAMO and Optimisation Software. System Dynamics Review, Vol. 2, no. 1.

Keloharju, R. (1983). Relativity Dynamics. Helsinki School of Economics.

Mohapatra, P. J. K. and S. K. Sharma (1985). Synthetic Design of Policy Dynamics in System Dynamic Models: A Model control Theory of Approach. System Dynamics Review, Vol. 1, no. 1.

Richardson, G. P. and A. L. Pugh III (1981). Introduction to System Dynamics Modelling with DYNAMO.

Sharma, S. K. (1985). Policy Design in System Dynamics Models: Some Control Theory Applications. Doctoral Thesis. Indian Institute of Technology.

Appendix 1

```
* REVISED PROJECT MODEL
NOTE
NOTE  REAL PROGRESS
NOTE
L CRPRG.K=CRPRG.J+DT*RPRG.JK
N CRPRG=0
R RPRG.KL=APPRG.K*FSAT.K
A APPRG.K=WF.K*GPROD.K*ESPGP.K
A GPROD.K=NGPROD*EEXPGP.K
C NGPROD=1
A FSAT.K=NFSAT.K*EEXPFS.K*ESPFS.K
A NFSAT.K=TABHL(TNFSAT,FCOMP.K,0,1,.2)
T TNFSAT=.5/.55/.63/.75/.9/1
NOTE
NOTE  EFFECTS OF EXPERIENCE AND SCHEDULE PRESSURE
NOTE
A EEXPGP.K=TABHL(TEEXPG,FEXP.K,0,1,.2)
T TEEXPG=.5/.55/.65/.75/.87/1
A FEXP.K=EXPWF.K/WF.K
A EEXPFS.K=TABHL(TEEXPF,FEXP.K,0,1,.2)
T TEEXPF=.5/.6/.7/.8/.9/1
A ESPFS.K=TABHL(TESPFS,ICD.K/SCD.K,.9,1.2,.05)
T TESPFS=1.1/1.06/1/.96/.9/.83/.75
A ESPGP.K=TABHL(TESPGP,ICD.K/SCD.K,.9,1.2,.05)
T TESPGP=.9/.92/1/1.1/1.18/1.23/1.25
NOTE
NOTE  UNDISCOVERED REWORK
NOTE
R GURW.KL=APPRG.K*(1-FSAT.K)
L URW.K=URW.J+DT*(GURW.JK-DURW.JK)
N URW=0
R DURW.KL=URW.K/TDRW.K
A TDRW.K=TABHL(TTDRW,FPCOMP.K,0,1,.2)
T TTDRW=12/12/12/10/5/.5
A CPPRG.K=CRPRG.K+URW.K
A FPCOMP.K=CPPRG.K/CPD.K
A CPD.K=TABHL(TCPD,FCOMP.K,0,1,.2)
T TCPD=800/830/900/1000/1140/1200
A FCOMP.K=CRPRG.K/1200
NOTE
NOTE  EFFORT PERCEIVED REMAINING
NOTE
A EPREM.K=OE1*(CPD.K-ACPRG.K)/PPROD.K
C OE1=1
A ACPRG.K=CPPRG.K-AURW.K
A AURW.K=ADURW.K*ATDRW.K
A ADURW.K=SMOOTH(DURW.JK,TADURW)
C TADURW=8
A ATDRW.K=TABHL(TATDRW,FPCOMP.K,0,1,.2)
T TATDRW=8/8/7/5/3/1.5
L PPROD.K=PPROD.J+(DT/TPPROD)(IPROD.J-PPROD.J)
N PPROD=GPROD
A IPROD.K=WTRP.K*RPROD.K+(1-WTRP.K)*NGPROD
C TPPROD=6
A WTRP.K=TABHL(TWTRP,FPCOMP.K,0,1,.2)
T TWTRP=0/.1/.25/.5/.9/1
A RPROD.K=NGPROD*FSAT.K
```

```
NOTE
NOTE  HIRING
NOTE
A WF.K=EXPWF.K+NEWWF.K
L EXPWF.K=EXPWF.J+DT*WFAR.JK
N EXPWF=EXPWFN
C EXPWFN=2
R WFAR.KL=NEWWF.K/ASMT
C ASMT=6
L NEWWF.K=NEWWF.J+DT*(HR.JK-WFAR.JK)
N NEWWF=NEWWFN
C NEWWFN=1
R HR.KL=(WFS.K-WF.K)/WFAT
C WFAT=3
A WFS.K=WCWF.K*IWF.K+(1-WCWF.K)*WF.K
A WCWF.K=TABHL(TWCWF,TREM.K,0,21,3)
T TWCWF=0/0/0/.1/.3/.7/.9/1
A IWF.K=OE2*EPREM.K/TREM.K
C OE2=1
NOTE
NOTE  SCHEDULING
NOTE
A TREM.K=SCD.K-TIME.K
L SCD.K=SCD.J+DT*NAS.JK
N SCD=SCDN
C SCDN=40
R NAS.KL=(ICD.K-SCD.K)/SAT
C SAT=6
A ICD.K=TIME.K+TPREQ.K
A TPREQ.K=EPREM.K/WFS.K
NOTE  OBJECTIVE FUNCTIONS
NOTE
NOTE  INDICATORS
NOTE
L CUMEFF.K=CUMEFF.J+DT*(WF.J*ESPGP.J)
N CUMEFF=0
A COST.K=CPMM*CUMEFF.K
C CPMM=3000
NOTE
NOTE  ******EQUATIONS FOR OPTIMISATION ***************
NOTE                                                 *
A AUX1.K=CLIP(TIME.K,0.25,CRPRG.K,CPD.K)             *
A AUX2.K=SAMPLE(AUX1.K,AUX1.K,0)                     *
A AUX3.K=CLIP(100,0.25,AUX1.K-AUX2.K,1)              *
A AUX4.K=SAMPLE(AUX2.K,AUX3.K,0)                     *
A AUXC1.K=SAMPLE(COST.K,AUX1.K,0)
A AUXC2.K=SAMPLE(AUXC1.K,AUX3.K,0)
A AUX5.K=MAX(40-AUX4.K,0)                            *
C INIT=0                                             *
C INCR=0.01                                          *
A TARG.K=(1+INCR)*INIT                               *
A OBJ1.K=WEIGHT*(1E+5)*AUX4.K+(1-WEIGHT)*AUXC2.K+    *
X (1E+5)*AUX5.K                                      *
A OBJ2.K=MAX(AUX4.K-TARG.K,TARG.K-AUX4.K)            *
C WEIGHT=0.5                                         *
NOTE                                                 *
NOTE  ************************************************
NOTE
NOTE CONTROL STATEMENTS
NOTE
C DT=0.25
C PLTPER=2
C PRTPER=40
C LENGTH=80
PRINT 1)WF,SCD,OBJ2/2)CUMEFF,FCOMP,FPCOMP/3)COST,OBJ1,
X AUXC1/4)AUXC2,AUX1,AUX2/5)AUX3,AUX4,AUX5
PLOT WF=W(0,80)/SCD=S(30,70)/CPPRG=P,CRPRG=R,URW=U(0,1200)/
X PPROD=*(.6,1)
RUN
+
```