Improving Computer-Aided Process Plant Scheduling and Design

By

James Duncan Bloom

December 2001

Supervised by

Prof Nilay Shah

Department of Chemical Engineering and Chemical Technology,

Imperial College of Science, Technology and Medicine, London, SW7 2BY,

Abstract

This report is concerned with efficient scheduling and design for process plant. The design is dependant on future schedules for a given process plant and must therefore be considered simultaneously if the optimum design is to be found. Every process has trade-offs between capital costs, revenues and operational flexibility, there must be justification for choosing one design solution over another. The optimum design is determined subject to some desired criterion such as maximization of profit or minimization of costs.

Pantelides (1994) proposed a mixed integer linear programming (MILP) formulation that allowed detailed consideration of the design problem taking in account the selection of the required processing and storage equipment items as well as the required levels of provisions of other production resources such as utilities, labor, cleaning, and transportation. The MILP was based on a detailed discrete-time Resource-Task Network (RTN). The RTN framework is a very general representation, which allows a wide range of different process problems to be tackled.

The need to model the system in detail often leads to large-scale process models. The model size is further increased if there are decisions on widely differing time-scales for example planning horizons of weeks with process decisions in hours, or if the problem involves many different resource types. A detailed formulation may be too large to be computationally tractable, therefore a general method for reducing the size has been proposed by Wilkinson (1996) resulting in a smaller aggregate formulation derived directly from the detailed formulation via the mathematical manipulation of the RTN formulation.

In this report, general Batch Scheduling System was introduced as scheduling and design software for process plants. Although this software works well there are substantial limitations. The time to generate a solution is slow, and a special language has to been learnt to enter models into gBSS. Wilkinson's aggregation was introduced as a method of reducing the model size and therefore reducing the time to generate a solution. A Graphical User Interface (GUI) was introduced as an alternative to the gBSS modelling language.

Table of Contents

Abstract **2**

Table of Contents **3**

1. Introduction **3**

Scheduling 3

Design 3

Benefits of Scheduling and Design 3

Outline of Report 3

2. State-Task Network Process Diagrams **3**

Hydrolubes Plant Example 3

3. The STN Framework for Mathematical Process Modelling **3**

Assumptions 3

Time Representation 3

Model Variables 3

Model Equations 3

The Objective Function *3*

Model Constraints *3*

4. general Batch Scheduling System **3**

Input Files 3

Mixed Integer Linear Programming Problems 3

Branch-and-Bound algorithms for the solution of MILPs 3

Output Files 3

gBSS Limitations 3

5. Graphical User Interface for gBSS **3**

GUI Conclusions 3

6. Resource-Task Networks **3**

Hydrolubes Plant Example 3

7. The RTN Framework for Mathematical Process Modelling **3**

Assumptions 3

Model Variables 3

Model Equations 3
The Objective Function *3*
Model Constraints *3*
8. Aggregation 3
Time Representation 3
Aggregate Constraints 3
Weighting Functions 3
Aggregate Variables 3
First Order Aggregated RTN Formulation 3
The Objective Function *3*
Model Constraints *3*
9. Assessment of Aggregate Solution Method 3
Method 3
Results 3
Conclusion 3
10. Conclusions and Further Work 3
Potential Directions for Further Work 3
11. References 3
12. Appendix A, gBSS Input Files for Hydrolubes Plant 3
Problem File; Hydro.prb 3
Recipe Files; Hydro.stn 3
Resource File; Hydro.uni 3
13. Appendix B, gBSS Output Files for Hydrolubes Plant 3
Gantt Chart 3
Hydrolubes gBSS Output File; Hydro.out 3
14. Appendix C, GAMS Input Files for Hydrolubes Plant 3

1. Introduction

Scheduling

Scheduling in process plants is a very difficult and complex task. It involves making decisions on hundreds if not thousands of tradeoffs regarding the allocation of production resources, such as utilities or processing equipment. The complexity arises mainly from the multitude of different possibilities for production routes. An analogy can be drawn with the production of portions of Spaghetti Bolognaise in your kitchen at home. Not only are there different variations of the recipe but also many different ways to follow each variation. The ingredients could be freshly made, prepared earlier and stored or bought pre-made. A schedule for the production of Spaghetti Bolognaise would be a set of instructions detailing exactly how you would use the different equipment items in the kitchen to produce portions of Spaghetti Bolognaise. The schedule would include exact timings and quantities for each task.

Design

Design in process plants involves the selection of the number and type of equipment items that need to be incorporated in the plant. A design for the production of portions of Spaghetti Bolognaise would include the selection of the correct equipment items and their capacities. It could be reasonably assumed that the kitchen will have at least one cooker hob, but it may be more advantageous to have several cooker hobs if multiple smaller batches are to be performed rather than a few large batches. Due to the flexibility of equipment types and sizes operational scheduling considerations need to be taken into account even at the design stage. If an optimum processes are to be achieved, scheduling has to be an integrated part of design.

Benefits of Scheduling and Design

Computer-aided scheduling or design is based on the optimisation of an objective function. This type of approach allows the minimization or maximization of a number of different objectives such as maximum profit, or maximum number of completed customers orders. The benefits of these objectives are clear, however there are also some less obvious benefits. Efficient scheduling can help a company to change its strategy, for example a shift in production practices from the ‘just-in-case’ approach to the ‘just-in-time’ approach can reduce the money tied up in raw materials and unsold products. The main benefits of scheduling and design are explained in more detail below:

1. Maximizing Profits

   An objective function could be modelled to produce an optimum scheduling with respect to profit such as maximum, annual revenue minus annual running costs. In a design model the maximized objective would be annual revenue minus annual running costs minus annualised capital expenditure.

2. Increasing Customer Satisfaction

   Always meeting your customer’s orders is a very effective marketing tool. Customers who rely on produces being delivered are likely to commend suppliers who meet their demands. This can be achieved by formulating an objective function that minimizes late deliveries. It is even possible to model soft and hard deliveries times with product values being a function of time past the soft deadline and falling to zero after the hard deadline.

3. Reduced Stock

   Thirty years ago manufacturing usual employed a ‘just-in-case’ approach where managers often required large stock inventories to allow for inefficiencies. The use of scheduling or design allows an optimum process that can be operated under a ‘just-in-time’ approach. The advantage of this is to reduce the inventory of stock and unsold products therefore reducing the sunken capital tied up in the process, allowing it to be free for alternative investments.

4. Product Diversification

   Customers often require products to match their individual requirements. Optimum scheduling improves a plant’s capability to produce a wider range of similar products to meet the exact customer specification such as lubricants which have common ingredients blended in different quantities with various additives to produce many different products.

5. Improves Multisite Production

   An optimum schedule can increase efficiency of a multisite production system by calculating the correct trade off between economies of scale, encouraging production of each chemical on one plant, and cost of transportation, encourage each plant to produce its own chemicals.

Outline of Report

The objective of this report is to outline methods to improve process plant scheduling and design software. Particular attention is paid to general Batch Scheduling Systems, which has been under development, by the Centre for Process Systems Engineering since 1988. The State-Task Network and its accompanying framework for mathematical process modelling of Kondili *et al* (1988) are introduced in chapter 2 and 3. general Batch Scheduling System and its limitation are discussed in chapter 4 with techniques to improve the software being introduced in chapters 5 and 8. The more modern Resource-Task Network framework for mathematical process modelling proposed by Pantelides (1994) is introduced in chapters 6 and 7 to explain aggregate process models of Wilkinson (1996) that are discussed in chapter 8. Finally, chapter 10 summarizes the important issues raised in the report as well as indicating areas for further research. Through out most chapters in this report reference is made to the same process example called the Hydrolubes Plant, the example has been taken from Shah N. (2001).

2. State-Task Network Process Diagrams

The State-Task Network (STN) of Kondili *et al* (1988) was developed to represent the complexity in process recipes, and is a useful tool to enable reliable building of process models. In an STN the product of one task may be used as the input of one or more other tasks. The tasks shown as rectangles correspond to individual processing steps, which may comprise not only conventional chemical or physical unit operation such as reaction or blending, but also packaging operations such as loading and unloading of material from transport facilities. The states shown as circles correspond to material at various stages of processing i.e. raw materials, intermediates, or final products.

It is important to note that the STN only describes the states and the tasks that transform input states to output states, it does not contain any information on the processing equipment or other resources such as manpower or electricity. This type of information is included in the more modern Resources-Task Network proposed by Pantelides (1994), which is introduced in chapters eight and nine.

Hydrolubes Plant Example

Taken from the Flexible Plant Scheduling course notes of Prof Shah (2001).

Recipe

Four products BlendA, Prod1, Prod2, Prod3 are to be produced from six feedstocks FeedA, FeedB, FeedC, FeedD, Add1 and Add2 according to the following recipe:

Table 1, Hydrolubes Plant Processing Tasks

Task	Inputs, mole fractions	Outputs, mole fractions	Processing Time, hr	Operators Required (1st hour)
Reaction	FeedA	ReactProd	5	1
BlendingA	ReactProd (99.9%) + Add1 (0.1%)	BlendA	2	0.5
BlendingB	ReactProd (99.9%) + Add2 (0.1%)	Int1	2	0.5
Mixing1	Int1 (38%) + FeedB (62%)	Prod1	2	0.5
Mixing2	Int1 (40%) + FeedC (60%)	Prod2	2	0.5
Mixing3	Int1 (53%) + FeedD (47%)	Prod3	2	0.5

State Task Network

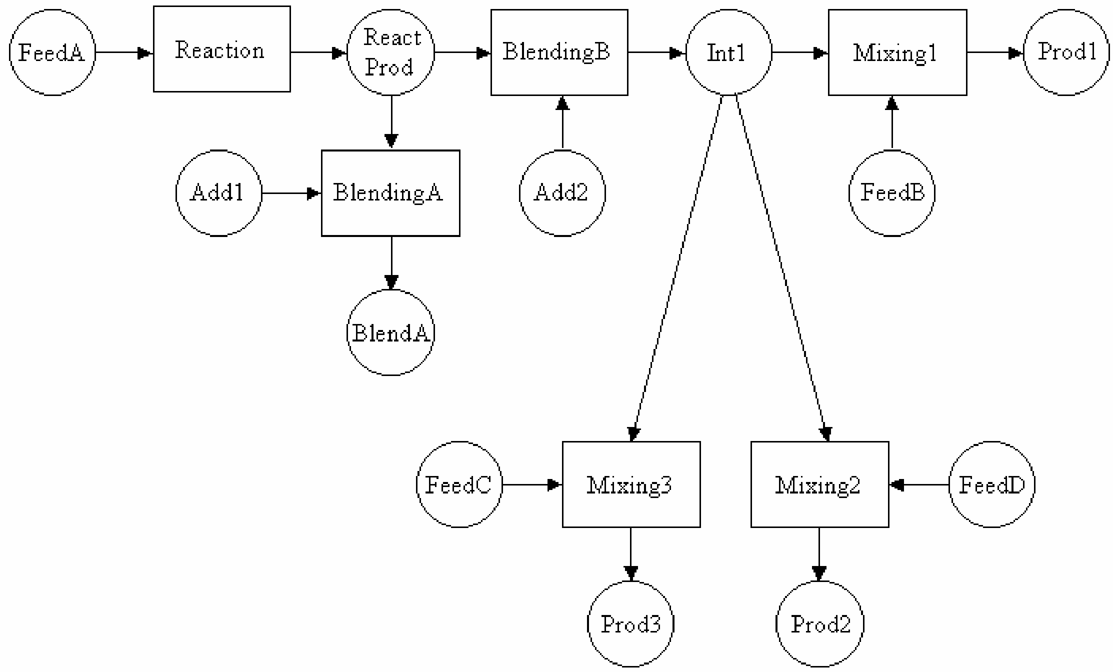


Figure 1, Hydrolubes Plant State-Task Network

Available Resources

There are ten operators available at all times who can operate the following units:

Table 2, Hydrolubes Plant Processing Resources

Unit Type	Number Available	Size, te (each)	Suitable for
Reactor	1	50	Reaction
Blender	2	45	BlendingA, BlendingB
Mixer	3	45	Mixing1, Mixing2, Mixing3
Intermediates Storage	1	75	Int1
	None	None	ReactProd
Other Storage	Unlimited	Unlimited	Feeds, Products

Product Requirements

The plant operations are to be scheduled over 4 shifts of 8 hours each, giving a 32-hour time horizon. The product demands and their values are:

Table 3, Hydrolubes Plant Product Requirements

Product	Time, hr	Amount, kg	Value, £/kg
BlendA	22	40	15
	32	46	15
Prod1	10	20	20
	16	25	20
	22	25	20
	32	30	20
Prod2	10	25	20
	16	30	20
	22	30	20
	32	35	20
Prod3	10	30	20
	16	35	20
	22	35	20
	32	40	20

Initial Stock

Table 4, Initial Stock

Raw Material	Amount, kg
FeedA	200
FeedB	200
FeedC	200
FeedD	200
Add1	20
Add2	20
All other materials	0

3. The STN Framework for Mathematical Process Modelling

To produce a schedule or design, a model of the process in question needs to be developed. Developing a model involves the formulation of a set of mathematical expressions that mimic the behaviour of the process, one such model is the STN framework, which was developed by Kondili *et al* in 1988.

Assumptions

- The process recipes are fixed and independent of the batch-size including input fractions, output fractions and processing time for every task.
- Once a task starts, it continues to completion.
- Material is transferred instantaneously from one unit to another at the start or finish of each task.

Time Representation

A truly rigorous mathematical model developed would have a continuous timescale enabling tasks to start and finish at any point. The solution of this type of model would require a great deal of computation and is therefore rendered intractable. To reduce the computation required, a discrete representation of time is used in the STN framework. Consider the time horizon of interest to be discretised into a number of time intervals of equal duration.

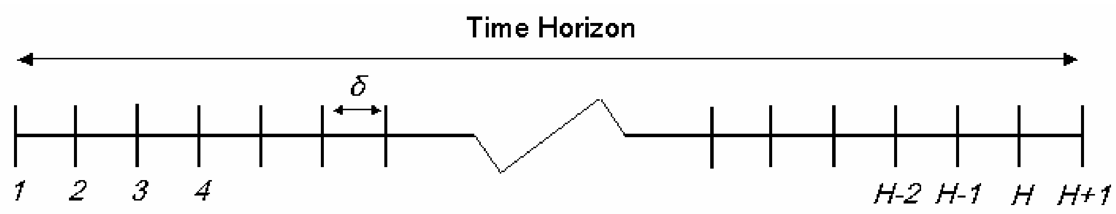


Figure 2, Time Horizon Discretisation

There are H intervals, numbered from 1 to H each of duration δ. This approach imposes the restriction that most events occurring in the plant, such as the start or finish of any task, must occur at the start or end of each interval. As the model is required to mimic the process accurately it might appear sensible to chose a small value of δ allowing the process tasks to start and finish more freely due to the increased number of intervals. In fact the size of δ is inversely proportional to the model size therefore the correct value of δ is the greatest common divisor of all the processing times for each task. In the Hydrolubes Plant the greatest common divisor of the processing times is 1 hour and the time horizon is 32 hours therefore $\delta = 1$ and $H = 32$.

Model Variables

The model variables represent all tasks, materials, and utilities in each time interval. All variables can take non-negative continuous values except W_{ijt}, which is an integer variable that can only take values of 1 or 0.

W_{ijt} = 1 if equipment item j starts processing task i at the beginning of time interval t and 0 otherwise

B_{ijt} = quantity of material undergoing task i in the equipment item j at the beginning of time interval t

S_{st} = amount of material in state s held in storage over time interval t

R_{st} = amount of material in state s received from suppliers at beginning of time interval t

D_{st} = amount of material in state s delivered to customers at beginning of time interval t

U_{st} = total demand placed on utility u over time interval t

Subscripts:

i for tasks, j for units, t for time period, θ for time relative to start of task, s for states and u for utilities

Other Notation:

C_S = maximum storage capacity dedicated to state *s*

C^U_{ut} = unit cost of utility u at time *t*

H = number of intervals in time horizon

I_j = set of processing tasks which can be performed by unit *j*

K_i = set of units capable of performing task *i*

p_i = processing time of task *i* $\left(\equiv \max_{i \in \bar{S}_i} \; p_{is} \right)$

p_{is} = processing time for the output of task *i* to state $s \in \bar{S}_i$

$\bar{S}_i$ = set of states produced by task *i*

T_s = set of tasks requiring material from state *s*

$\bar{T}_s$ = set of tasks producing material in state *s*

U^{max}_{ut} = maximum available level of utility *u* over time interval *t*

V^{max}_{ij} = maximum capacity of unit *j* when used for performing task *i*

V^{min}_{ij} = minimum capacity of unit *j* when used for performing task *i*

Model Equations

The Objective Function

The objective function of the model is the equation that maintains the model objective for example in a design model the objective function is an economic performance criterion that maximises the annual revenue minus annual running costs minus annualised capital cost. The objective function is maximized or minimized subject to all the process constraints being satisfied. The process constraints limit the values the variables that represent the process can take enabling the solution of the model to represent a feasible process schedule or plant design. There are six sets of process constraints. The objective function:

$$Maximise \sum_s \pi_s \left(S_{s,H+1} - S_{s,0} \right) - \sum_u \sum_{t=1}^{H} C^U_{ut} U_{ut} \qquad [3.0]$$

Model Constraints

1. Process equipment allocation constraints

This constraint guarantees that an item of equipment never performs more than one task at any time. It prevents new tasks from starting in an item of equipment if it is still performing any task. If task i' is being performed in equipment item j which has processing time $p_{i'}$ then a new task i cannot be performed until task i' has finished. If I_j is the sets of tasks for which equipment item j is suitable then this constraint can be mathematically modelled as follows:

$$W_{i'j,t-\theta} + \sum_{i \in I_j} W_{ijt} \leq 1 \qquad \forall\, j,t,i' \in I_j, \quad 1 \leq \theta \leq p_{i'} - 1 \qquad [3.1]$$

2. Process equipment capacity constraints

This constraint prevents B_{ijt}, the amount of material undergoing task i in equipment item j, from exceeding the vessel's maximum capacity V_j^{max}. As many equipment items also require a minimum capacity to enable proper functioning, this constraint also includes the vessel's minimum capacity V_j^{min}.

$$V_j^{min} W_{ijt} \leq B_{ijt} \leq V_j^{max} W_{ijt} \qquad \forall\, j,i \in I_j, t \qquad [3.2]$$

3. Storage capacity constraints

The amount of material held in storage can never exceed the storage capacity or be negative.

$$0 \leq S_{st} \leq C_s \qquad \forall\, s,t \qquad [3.3]$$

4. Material balance constraints

A material balance is performed on all states. The amount of state s in time interval t must equal the amount of state s in time interval $t-1$ minus the amount consumed, T_s, or delivered to customers, D_{st}, plus the amount produced, $\overline{T}_S$, or received from suppliers, R_{st}.

$$S_{st} = S_{s,t-1} + \sum_{i \in \overline{T}_s} \sum_{j \in K_i} \overline{\rho}_{is} B_{ij,t-p_{is}} + R_{st} - \sum_{i \in T_s} \sum_{j \in K_i} \rho_{is} B_{ij,t-p_{is}} - D_{st} \qquad \forall s,t \qquad [3.4]$$

5. Utility constraints

Utility usage often changes during a task for example an exothermic reaction task may initially require steam to raise the reactor to the operating temperature, once the correct temperature is reached cooling water maybe be required to maintain an isothermal reaction. Utilities are therefore modelled with two variables that depend on θ the time elapsed since the start of the task. $\alpha_{ui\theta}$ is the fixed amount of utility used at time θ from the start of task i, and $\beta_{ui\theta}$ is the batch-size dependant utility used at time θ from the start of task i. Therefore, the amount of utility u consumed by task i during time interval θ from the start of the task is given by:

$$\alpha_{ui\theta} W_{ij,t-\theta} + \beta_{ui\theta} B_{ij,t-\theta} \qquad [3.5]$$

This leads to the total utility used in time t:

$$U_{ut} \equiv \sum_{i} \sum_{j \in K_i} \sum_{\theta=0}^{p_i-1} \left(\alpha_{ui\theta} W_{ij,t-\theta} + \beta_{ui\theta} B_{ij,t-\theta} \right) \qquad \forall u,t \qquad [3.6]$$

Which is constrained by the maximum amount of utility available as follows:

$$0 \le U_{ut} \le U_{ut}^{max} \qquad [3.7]$$

6. Equipment unavailability constraints

If an equipment item is unavailable between time interval t_1 and time interval t_2 task i is not able to start in that equipment item not only in the time period t_1 *to* t_2 but also in any time period between t_1-p_i+1 and t_2, where p_i is the processing time for task i. This is modelled by the following constraint:

$$W_{ijt} = 0 \qquad \forall i \in I_j, t_1 - p_i + 1 \le t \le t_2 - 1 \qquad [3.8]$$

4. general Batch Scheduling System

Process scheduling and design requires large amounts of data regarding the process recipe, the plant equipment, and the demands imposed on the plant. Even once all the necessary data has been collected and the optimisation object has been decided, the correct formulation of the equations then subsequent solution is a very difficult task. For this reason general Batch Scheduling System software also called gBSS has been under development in the Centre for Process Systems Engineering at Imperial College since 1988. The fundamental aim of gBSS is to address the difficulties in scheduling and design while shielding the user from the complexity of the underlying mathematics.

Input Files

The plant information is input into gBSS in a format that is based on STN diagrams. There are three input files a recipe file, a resource file, and a problem file. The recipe file, which is based on the STN, gives details of all the material states and processing tasks. The resource file specifies the available processing and storage units. The problem file contains the information that is expected to change, such as product demands and their values as well as deliveries or initial quantities of feed. The three gBSS design input files for the Hydrolubes plant example are in Appendix A page 3. As gBSS reads the input files it makes several consistency checks for example making sure input fractions to tasks add up to one. When the consistency checks are complete and successful the problem is posed as a mixed integer linear programming (MILP) problem.

Mixed Integer Linear Programming Problems

It can be verified that all constraints and the objective function are linear with respect to the unknowns leading to a linear optimisation problem. In general, linear optimisation problems can be solved without too much difficulty using linear programming techniques. However, what complicates the solution of this particular problem is the requirement that W_{ijt} must only take integer values of 0 or 1. Therefore, instead of a simple linear program called an LP, we have a MILP. A rather naive way of solving an MILP is to fix all the integer variables to either 1 or 0 we would then have a set of resulting LPs. These could be readily solved, although some may turn out to be infeasible. Since there are only a finite number of integer combinations if all the LPs were solved and the best solution was selected, this would be the optimum solution to the MILP. In practice, there are 2^N combinations with N variables; this makes the approach too slow for all but the smallest problems.

Branch-and-Bound algorithms for the solution of MILPs

First, the integer variables are relaxed to continuous variables in the range 1 to 0. Then the problem is solved, if the integer variable values happen to fall on 0 and 1 the answer is found, but this is very uncommon instead several integer variables will have fractional values. The algorithm then tests the objective function with one of the fractional variables set to 0 this is compared to the objective function with the same variable fixed to 1, this process is called branching on a variable. The integer variable value is then fixed to the value which gives the best objective function, all solutions with the integer variable value that gives the worst objective function are then discarded. The algorithm then fixes the next variable to 0 and 1 and tests for the best solution, this continues until one of the solving criterions has been meet, such as all possibilities tested, first feasible solution found or 5% from fully relaxed non-integer solution. The main limitation with the Branch-and-Bound algorithm is that it does not remove sufficient integer possibilities therefore for large problems the solution times can still be hours or even days, rendering this method intractable for large problems.

Output Files

When gBSS has obtained the solution for the MILP two solution files are produced. One file contains a Gantt chart, which is a standard graphical representation of all the units in the process it shows when the different tasks start and finish in each unit. The other output file contains tables detailing the relevant information for the units, tasks, and materials, such as the number and capacity of different units for a design problem. Two example files from a design for the Hydrolubes Plant are in Appendix B on page 3.

gBSS Limitations

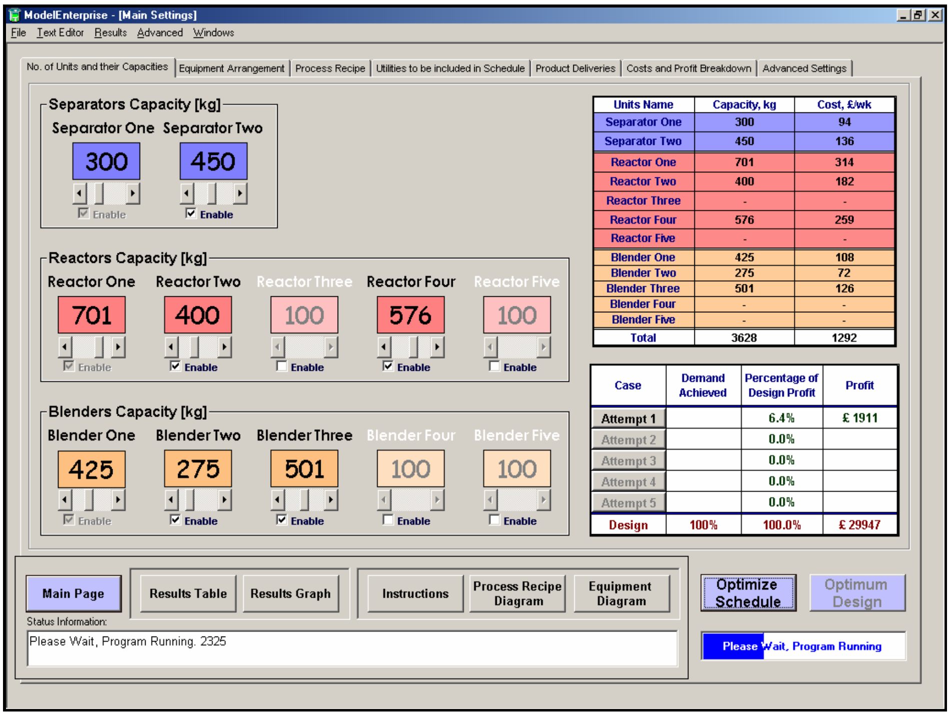
Although gBSS works well the solution process is very time consuming particularly for a design problem. The models are large and slow to solve for all but the smallest problems. Multisite planning for example may have a planning horizon of several months involving several plants with processing steps on a times-scale of less than one hour, this problem would result in an MILP involving tens of thousands of integer variables and would take current branch-and-bound algorithms days or weeks to solve.

There are three main methods to improve the solution speed of the MILPs produced by gBSS. The first and most obvious method is to increase the power or the number of the computers on which the MILPs are solved. Another method to reduce solution time is to incorporate knowledge of the process into the solver so that the number of LPs produced during the Branch & Bound solution can be reduced. The most general approach is to aggregate variables or constraints in the model to produce a smaller model that is faster to solve. The solution of the aggregate model is a close upper bound to the solution of the detailed model, for a model with a maximised objective function. Once the aggregate model has been solved certain variables such as number of equipment items can be fixed during the solution of the detail model resulting in a significant reduction in solution time for the detailed model, this is discussed further in chapter 9.

Although gBSS had its own modelling language to enter the process models, which is easier than the formulation of mathematical process equations, the inexperienced user has to spend considerable time learning how to write correct models in the gBSS modelling language. The development of a graphical user interface (GUI) would remove the need to learn the gBSS modelling language; this is discussed in the next chapter.

5. Graphical User Interface for gBSS

A windows based graphical user interface (GUI) was developed during the summer of 2001 to test methods of creating a demo to market gBSS. To create an effective demo it was essential to produce a GUI that is quick and easy to understand. The GUI was originally written in Visual Basic for Applications to run as a Microsoft Excel Macro, but after several weeks Visual Basic 6.0 was adopted to allow a more powerful GUI that could run on any Microsoft Windows Operating System. The GUI waits for the user to enter the correct information then after writing the gBSS input files run gBSS once gBSS had finished solving the MILP the results are then reported to the user in the form of tables and graphs. The GUI was tested on several users. During each test the user was asked to use the GUI with no help from the software designer. Any questions or difficulties the users had, while using the GUI, were documented as well as the suggestions comments made once the user had fully tested the GUI. The GUI had several rounds of testing each time the suggestions were used to improve the GUI then it was tested on a fresh new user.



Units Name	Capacity, kg	Cost, £/wk
Separator One	300	94
Separator Two	450	136
Reactor One	701	314
Reactor Two	400	182
Reactor Three	-	-
Reactor Four	576	259
Reactor Five	-	-
Blender One	425	108
Blender Two	275	72
Blender Three	501	126
Blender Four	-	-
Blender Five	-	-
Total	3628	1292

Case	Demand Achieved	Percentage of Design Profit	Profit
Attempt 1		6.4%	£ 1911
Attempt 2		0.0%	
Attempt 3		0.0%	
Attempt 4		0.0%	
Attempt 5		0.0%	
Design	100%	100.0%	£ 29947

Figure 3, GUI Equipment Selection

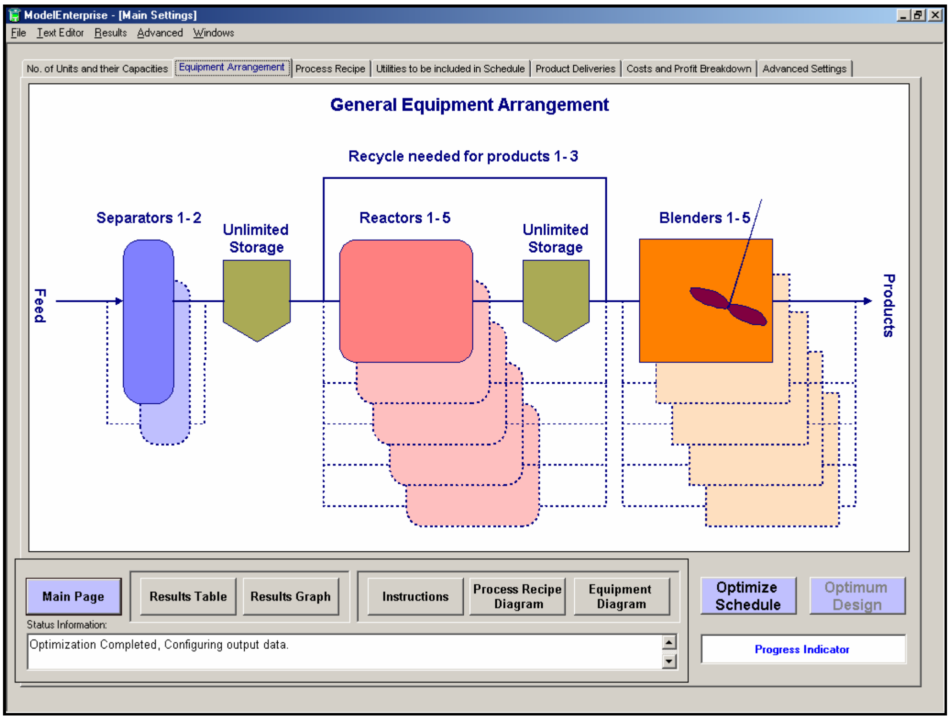


Figure 4, GUI Equipment Representation

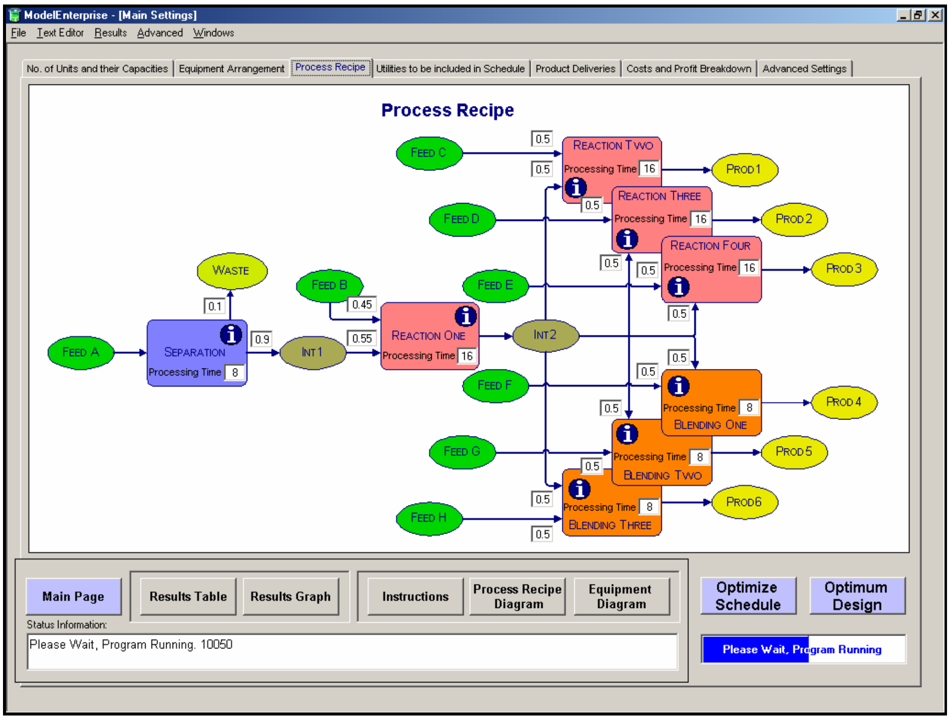


Figure 5, GUI Editable State-Task Network

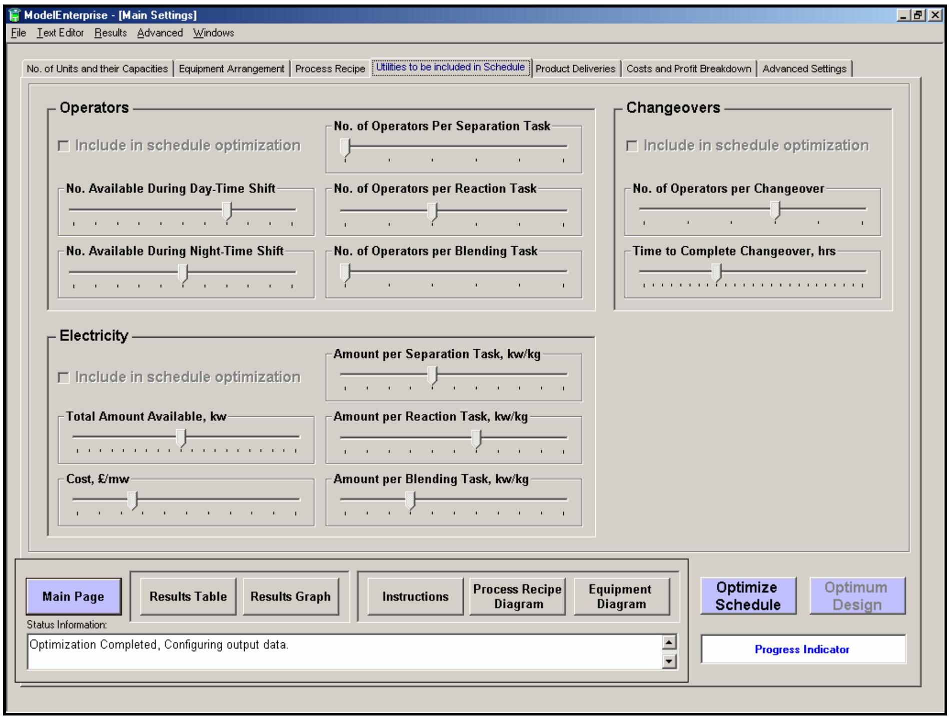


Figure 6, GUI Utility Settings

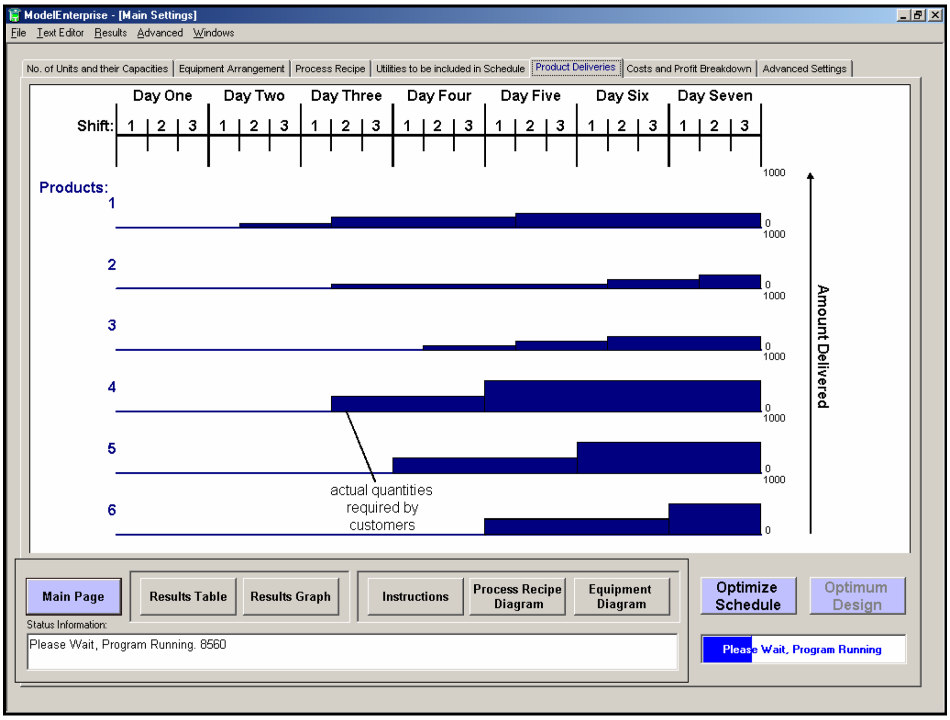


Figure 7, GUI Product Delivery

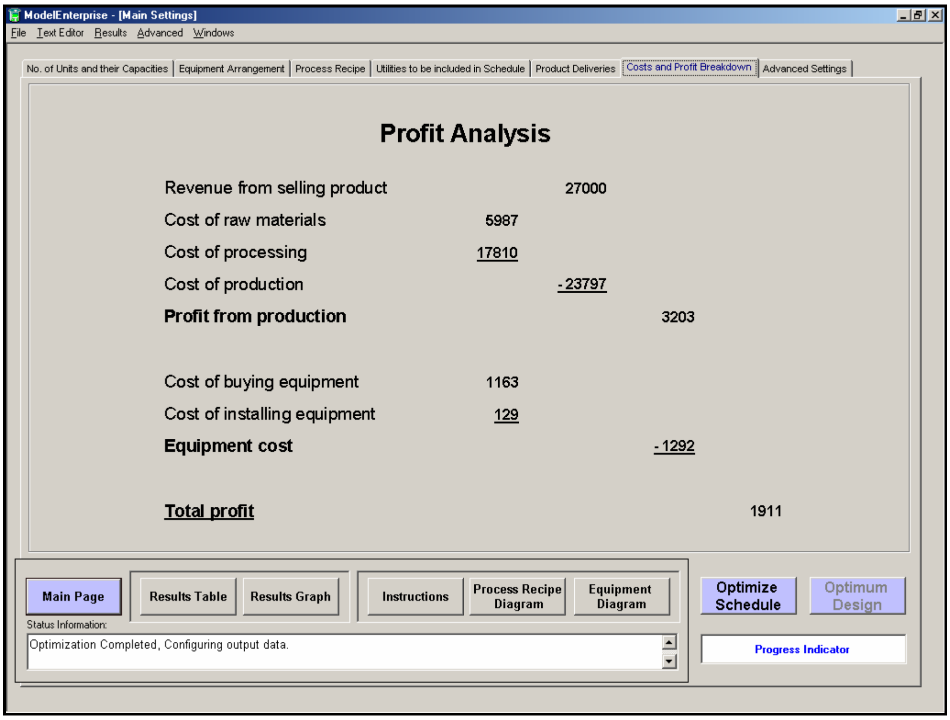


Figure 8, GUI Profit Analysis

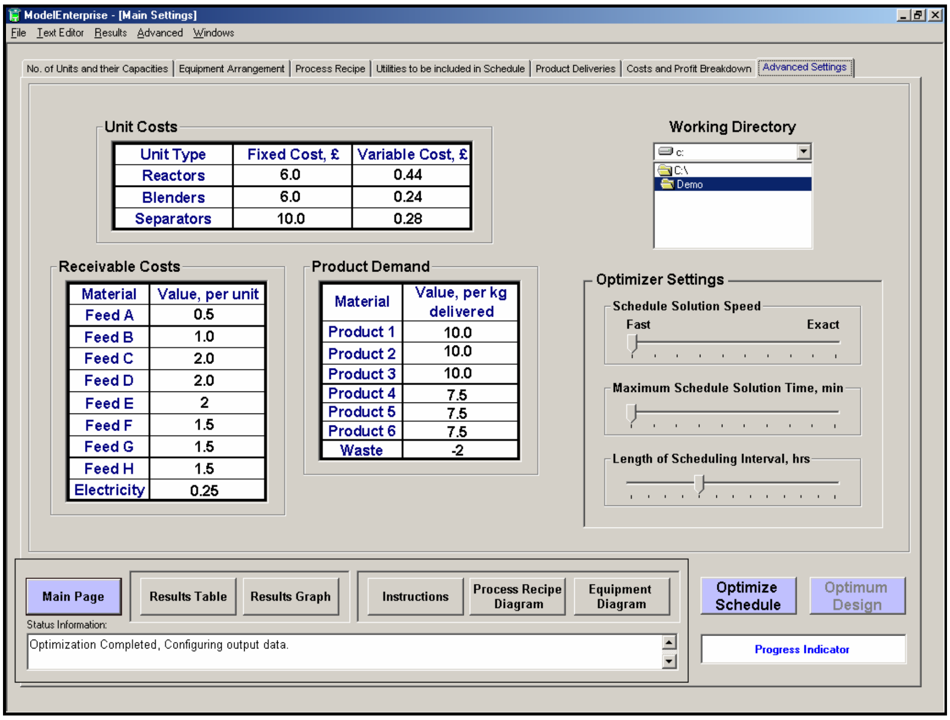


Figure 9, GUI Costs and Demands

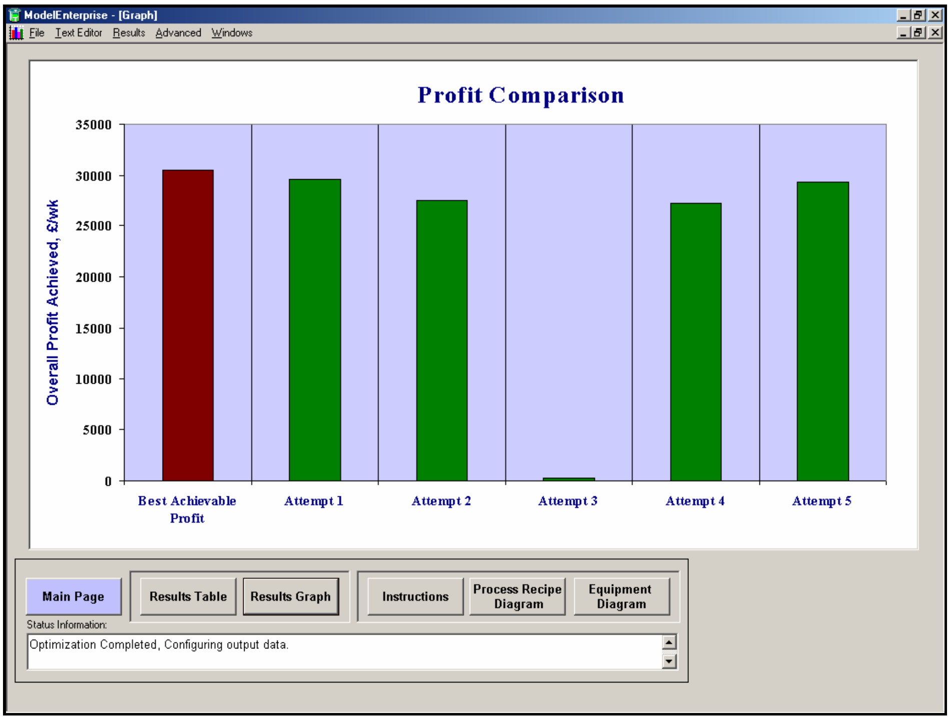
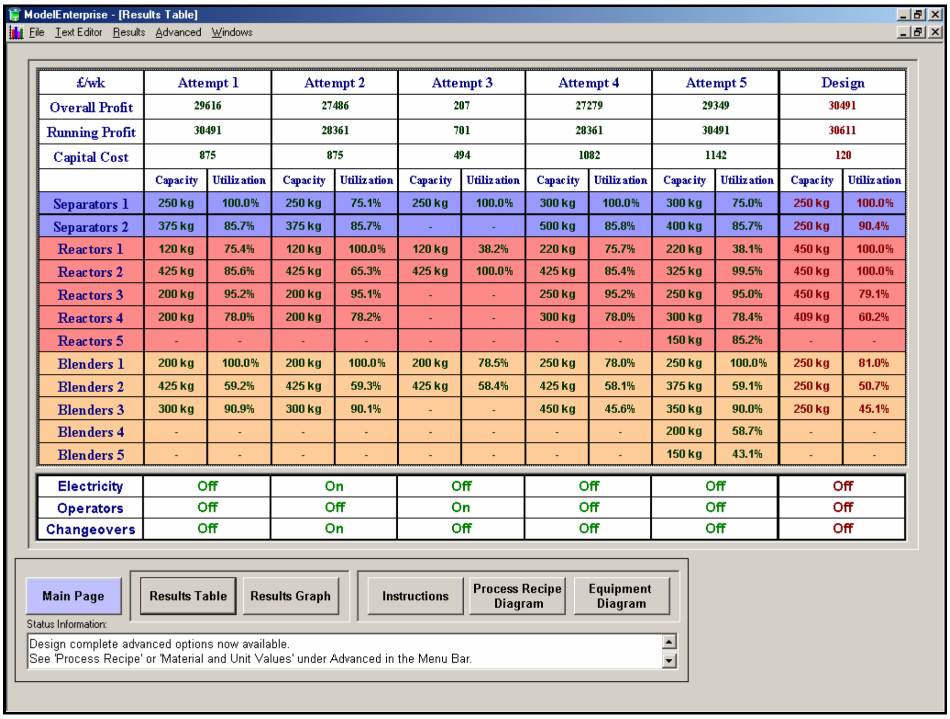


Figure 10, GUI Results Comparison



£/wk	Attempt 1		Attempt 2		Attempt 3		Attempt 4		Attempt 5		Design	
Overall Profit	29616		27486		207		27279		29349		30491	
Running Profit	30491		28361		701		28361		30491		30611	
Capital Cost	875		875		494		1082		1142		120	
	Capacity	Utilization	Capacity	Utilization	Capacity	Utilization	Capacity	Utilization	Capacity	Utilization	Capacity	Utilization
Separators 1	250 kg	100.0%	250 kg	75.1%	250 kg	100.0%	300 kg	100.0%	300 kg	75.0%	250 kg	100.0%
Separators 2	375 kg	85.7%	375 kg	85.7%	-	-	500 kg	85.8%	400 kg	85.7%	250 kg	90.4%
Reactors 1	120 kg	75.4%	120 kg	100.0%	120 kg	38.2%	220 kg	75.7%	220 kg	38.1%	450 kg	100.0%
Reactors 2	425 kg	85.6%	425 kg	65.3%	425 kg	100.0%	425 kg	85.4%	325 kg	99.5%	450 kg	100.0%
Reactors 3	200 kg	95.2%	200 kg	95.1%	-	-	250 kg	95.2%	250 kg	95.0%	450 kg	79.1%
Reactors 4	200 kg	78.0%	200 kg	78.2%	-	-	300 kg	78.0%	300 kg	78.4%	409 kg	60.2%
Reactors 5	-	-	-	-	-	-	-	-	150 kg	85.2%	-	-
Blenders 1	200 kg	100.0%	200 kg	100.0%	200 kg	78.5%	250 kg	78.0%	250 kg	100.0%	250 kg	81.0%
Blenders 2	425 kg	59.2%	425 kg	59.3%	425 kg	58.4%	425 kg	58.1%	375 kg	59.1%	250 kg	50.7%
Blenders 3	300 kg	90.9%	300 kg	90.1%	-	-	450 kg	45.6%	350 kg	90.0%	250 kg	45.1%
Blenders 4	-	-	-	-	-	-	-	-	200 kg	58.7%	-	-
Blenders 5	-	-	-	-	-	-	-	-	150 kg	43.1%	-	-

Electricity	Off	On	Off	Off	Off	Off
Operators	Off	Off	On	Off	Off	Off
Changeovers	Off	On	Off	Off	Off	Off

Figure 11, GUI Results Table

GUI Conclusions

The resulting program incorporated many diagrams and illustrations as it was found that most users found diagrammatic representations faster and easy to understand then written explanations. The GUI incorporated 26 diagrams and 8 tables many with values that could be edited or modified, such as the costs and demands windows on page 3 where the values or costs of equipment units, products and raw materials could be edited or the State-Task Network diagram window on page 3 where the task input and output fractions and processing times could be edited. A general GUI for scheduling or design software such as a GUI for gBSS should allow the entry of a process recipe by use of the State-Task Network or the Resource-Task Network[†]. This approach was not tested in the GUI developed in the summer of 2001 due to the time required to complete such as program, but should be the next stage for any further research into GUI development for scheduling or design software.

[†] The Resource Task Network is introduced in chapter 6

6. Resource-Task Networks

The Resource-Task Network (RTN) proposed by Pantelides (1994), which is similar in function but more modern than the STN, was developed to represent the complexity in process recipes, and is a useful tool to enable reliable building of process models. A Resource-Task Network (RTN) is diagrammatically similar to an STN but conceptually very different because, not only are the states and tasks included but also, all the resources involved in the process are included. No conceptual distinction is made between, for example, raw materials, processing equipment items, utilities and transportation resources. Consistent with the STN approach, RTN tasks are shown as rectangles corresponding to individual processing steps but instead of just consuming and producing states a task in an RTN consumes and produces resources shown as circles, these could be a material state or an equipment item's potential capacity for processing given tasks. Resources are classified into the minimum number of resource types such that any resource type is functionally equivalent to any other. Equipment items of different capacities, for example, are regarded as being in different resource types, even though they may be used for exactly the same transformations of materials. By adopting a uniform representation of all available resources, the RTN framework allows scheduling and design to consider complicated production features such as sequence dependent cleaning, or maintenance.

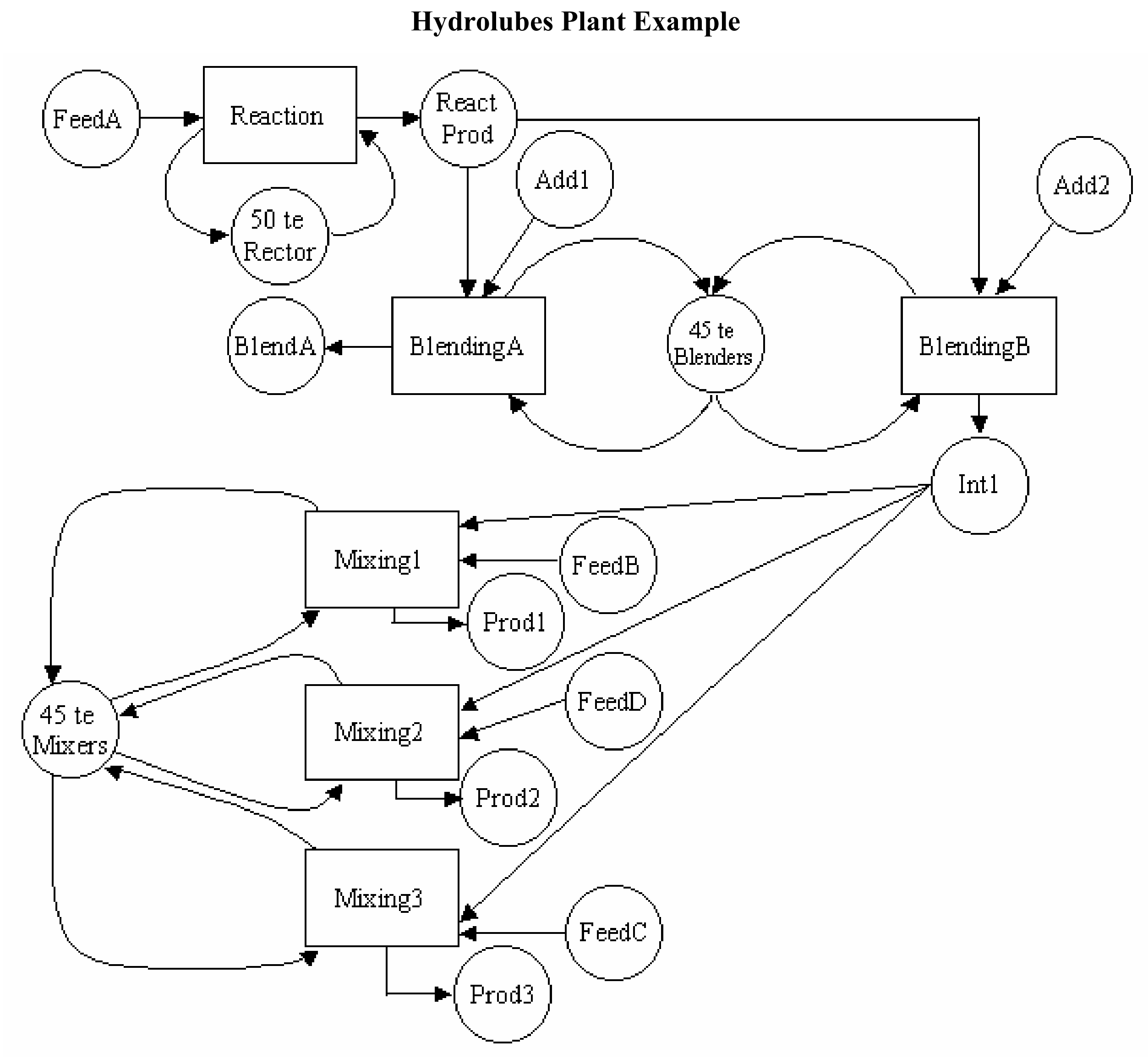


Figure 12, Hydrolubes Plant Resource-Task Network

7. The RTN Framework for Mathematical Process Modelling

To generate an aggregate formulation of the process first a detail model needs to been developed using the RTN formulation approach. Pantelides (1994) presented the RTN formulation based on a uniform, discrete representation of time that is identical to the time representation used for the STN formulation, see Figure 2on page 3.

Assumptions

The modelling assumptions are the same for the RTN formulation as for the STN formulation they are as follows:

- The process recipes are fixed and independent of the batch-size including input fractions, output fractions and processing time for every task.
- Once a task starts, it continues to completion.
- Material is transferred instantaneously from one unit to another at the start or finish of each task

Model Variables

The model variables represent all tasks and resources in each time interval. All variables have to take non-negative real values except N_{kt}, which is an integer variable that can only take values of 1 or 0. Each task starting at time t is characterised by:

N_{kt} = 1 if task k starts at the beginning of time interval t and 0 otherwise

ξ_{kt} = quantity of material undergoing task k at the beginning of time interval t

τ_k = processing time for task k

Subscripts:

k for tasks, r for resources, t for time period and θ for time relative to start of task

Other Notation:

C_k^F = fixed cost of task k

C_k^V = variable cost of task k

C_{rt}^E = cost or revenue from receiving or delivering resource r to external sources at time interval t

C_r^F = end of horizon value of resource r for $r \in MR$

C_r^{CAP} = capital cost of processing resource r for $r \in PE$

C_r^I = cost of storing resource r

MR = set of material resources

R_{rt} = excess (available) resource r in time interval t

R_r^{max} = maximum allowed amount of excess resource r

R_r^{min} = minimum allowed amount of excess resource r

V_k^{max} = maximum allowed size of task k

V_k^{min} = minimum allowed size of task k

PE_k = set of processing resources that can perform task k

$\mu_{kr\theta}$ = resource produced or consumed per instance of task k at time θ time relative to start of task k

$\nu_{kr\theta}$ = resource produced or consumed per unit amount of task k at time θ time relative to start of task k

Π_{rt} = resource produced or consumed per unit amount of task k at time θ time

Model Equations

The Objective Function

The objective function of the model is the equation that maintains the model objective typically maximising the total value-added at the end of the planning horizon. The objective function must be optimised subject to all the model constraints being satisfied. The objective function can take many forms depending on exactly which costs and revenues are to be included in the model for example:

$$max \sum_{r \in MR} \left(C_r^F \left(R_{rH} - R_{r0} \right) - C_r^I \sum_{t=1}^{H} R_{rt} \right) - \sum_k \sum_{t=1}^{H} \left(C_k^F N_{kt} + C_k^V \xi_{kt} \right) - \sum_{t=1}^{H} C_{rt}^E \Pi_{rt} \qquad [7.0]$$

Model Constraints

1. Excess Resource Balances

The amount of resource type r produced at time θ relative to the start of task k at time t is:

$$\mu_{kr\theta} N_{k,t-\theta} + \nu_{kr\theta} \xi_{k,t-\theta} \qquad [7.1]$$

As with the STN formulation the RTN formulation allows tasks with time-dependent utilities by the use of θ, the time relative to the start a task. All non-utility resources can only changes value at interval boundaries.

The resources in an RTN formulation are model by considering the excess resource, which is the amount not involved in active tasks. The change in excess resource from one interval to the next is called the excess resource balance. If Π_{rt} is the amount or resource r made available form external sources at time interval t the resource balance can be mathematical represented as follows:

$$R_{rt} = R_{r,t-1} + \sum_k \sum_{\theta=0}^{\tau_k} \left(\mu_{kr\theta} N_{k,t-\theta} + \nu_{kr\theta} \xi_{k,t-\theta} \right) + \Pi_{rt} \qquad \forall\, r,t \qquad [7.2]$$

The amount of excess resource is constrained by different factors in the plant such as whether there is adequate storage, this can be written as follows:

$$R_r^{min} \le R_{rt} \le R_r^{max} \qquad \forall\, r,t \tag{7.3}$$

2. Capacity and Batch-Size Constraints

These constraints ensure that the amount processed for a given task k is greater than any equipment item's minimum capacity V_k^{min} that will process task k and less than any equipment item's maximum capacity V_k^{max} for a given resource $r \in PE_k$ where PE_k is the set of processing equipment resource types that can perform task k.

$$V_k^{min} N_{kt} \le \xi_{kt} \le V_k^{max} N_{kt} \qquad \forall\, k,t,r \in PE_k \tag{7.4}$$

8. Aggregation

Aggregation provides a method for producing a MILP that involves fewer integer variables but still provides enough detail to produce an optimal or near-optimal solution. An aggregation operator is used to derive a set of aggregate scheduling or design formulations of increasing accuracy based on the RTN formulation of Pantelides (1994). Generating the aggregate formulations involves replacing groups of related variables in the RTN formulation with aggregate variables to reduce the size of the MILP and therefore the solution time. The approach adopted in this report is to use temporal aggregation taken from Wilkinson S.J. Ph.D. (1996), which involves grouping variables in the RTN formulation by their time intervals.

Time Representation

Temporal aggregation involves splitting the planning horizon of length H into n smaller portions called Aggregated Time Periods or ATPs. Each ATP has h original time intervals from the detailed RTN formulation. A time-weighted summation is performed for the detailed variables, from the RTN formulation, in each ATP giving rise to a set of aggregated variables. Tasks that start near the end of one ATP may continue in the next ATP these tasks give rise to linking variables that relate the ATP in which the tasks starts to the ATP in which the task finishes. A linking variable is the extent variable of a task that is being performed across an ATP boundary. Other constraints from the detailed RTN formulation such as capacity and batch-size constraints can also be summed over time in the same way to give constraints that involve the same aggregated and linking variables.

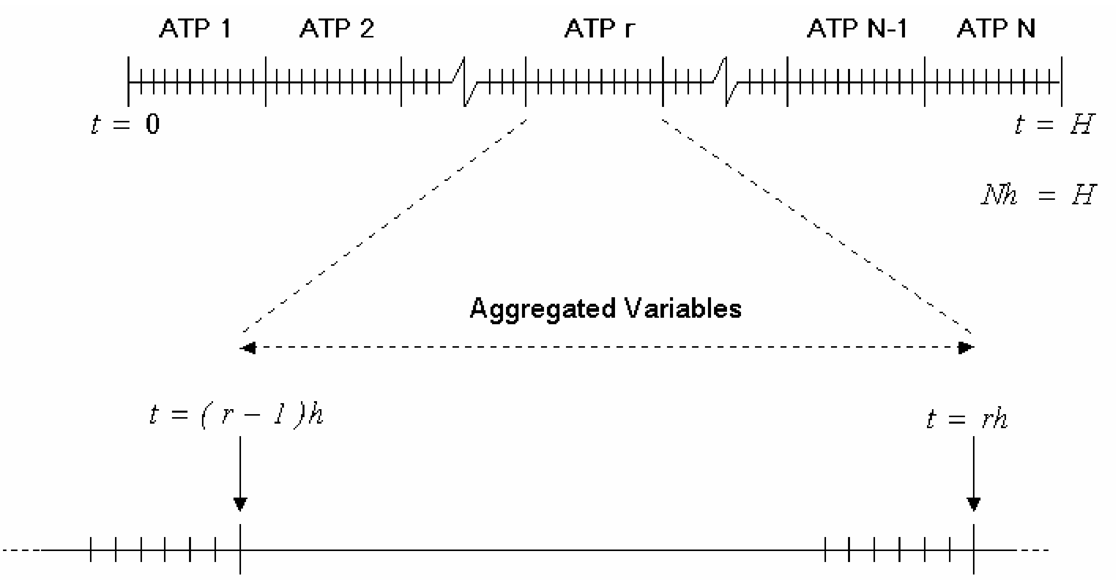


Figure 13, Aggregation of Planning Horizon

Aggregate Constraints

Consider a general inequality equation that holds for each time *t* in the planning horizon of length *H*:

$$C_t(\ .\ ) \leq 0 \qquad \forall\, t = h, 2h, \ldots, nh \quad (nh = H) \tag{8.0}$$

This inequality can be summed for *n* time intervals, from the detail formulation, in each ATP to obtain an aggregate inequality for each ATP.

$$\sum_{t'=t-h+1}^{t} C_{t'}(\ .\ ) \leq 0 \qquad \forall\, t = h, 2h, \ldots, H \tag{8.1}$$

To take account of the position in time within each ATP when task start and finish a non-negative time-dependant weighting function $f(t-t')$ can be incorporated into equation [8.1] as follows.

$$\sum_{t'=t-h+1}^{t} f(t-t')C_{t'}(\ .\ ) \leq 0 \qquad \forall\, t = h, 2h, \ldots, H \tag{8.2}$$

As the weighting function f is a function of $t - t'$ where $t' = t - h + 1, t - h + 2, ..., t$ the variation over each ATP is identical for every ATP.

Weighting Functions

Consider the following family of weighting functions, which has a non-negative value over the specified range.

$$f^p(t - t') \equiv (t - t + 1)^p \qquad \forall\, p = 0, 1, ..., m \qquad [8.3]$$

This function's variation over each ATP is plotted in Figure 14 for $p = 0,1$ and 2.

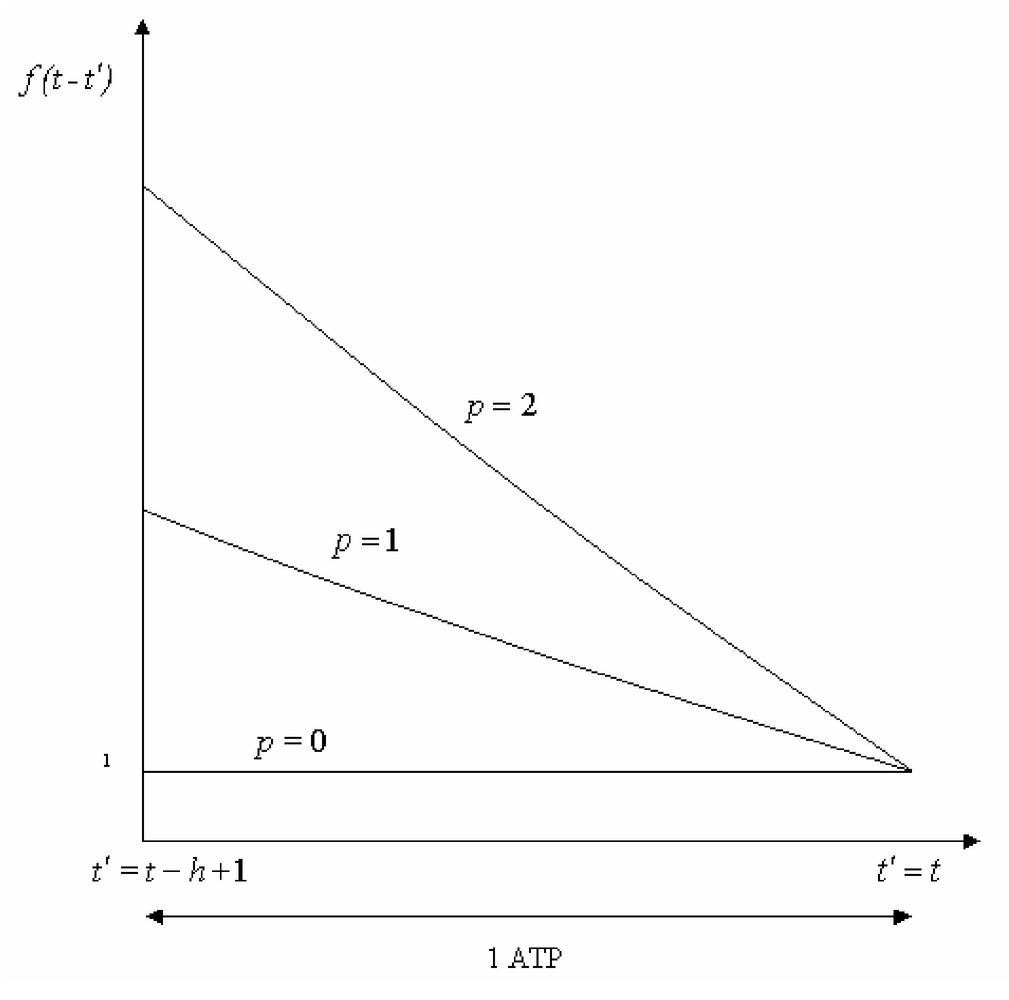


Figure 14, Aggregation weighting functions

Use of the weighting function given by [8.3] leads to a family of aggregated constraints of the form:

$$\sum_{t'=t-h+1}^{t} f^{p}(t-t')C_{t'}(\ .\ ) \leq 0 \qquad \forall\, p = 0,1,\ldots,m\ , t = h,2h,\ldots,H \qquad [8.4]$$

Aggregate Variables

In an RTN formulation the constraints are linear therefore we can replace all the detailed variables with aggregated variables except for those that correspond to tasks that exist in more than one ATP, these tasks give rise to the linking variables introduced on page 3. For a p^{th} order aggregate formulation, we will need the aggregate variables $\widetilde{N}_{kt}^{(i)}\ , \widetilde{\xi}_{kt}^{(i)}\ , \widetilde{\mu}_{kr\theta}^{(p)}\ , \widetilde{\nu}_{kr\theta}^{(p)}\ , i = 0,\ldots,p$ and $\widetilde{R}_{rt}^{(i)}\ , i = 0,\ldots,p-1$. In general, The following variables are required for an m^{th} order aggregate formulation.

1. Linking variables

   i. Excess resource variables:

$$R_{rt} \qquad \forall\, r\ , t = h,2h,\ldots,H$$

   ii. Task extent variables:

$$N_{k,t+1-\theta}\ , \xi_{k,t+1-\theta} \qquad \forall\, k\ , t = h,2h,\ldots,H\ , \theta = 1,\ldots,\tau_k$$

2. Aggregate variables

   i. Excess resource variables:

$$\widetilde{R}_{rt}^{(p)} \qquad \forall\, r\ , t = h,2h,\ldots,H\ , p = 0,\ldots,m-1$$

defined by:

$$\widetilde{R}_{rt}^{(p)} \equiv \sum_{t'=t-h+1}^{t} (t-t'+1)^p R_{rt'} \qquad [8.5]$$

ii. Task extent variables

$$\widetilde{N}_{kt}^{(p)}, \widetilde{\xi}_{kt}^{(p)} \qquad \forall k, t = h,2h,...,H, p = 0,...,m$$

defined by:

$$\widetilde{N}_{kt}^{(p)} \equiv \sum_{t'=t\text{-}h+1}^{t} (t-t'+1)^p N_{kt'} \qquad [8.6]$$

$$\widetilde{\xi}_{kt}^{(p)} \equiv \sum_{t'=t\text{-}h+1}^{t} (t-t'+1)^p \xi_{kt'} \qquad [8.7]$$

iii. Parameters

$$\widetilde{\mu}_{kr\theta}^{(p)}, \widetilde{\nu}_{kr\theta}^{(p)} \qquad \forall k, r, \theta = 1,...,\tau_k, p = 0,...,m$$

$$\widetilde{\Pi}_{rt}^{(p)} \qquad \forall r, t = h,2h,...,H, p = 0,...,m$$

defined by:

$$\widetilde{\mu}_{kr\theta}^{(p)} \equiv \sum_{\theta'=\theta}^{\tau_k} (\theta')^p \mu_{kr\theta'} \qquad [8.8]$$

$$\widetilde{\nu}_{kr\theta}^{(p)} \equiv \sum_{\theta'=\theta}^{\tau_k} (\theta')^p \nu_{kr\theta'} \qquad [8.9]$$

$$\widetilde{\Pi}_{rt}^{(p)} \equiv \sum_{t'=t-h+1}^{t} (t-t'+1)^p \Pi_{rt'} \qquad [8.10]$$

First Order Aggregated RTN Formulation

The Objective Function

The aggregate form of the objective function is independent of the overall order of aggregation for the RTN formulation. All terms involving a summation of variables over time involve zero order aggregate variables only, to give the following:

$$max \sum_{r \in MR} \left(C_r^F (R_{rH} - R_{r0}) - C_r^I \sum_{t=1}^{H} R_{rt} \right) - \sum_k \sum_{t=1}^{H} (C_k^F N_{kt} + C_k^v \xi_{kt}) - \sum_{t=1}^{H} C_{rt}^E \Pi_{rt} \qquad [8.11]$$

Model Constraints

1. Excess Resource Balances

The first order resource balance is written as follows:

$$\begin{aligned} \widetilde{R}_{rt}^{(0)} - hR_{r,t-h} = & \sum_k \widetilde{\mu}_{kr0}^{(0)} \widetilde{N}_{kt}^{(1)} - \sum_k \widetilde{\mu}_{kr1}^{(1)} \widetilde{N}_{kt}^{(0)} + \sum_k \widetilde{\nu}_{kr1}^{(1)} \widetilde{\xi}_{kt}^{(0)} + \\ & \sum_k \sum_{\theta'=1}^{\tau_k} N_{k,t-h+1-\theta'} \sum_{\theta=\theta'}^{\tau_k} (h+\theta'-\vartheta) \mu_{kr\theta} - \sum_k \sum_{\theta'=1}^{\tau_k} N_{k,t-h+1-\theta'} \sum_{\theta=\theta'}^{\tau_k} (\theta'-\vartheta) \mu_{kr\theta} + \\ & \sum_k \sum_{\theta'=1}^{\tau_k} \xi_{k,t-h+1-\theta'} \sum_{\theta=\theta'}^{\tau_k} (h+\theta'-\vartheta) \nu_{kr\theta} - \sum_k \sum_{\theta'=1}^{\tau_k} \xi_{k,t-h+1-\theta'} \sum_{\theta=\theta'}^{\tau_k} (\theta'-\vartheta) \nu_{kr\theta} + \widetilde{\Pi}_{rt}^{(1)} \end{aligned} \qquad [8.12]$$

where:

$$\widetilde{N}_{kt}^{(1)} \equiv \sum_{t'=t-h+1}^{t} (t-t'+1) N_{kt'} \qquad [8.13]$$

$$\tilde{\xi}_{kt}^{(1)} \equiv \sum_{t'=t-h+1}^{t} (t-t'+1)\xi_{kt'} \qquad [8.14]$$

$$\tilde{R}_{rt}^{(0)} \equiv \sum_{t'=t-h+1}^{t} R_{rt'} \qquad [8.15]$$

The first order resource balance is constrained by the following aggregated inequality:

$$0 \le \tilde{R}_{rt}^{(1)} - 2\tilde{R}_{rt}^{(0)} + R_{rt} \le \tfrac{1}{2}(h-1)(h-2)R_r^{max} \qquad \forall\, r\,,\, t = h,2h,\ldots,H \qquad [8.16]$$

2. Capacity and Batch-Size Constraints

The first order operational constraints are written as follows:

$$\begin{aligned} &V_{kr}^{min}\left(\tilde{N}_{kt}^{(1)} - (\tau_k + 1)\tilde{N}_{kt}^{(0)} + \sum_{\theta=1}^{\tau_k} (\tau_k + 1 - \theta) N_{k,t+1-\theta} \right) \le \\ &\tilde{\xi}_{kt}^{(1)} - (\tau_k + 1)\tilde{\xi}_{kt}^{(0)} + \sum_{\theta=1}^{\tau_k} (\tau_k + 1 - \theta)\xi_{k,t+1-\theta} \\ &\le V_{kr}^{max}\left(\tilde{N}_{kt}^{(1)} - (\tau_k + 1)\tilde{N}_{kt}^{(0)} + \sum_{\theta=1}^{\tau_k} (\tau_k + 1 - \theta) N_{k,t+1-\theta} \right) \qquad [8.17] \\ &\forall\, k\,,\, t = h,2h,\ldots,H\,,\, r \in PE_k \end{aligned}$$

and

$$\begin{aligned} &V_{kr}^{min}\left(h\tilde{N}_{kt}^{(0)} - \tilde{N}_{kt}^{(1)} - \sum_{\theta=1}^{\tau_k} (h-\theta) N_{k,t+1-\theta} \right) \le h\tilde{\xi}_{kt}^{(0)} - \tilde{\xi}_{kt}^{(1)} - \sum_{\theta=1}^{\tau_k} (h-\theta)\xi_{k,t+1-\theta} \\ &\le V_{kr}^{max}\left(h\tilde{N}_{kt}^{(0)} - \tilde{N}_{kt}^{(1)} - \sum_{\theta=1}^{\tau_k} (h-\theta) N_{k,t+1-\theta} \right) \qquad \forall\, k\,,\, t = h,2h,\ldots,H\,,\, r \in PE_k \end{aligned} \qquad [8.18]$$

9. Assessment of Aggregate Solution Method

Method

First, an aggregate schedule model was obtained for a schedule case very similar to the Hydrolubes Plant example from Panos (2001). This schedule model was altered so that it accurately represented the Hydrolubes Plant example. The schedule model was then turned into a design model by changing the initial excess resource constants, R_{r0}, into variables. The equipment initial excess resource variables, $R_{r0}\ \forall r \in PE$, were constrained to integers in the range 0 to 100. The rest of the initial excess resource variables, $R_{r0}\ \forall r \notin PE$, had upper and lower bounds that both took the same value as the constants, $R_{r0}\ \forall r \notin PE$, had for the schedule case therefore restricting the variables to a region that made them behave as constants with the same values as they had in the schedule model. To enable the optimum design to be achieved an objective function was formulated to maximize the product at the end of the time horizon while minimizing the equipment costs, as follows:

$$Max \sum_{r \in MR} R_{rH} - \sum_{r \in PE} R_{r0} C_r^{CAP} \qquad [9.0]$$

The values obtained for initial excess resource variables, $R_{r0}\ \forall r \in PE$, where used to run a schedule model in gBSS. The input file for the GAMS aggregate design model is in Appendix C on page 3.

The time to solve an aggregated design in GAMS plus the time to solve a schedule in gBSS was considered the total time to solve an aggregated design model. This was compared with the time to solve a detailed design model in gBSS. A design model was used to compare the aggregate solution time with the detail model solution time, because not only did design models take longer to run but also this approach allowed both methods to produce an identical solution output. The gBSS design method produced the normal gBSS solution output while the aggregate design method also produced the normal gBSS solution output by running a gBSS schedule once the aggregate design in GAMS had been finished. This technique allowed solution time comparisons between two methods that produced identical solution outputs.

To test the two design model solution methods I used two versions of the Hydrolubes Plant example one which was run for three different time horizons. The first version had delivery demands in every shift throughout the time horizon forcing the plant to make all products continuously throughout the schedule. The second model, which was run for three different time horizons, had the same total amounts delivered as the first model except the deliveries were are in the last time interval.

The solution times were considered the time until the first feasible solution was produced with an objective function within 5% of the solution objective function for the LP produced when the MILP integers are relaxed to continuous variables. The plant schedules produced by the two solution methods were identical for both versions of the Hydrolubes Plant example.

Results

Table 5, Comparison between Aggregate and Un-Aggregate (detailed) Solution Times

Model		Solution Time in seconds			
No.	Time Horizon, hr	gBSS un-aggregated design	GAMS aggregated design	gBSS schedule	Total aggregated design
1	32	35	3.6	2	5.6
2	32	26	0.24	4	4.24
	64	112	0.5	4	4.5
	96	163	0.7	7	7.7
Average		84	1.26 +	4.25 =	5.51

Both solution methods provided a feasible similar plant schedule, but due to computational limitations, not every sub LP with in the MILPs was investigated, as this would have taken days to complete. The available computers licensed to run the software required were not able to be used for such long periods of time. All the integer solutions obtained from the MILPs were within 5% of the non-integer solution, which represent the best possible value an integer solution could have.

Conclusion

The aggregated design method obtains a solution 15 times faster then the original detailed method with no loss in scheduling detail. Both solution methods produced identical, or very close to identical solutions, the reason some solutions were not exactly identical is, at least, partly because the solution procedure was not run long enough to check every feasible integer combination, which would have ensured the absolute optimum solution being found. The comparison completed shows overwhelmingly that aggregation can significantly reduce the solution time for process plant designs.

10. Conclusions and Further Work

This report explained computer-aided scheduling and design with particular reference to general Batch Scheduling System (gBSS) and its limitations. Techniques were then presented to solve the limitations of gBSS.

The process modelling techniques presented in this report such as the Resource-Task Network (RTN) formulation allows detailed consideration of a design or schedule problem taking into account the tradeoffs between revenue, and running or capital cost. This enables optimal solutions for process design or scheduling to be obtained.

The optimal solution of systems is an area that will expand at an even increasing rate in the future, as computers become more powerful and optimisation techniques improve. This report introduces an important technique to significantly reduce the solution time for time-discretised MILP models. These models are not only of benefit to the process industry but also many other areas in society from financial investors to train timetable designers.

Graphical User Interfaces (GUIs) are also discussed. It is noted that diagrams, tables, and graphs can significantly increase a new user's ability to understand software.

Potential Directions for Further Work

There are several areas for further work as follows:

1. The development of an MILP that incorporates aggregation should be the next goal in developing temporal aggregation techniques. The higher the order aggregation is more detailed but also more likely to become infeasible to solve as each increased order of aggregation introduces new constraints restricting the solution space. Therefore, any software implementing aggregation would need a mechanism to cope with infeasible aggregates solutions. This would normally then require the software to lower the aggregation order and try again.

2. Another approach to reducing solution times would be to allow several computers on a network to simultaneously solve the same problem. For example, the branch-and-bound algorithm controlled on a central computer could send out small groups of the sub LPs from the main MILP to other computers on a network to solve independently. This approach would allow the joining of main computer to share their CPUs for large model solutions.

3. There is no method to take account of uncertainty. Several resources on a plant might be subject to fluctuation therefore it would be beneficial to be able to generate a schedule that was truly robust in certain areas. A factor could be introduced that took account of how reliable the model data was, several schedules could be produced to account for any possible plant scenario.

4. The GUI developed in this report was restricted to once process recipe only, although the task input fraction and process time could be altered this was very restricting. A fully flexible GUI based on a RTN or STN should be developed. This could be a general GUI for scheduling or design in gBSS that allowed the entry of a process recipe by use of the State-Task Network or the Resource-Task Network.

5. The models in this report were based on long time scales. These models could be incorporated with a dynamic short time scale models to enable total optimal control of process plant. The short time scale scheduler could be linked to the sensors on a plant so that it responded in real time with the best optimal scheduling solutions. Information from short time scheduler could be collected from several plants allowing optimal solution for multisite and long term scheduling.

11. References

Kondili, E., C.C. Pantelides and R.W.H. Sargent, "A General Algorithm for Short-Term Scheduling of Batch Operations. Part I – Mathematical Formulation", *Comput. Chem.. Engng*., 17, 211-227 (1993)

Pantelides, C.C., "Unified Framework for Optimal Process Planning and Scheduling", *Proc. Second Conf. on Foundations of Computer-Aided Operations*, Rippin D.W.T and J.Hale eds., CACHE Publications, 235-274 (1994)

Barbosa-Póvoa, A.P.F.D.; Pantelides, C.C., "Design of Multipurpose Production Facilities; A RTN Decomposition-Based Algorithm", *European Symposium on Computer Aided Process Engineering-9,* G.V.Reklaitis eds,PERGAMON Publications, S7-S10 Vol. 23 (1999)

Wilkinson S.J., "Aggregate Formulations for Large-Scale Process Scheduling Problems", Ph.D Thesis, Imperial College of Science, Technology and Medicine, London, 50-73 (1996)

Shah N., "Flexible Plant Scheduling Option - Course Notes" Chemical Engineering and Chemical Technology Department Imperial College of Science, Technology and Medicine, London 14-28 (2001)

?? Panos, "GAMS Model"

12. Appendix A, gBSS Input Files for Hydrolubes Plant

Problem File; Hydro.prb

```
TITLE Hydrolubes
*
PROBLEM_TYPE SHORT_TERM_DESIGN
*
RECIPE_FILE Hydro
RESOURCE_FILE Hydro
*
TIME_MODE RELATIVE
HORIZON TO 32
INTERVAL 1
*
* ------------------
* METHOD DESCRIPTION
* ------------------
*
METHOD
      MODE FOREIGN_MIP
      SOLVER XPRESS
*
* -----------------
* STATE DESCRIPTION
* -----------------
*
STATE FeedA
      RECEIVE 200 AT 0 VALUE 10
STATE FeedB
      RECEIVE 200 AT 0 VALUE 10
STATE FeedC
      RECEIVE 200 AT 0 VALUE 10
STATE FeedD
      RECEIVE 200 AT 0 VALUE 10
STATE Add1
      RECEIVE 20 AT 0 VALUE 5
STATE Add2
      RECEIVE 20 AT 0 VALUE 5
STATE Prod1
      DELIVER 20 AT 10 VALUE 20
      DELIVER 25 AT 16 VALUE 20
      DELIVER 25 AT 22 VALUE 20
      DELIVER 30 AT 32 VALUE 20
STATE Prod2
      DELIVER 25 AT 10 VALUE 20
      DELIVER 30 AT 16 VALUE 20
      DELIVER 30 AT 22 VALUE 20
      DELIVER 35 AT 32 VALUE 20
STATE Prod3
      DELIVER 30 AT 10 VALUE 20
      DELIVER 35 AT 16 VALUE 20
      DELIVER 35 AT 22 VALUE 20
```

```
      DELIVER 40 AT 32 VALUE 20
STATE BlenA
      DELIVER 15 AT 22 VALUE 15
      DELIVER 25 AT 32 VALUE 15
```

Recipe Files; Hydro.stn

```
* -----------------------
* TASK TYPES DECLARATIONS
* -----------------------
*
TASK_TYPE TTReaction TTBlending TTMixing
*
* ------------------
* STATE DECLARATIONS
* ------------------
*
STATE FeedA
STATE FeedB
STATE FeedC
STATE FeedD
STATE ReacP
STATE Add1
STATE Add2
STATE Int1
STATE Prod1
STATE Prod2
STATE Prod3
STATE BlenA
*
* -----------------
* TASK DECLARATIONS
* -----------------
*
TASK TReaction
      INSTATE FeedA FRACTION 1.0
      OUTSTATE ReacP FRACTION 1.0 PROC_TIME 5.0
      TYPE TTReaction
      USE Operators
        FROM 0.0 To 1.0 FIXED 1.0
TASK TBlendingA
      INSTATE ReacP FRACTION 0.999
      INSTATE Add1 FRACTION 0.001
      OUTSTATE BlenA FRACTION 1.0 PROC_TIME 2.0
      TYPE TTBlending
      USE Operators
        FROM 0.0 To 1.0 FIXED 0.5
TASK TBlendingB
      INSTATE ReacP FRACTION 0.999
      INSTATE Add2 FRACTION 0.001
      OUTSTATE Int1 FRACTION 1.0 PROC_TIME 2.0
      TYPE TTBlending
      USE Operators
        FROM 0.0 To 1.0  FIXED 0.5
```

```
TASK TMixing1
      INSTATE Int1 FRACTION 0.38
      INSTATE FeedB FRACTION 0.62
      OUTSTATE Prod1 FRACTION 1.0 PROC_TIME 2.0
      TYPE TTMixing
      USE Operators
        FROM 0.0 To 1.0 FIXED 0.5
TASK TMixing2
      INSTATE Int1 FRACTION 0.4
      INSTATE FeedD FRACTION 0.6
      OUTSTATE Prod2 FRACTION 1.0 PROC_TIME 2.0
      TYPE TTMixing
      USE Operators
        FROM 0.0 To 1.0 FIXED 0.5
TASK TMixing3
      INSTATE Int1 FRACTION 0.53
      INSTATE FeedC FRACTION 0.47
      OUTSTATE Prod3 FRACTION 1.0 PROC_TIME 2.0
      TYPE TTMixing
      USE Operators
        FROM 0.0 To 1.0 FIXED 0.5
```

Resource File; Hydro.uni

```
* ----------------------
* UNIT TYPE DECLARATIONS
* ----------------------
*
UNIT_TYPE UTSTORAGEP
      CAPACITY 10:1000
UNIT_TYPE UTSTORAGEF
      CAPACITY 10:1000
UNIT_TYPE UTREACTOR
      CAPACITY 10:1000
      FIXED_COST 0.0
      VARIABLE_COST 10.0
UNIT_TYPE UTBLENDER
      CAPACITY 10:1000
      FIXED_COST 0.0
      VARIABLE_COST 10.0
UNIT_TYPE UTMIXER
      CAPACITY 10:1000
      FIXED_COST 0.0
      VARIABLE_COST 10.0
*
* -----------------
* UNIT DECLARATIONS
* -----------------
*
UNIT TANKFA
      STORE FEEDA
      TYPE UTSTORAGEF
UNIT TANKFB
      STORE FEEDB
```

```
      TYPE UTSTORAGEF
UNIT TANKFC
      STORE FEEDC
      TYPE UTSTORAGEF
UNIT TANKFD
      STORE FEEDD
      TYPE UTSTORAGEF
UNIT TANKA1
      STORE ADD1
      TYPE UTSTORAGEF
UNIT TANKA2
      STORE ADD2
      TYPE UTSTORAGEF
UNIT TANKINT1
      STORE INT1
      TYPE UTSTORAGEF
UNIT TANKP1
      STORE PROD1
      TYPE UTSTORAGEP
UNIT TANKP2
      STORE PROD2
      TYPE UTSTORAGEP
UNIT TANKP3
      STORE PROD3
      TYPE UTSTORAGEP
UNIT TANKBA
      STORE BLENA
      TYPE UTSTORAGEP
UNIT UREACTOR (5)
      PERFORM TTREACTION
      TYPE UTREACTOR
UNIT UBLENDER (5)
      PERFORM TTBLENDING
      TYPE UTBLENDER
UNIT UMIXING (5)
      PERFORM TTMIXING
      TYPE UTMIXER
*
* ----------------------
* UTILITIES DECLARATIONS
* ----------------------
*
UTILITY OPERATORS
      AVAILABLE 10
```

13. Appendix B, gBSS Output Files for Hydrolubes Plant

Gantt Chart

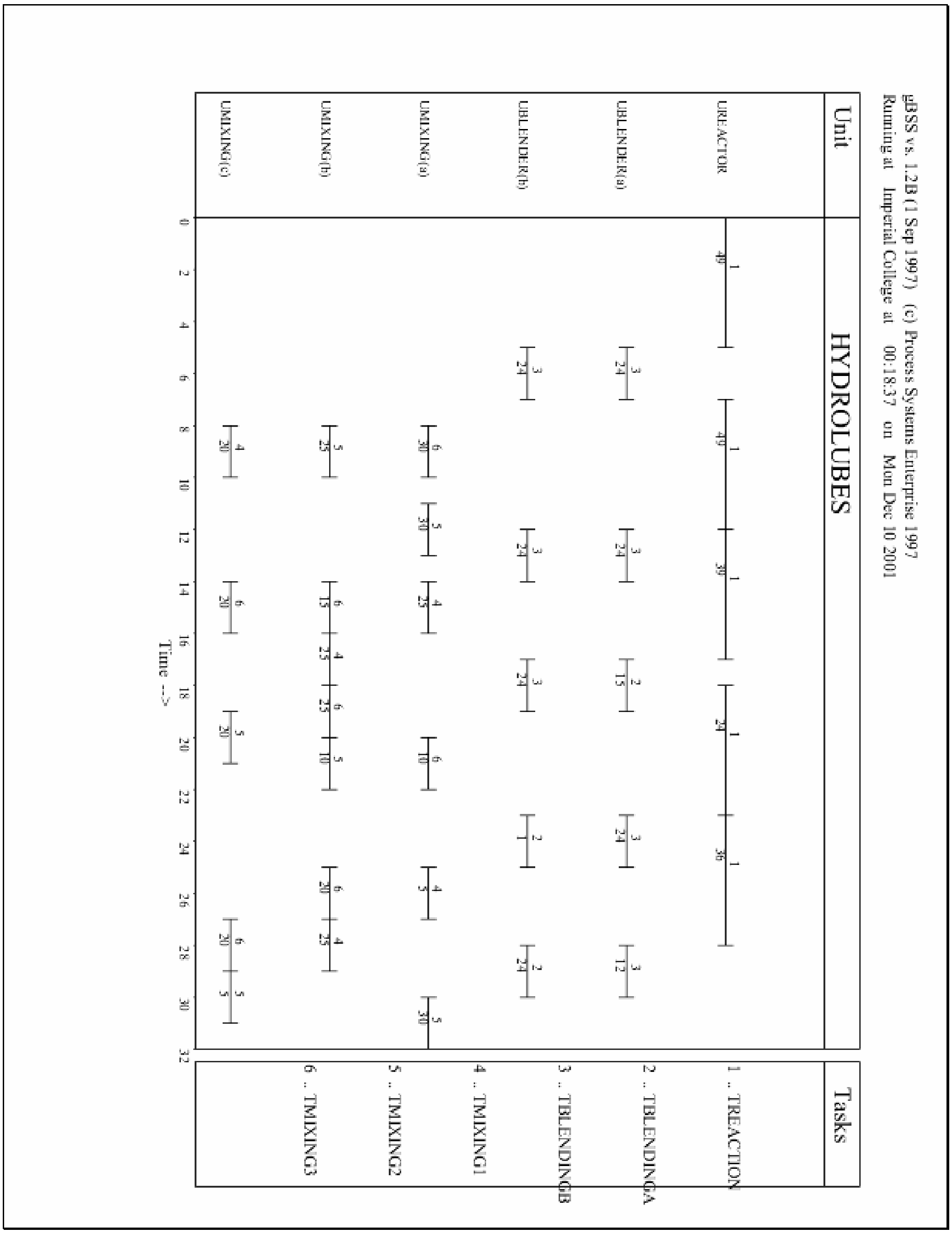


Figure 15, Gantt Chart Page One

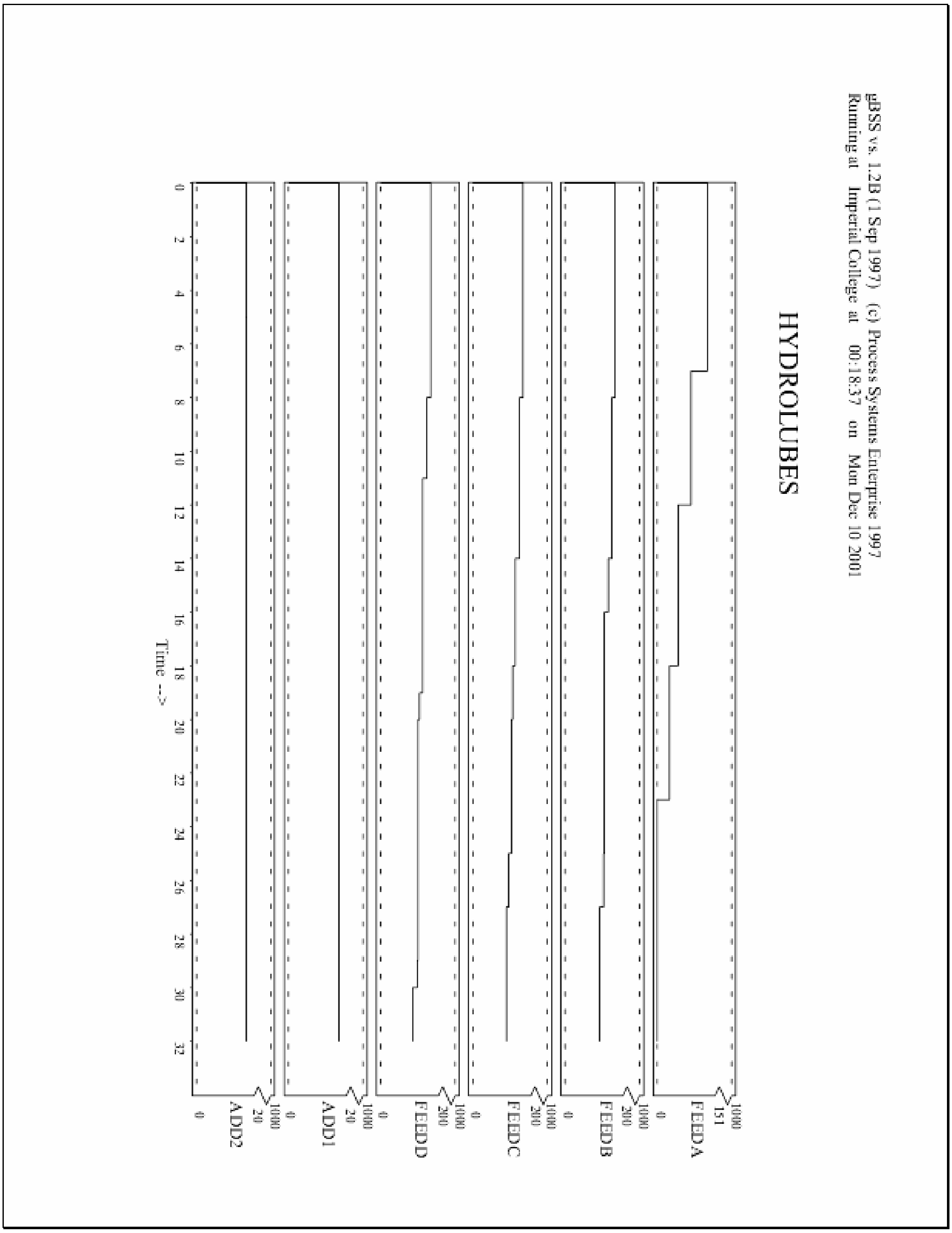


Figure 16, Gantt Chart Page Two

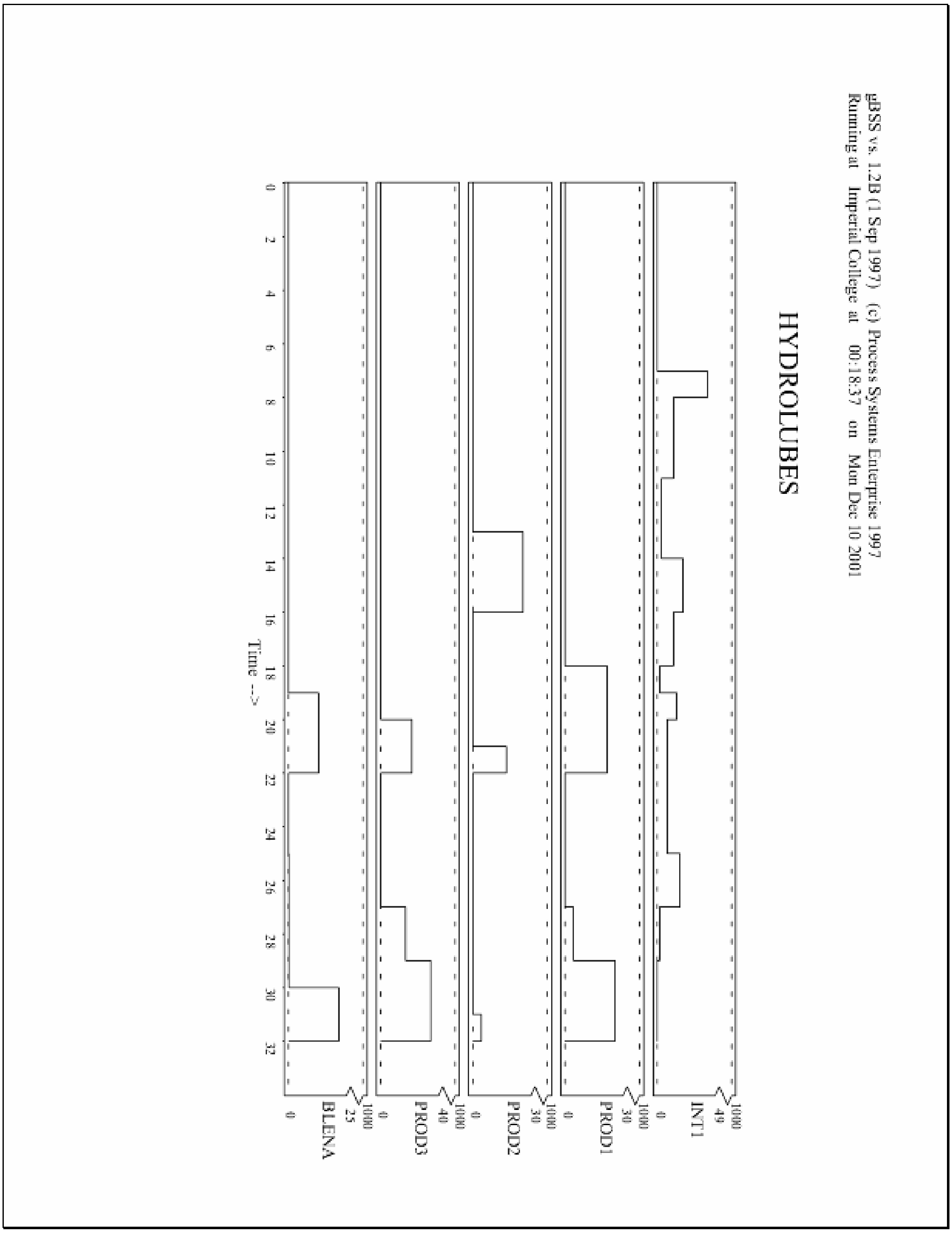


Figure 17, Gantt Chart Page Three

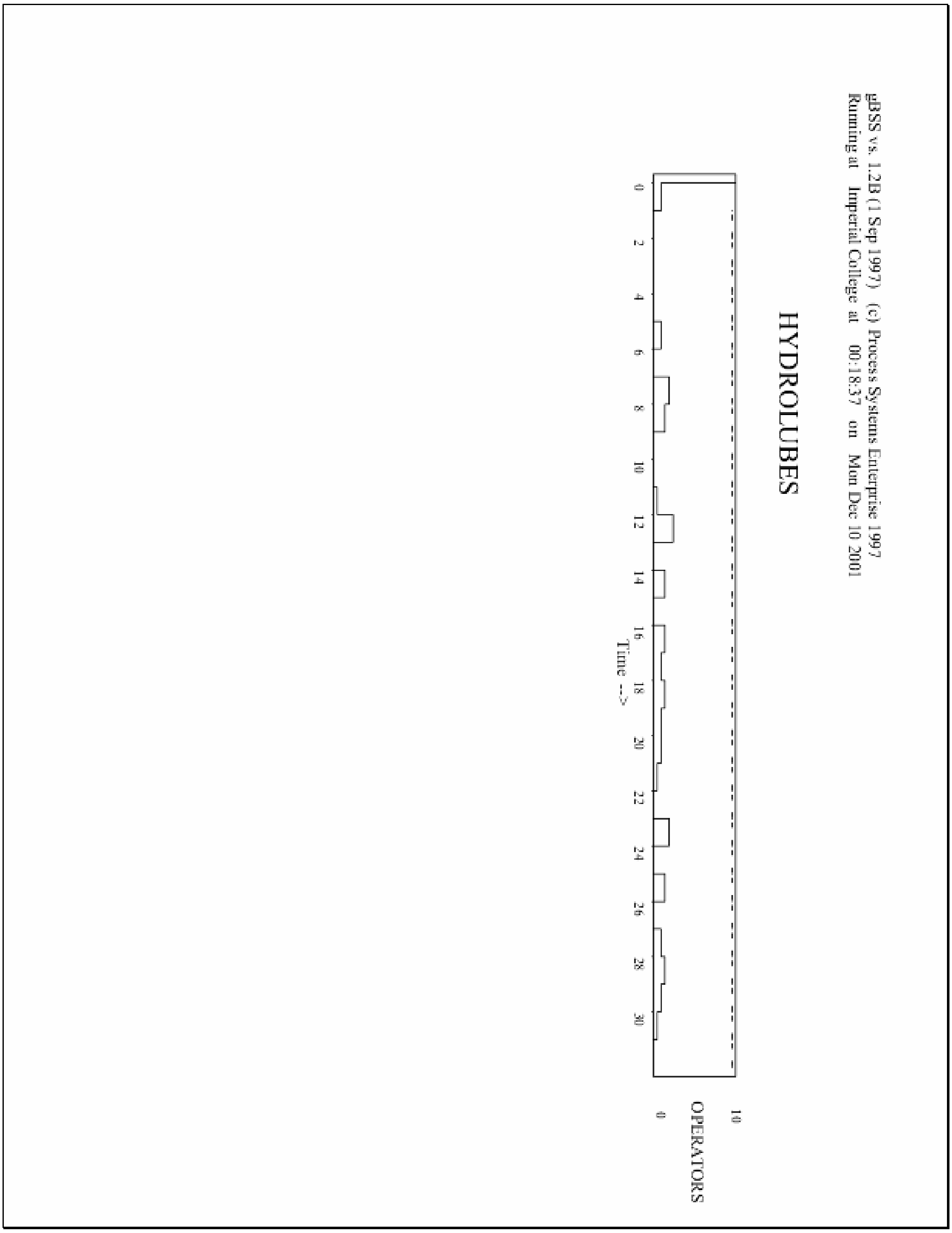


Figure 18, Gantt Chart Page Four

Hydrolubes gBSS Output File; Hydro.out

```
                 ######    #####    #####
                 #    #    #   #    #   #
                 #    #    #        #
          ####   ######    #####    #####
         #   #   #    #        #        #
         #   #   #    #    #   #    #   #
          ####   ######    #####    #####
             #
            #
           #

           general Batch Scheduling System

         Multipurpose Batch Optimisation Software

            Copyright Imperial College (1996)

              Running at Imperial College

                 Mon Dec 10 2001    00:16:29

1 Page:   1

*********************************************************************************************************************************
*      gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)       Copyright Imperial College 1996       *
*      Running at Imperial College, London at 00:16:29      on Mon Dec 10 2001                                    *
*********************************************************************************************************************************

                    ===================================
                    ***** PROBLEM FILE TRANSLATION *****
                    ===================================
 Processing Problem File  :  hydro
  1  TITLE  HYDROLUBES
  2
  3  *
  4
  5  PROBLEM_TYPE SHORT_TERM_DESIGN
  6
  7  *
  8
  9  RECIPE_FILE HYDRO
 10
 11  RESOURCE_FILE HYDRO
 12
 13  *
 14
 15  TIME_MODE RELATIVE
 16
 17  HORIZON TO 32
 18
 19  INTERVAL 1
 20
 21  *
 22
 23  * ------------------
 24
 25  * METHOD DESCRIPTION
 26
 27  * ------------------
 28
 29  *
 30
 31  METHOD
 32
 33            MODE FOREIGN_MIP
 34
 35            SOLVER XPRESS
 36
 37  *
 38
 39  * -----------------
 40
 41  * STATE DESCRIPTION
 42
 43  * -----------------
 44
 45  *
 46
 47  STATE FEEDA
 48
 49            RECEIVE 200 AT 0 VALUE 10
 50
 51  STATE FEEDB
 52
 53            RECEIVE 200 AT 0 VALUE 10
 54

1 Page:   2

*********************************************************************************************************************************
*      gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)       Copyright Imperial College 1996       *
*      Running at Imperial College, London at 00:16:29      on Mon Dec 10 2001                                    *
*********************************************************************************************************************************

 55  STATE FEEDC
 56
 57            RECEIVE 200 AT 0 VALUE 10
 58
 59  STATE FEEDD
 60
 61            RECEIVE 200 AT 0 VALUE 10
 62
 63  STATE ADD1
 64
```

```
              RECEIVE 20 AT 0 VALUE 5
 STATE ADD2
              RECEIVE 20 AT 0 VALUE 5
 STATE PROD1
              DELIVER 20 AT 10 VALUE 20
              DELIVER 25 AT 16 VALUE 20
              DELIVER 25 AT 22 VALUE 20
              DELIVER 30 AT 32 VALUE 20
 STATE PROD2
              DELIVER 25 AT 10 VALUE 20
              DELIVER 30 AT 16 VALUE 20
              DELIVER 30 AT 22 VALUE 20
              DELIVER 35 AT 32 VALUE 20
 STATE PROD3
              DELIVER 30 AT 10 VALUE 20
              DELIVER 35 AT 16 VALUE 20
              DELIVER 35 AT 22 VALUE 20
              DELIVER 40 AT 32 VALUE 20
 STATE BLENA
              DELIVER 15 AT 22 VALUE 15
              DELIVER 25 AT 32 VALUE 15
1 Page:   3

****************************************************************************************************************************
*       gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)        Copyright Imperial College 1996       *
*       Running at Imperial College, London at 00:16:29      on Mon Dec 10 2001                                    *
****************************************************************************************************************************

                    =====================================
                    ***** RECIPE FILE TRANSLATION *****
                    =====================================

 Processing Recipe File  :  hydro
 * ----------------------
 * TASK TYPES DECLARATIONS
 * ----------------------
 *
 TASK_TYPE TTREACTION TTBLENDING TTMIXING
 *
 * ------------------
 * STATE DECLARATIONS
 * ------------------
 *
 STATE FEEDA
 STATE FEEDB
 STATE FEEDC
 STATE FEEDD
 STATE REACP
 STATE ADD1
 STATE ADD2
 STATE INT1
 STATE PROD1
 STATE PROD2
 STATE PROD3
 STATE BLENA
 *
 * -----------------
 * TASK DECLARATIONS
 * -----------------
 *

1 Page:   4

****************************************************************************************************************************
*       gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)        Copyright Imperial College 1996       *
*       Running at Imperial College, London at 00:16:29      on Mon Dec 10 2001                                    *
```

```
*******************************************************************************************************************************

TASK TREACTION
              INSTATE FEEDA FRACTION 1.0
              OUTSTATE REACP FRACTION 1.0 PROC_TIME 5.0
              TYPE TTREACTION
              USE OPERATORS
                FROM 0.0 TO 1.0 FIXED 1.0
TASK TBLENDINGA
              INSTATE REACP FRACTION 0.999
              INSTATE ADD1 FRACTION 0.001
              OUTSTATE BLENA FRACTION 1.0 PROC_TIME 2.0
              TYPE TTBLENDING
              USE OPERATORS
                FROM 0.0 TO 1.0 FIXED 0.5
TASK TBLENDINGB
              INSTATE REACP FRACTION 0.999
              INSTATE ADD2 FRACTION 0.001
              OUTSTATE INT1 FRACTION 1.0 PROC_TIME 2.0
              TYPE TTBLENDING
              USE OPERATORS
                FROM 0.0 TO 1.0  FIXED 0.5
TASK TMIXING1
              INSTATE INT1 FRACTION 0.38
              INSTATE FEEDB FRACTION 0.62
              OUTSTATE PROD1 FRACTION 1.0 PROC_TIME 2.0
              TYPE TTMIXING
              USE OPERATORS
                FROM 0.0 TO 1.0 FIXED 0.5
TASK TMIXING2
              INSTATE INT1 FRACTION 0.4

1 Page:   5

*******************************************************************************************************************************
*         gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996          *
*         Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                         *
*******************************************************************************************************************************

              INSTATE FEEDD FRACTION 0.6
              OUTSTATE PROD2 FRACTION 1.0 PROC_TIME 2.0
              TYPE TTMIXING
              USE OPERATORS
                FROM 0.0 TO 1.0 FIXED 0.5
TASK TMIXING3
              INSTATE INT1 FRACTION 0.53
              INSTATE FEEDC FRACTION 0.47
              OUTSTATE PROD3 FRACTION 1.0 PROC_TIME 2.0
              TYPE TTMIXING
              USE OPERATORS
                FROM 0.0 TO 1.0 FIXED 0.5
1 Page:   6

*******************************************************************************************************************************
*         gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996          *
*         Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                         *
*******************************************************************************************************************************

                         ======================================
                         ***** RESOURCE FILE TRANSLATION *****
                         ======================================

Processing Resource File  :  hydro
* ---------------------
* UNIT TYPE DECLARATIONS
* ---------------------
*
```

```
UNIT_TYPE UTSTORAGEP
        CAPACITY 10:1000
UNIT_TYPE UTSTORAGEF
        CAPACITY 10:1000
UNIT_TYPE UTREACTOR
        CAPACITY 10:1000
        FIXED_COST 0.0
        VARIABLE_COST 10.0
UNIT_TYPE UTBLENDER
        CAPACITY 10:1000
        FIXED_COST 0.0
        VARIABLE_COST 10.0
UNIT_TYPE UTMIXER
        CAPACITY 10:1000
        FIXED_COST 0.0
        VARIABLE_COST 10.0
*
* -----------------
* UNIT DECLARATIONS
* -----------------
*
UNIT TANKFA
        STORE FEEDA

1 Page:   7

*********************************************************************************************************************************************
*       gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996          *
*       Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                        *
*********************************************************************************************************************************************

        TYPE UTSTORAGEF
UNIT TANKFB
        STORE FEEDB
        TYPE UTSTORAGEF
UNIT TANKFC
        STORE FEEDC
        TYPE UTSTORAGEF
UNIT TANKFD
        STORE FEEDD
        TYPE UTSTORAGEF
UNIT TANKA1
        STORE ADD1
        TYPE UTSTORAGEF
UNIT TANKA2
        STORE ADD2
        TYPE UTSTORAGEF
UNIT TANKINT1
        STORE INT1
        TYPE UTSTORAGEF
UNIT TANKP1
        STORE PROD1
        TYPE UTSTORAGEP
UNIT TANKP2
        STORE PROD2
        TYPE UTSTORAGEP
UNIT TANKP3
        STORE PROD3
        TYPE UTSTORAGEP
UNIT TANKBA
```

```
1 Page:  8

*****************************************************************************************************************************
*        gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996         *
*        Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                       *
*****************************************************************************************************************************

              STORE BLENA
              TYPE UTSTORAGEP
UNIT UREACTOR
              PERFORM TTREACTION
              TYPE UTREACTOR
UNIT UBLENDER (2)
              PERFORM TTBLENDING
              TYPE UTBLENDER
UNIT UMIXING (3)
              PERFORM TTMIXING
              TYPE UTMIXER
*
* ----------------------
* UTILITIES DECLARATIONS
* ----------------------
*
UTILITY OPERATORS
              AVAILABLE 10
No translation errors were detected - execution continues.

1 Page:  9

*****************************************************************************************************************************
*        gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996         *
*        Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                       *
*****************************************************************************************************************************

                       ==================================
                       ***** DATA CONSISTENCY CHECKS *****
                       ==================================

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Checking Problem File Consistency ...  OK
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Checking STN File Consistency ...  OK
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Checking Unit File Consistency ...  OK
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Checking Cleaning Consistency ...  OK
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Checking Utility File Consistency ...  OK
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Checking HX Consistency ...  OK
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Data found to be consistent - execution continues.

                  =======================================================
                  ***** DATA POST PROCESSING AND VERIFICATION CHECKS *****
                  =======================================================

Data post processing and verification successful.

1 Page:  10

*****************************************************************************************************************************
*        gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996         *
*        Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                       *
*****************************************************************************************************************************

======================
***** TASK TYPES *****
======================

   Type  #   1 called TTREACTION
   Type  #   2 called TTBLENDING
   Type  #   3 called TTMIXING

1 Page:  11

*****************************************************************************************************************************
*        gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996         *
*        Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                       *
```

```
*********************************************************************************************************************

                        ====================
                        ***** T A S K S *****
                        ====================

Problem involves 6 processing tasks.
--------------------------------------------------------------------------------------------------------
        I n p u t/ O u t p u t
          S t a t e s                                   U t i l i t i e s
      ------------------------------ Proc   Type      Unit     ------------------------------------------
  TaskName      Name      I/O  Frac Proc Time                    Name       From To  Fixed Variable
                               Time
--------------------------------------------------------------------------------------------------------
TREACTION    FEEDA        I  1.00     5  TTREACTION   UREACTOR     OPERATORS      0   1  1.00  0.00
      REACP        O  1.00  5

TBLENDINGA   REACP        I  1.00     2  TTBLENDING   UBLENDER     OPERATORS      0   1  0.50  0.00
      ADD1         I  0.00
      BLENA        O  1.00  2

TBLENDINGB   REACP        I  1.00     2  TTBLENDING   UBLENDER     OPERATORS      0   1  0.50  0.00
      ADD2         I  0.00
      INT1         O  1.00  2

TMIXING1     INT1         I  0.38     2  TTMIXING     UMIXING      OPERATORS      0   1  0.50  0.00
      FEEDB        I  0.62
      PROD1        O  1.00  2

TMIXING2     INT1         I  0.40     2  TTMIXING     UMIXING      OPERATORS      0   1  0.50  0.00
      FEEDD        I  0.60
      PROD2        O  1.00  2

TMIXING3     INT1         I  0.53     2  TTMIXING     UMIXING      OPERATORS      0   1  0.50  0.00
      FEEDC        I  0.47
      PROD3        O  1.00  2

--------------------------------------------------------------------------------------------------------
I/O : Input/Output States


1 Page:  12

*********************************************************************************************************************
*       gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)        Copyright Imperial College 1996        *
*       Running at Imperial College, London at 00:16:29      on Mon Dec 10 2001                                     *
*********************************************************************************************************************


                        ======================
                        ***** S T A T E S *****
                        ======================

Problem involves 12 states of material.
--------------------------------------------------------------------------------------------------------
  StateName    Type      NIS   Unit      Initial Price  D e m a n d    R e c e i p t s/ D e l i v e r i e s
                                                         ------------  ------------------------------------------
                                         Minim  Maxim  R/D  Minim  Maxim From To   Value Priority
--------------------------------------------------------------------------------------------------------
FEEDA                  TANKFA      @@@@@@   0.0    0.0  INF    R   200.0  200.0   0   0  10.0

FEEDB                  TANKFB      @@@@@@   0.0    0.0  INF    R   200.0  200.0   0   0  10.0

FEEDC                  TANKFC      @@@@@@   0.0    0.0  INF    R   200.0  200.0   0   0  10.0

FEEDD                  TANKFD      @@@@@@   0.0    0.0  INF    R   200.0  200.0   0   0  10.0

REACP
ADD1                   TANKA1      @@@@@@   0.0    0.0  INF    R    20.0   20.0   0   0   5.0

ADD2                   TANKA2      @@@@@@   0.0    0.0  INF    R    20.0   20.0   0   0   5.0

INT1                   TANKINT1

PROD1                  TANKP1          0   0.0    0.0  INF    D    20.0   20.0  10  10  20.0   0.0
                                               D    25.0   25.0  16  16  20.0   0.0
                                               D    25.0   25.0  22  22  20.0   0.0
                                               D    30.0   30.0  32  32  20.0   0.0

PROD2                  TANKP2          0   0.0    0.0  INF    D    25.0   25.0  10  10  20.0   0.0
                                               D    30.0   30.0  16  16  20.0   0.0
                                               D    30.0   30.0  22  22  20.0   0.0
                                               D    35.0   35.0  32  32  20.0   0.0

PROD3                  TANKP3          0   0.0    0.0  INF    D    30.0   30.0  10  10  20.0   0.0
                                               D    35.0   35.0  16  16  20.0   0.0
                                               D    35.0   35.0  22  22  20.0   0.0
                                               D    40.0   40.0  32  32  20.0   0.0

BLENA                  TANKBA          0   0.0    0.0  INF    D    15.0   15.0  22  22  15.0   0.0
                                               D    25.0   25.0  32  32  15.0   0.0

--------------------------------------------------------------------------------------------------------
(*)    : STABLE
INF    : INFINITY, i.e.   1000000.00
R/D    : Receipt/Delivery
!!!!!! : WARNING : End-product with no storage
@@@@@@ : WARNING : Feed stage with no initial amount


1 Page:  13

*********************************************************************************************************************
*       gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)        Copyright Imperial College 1996        *
*       Running at Imperial College, London at 00:16:29      on Mon Dec 10 2001                                     *
*********************************************************************************************************************


                    ============================
                    ***** U N I T   T Y P E S *****
                    ============================
```

UnitTypeName	Capacity Minim	Capacity Maxim	Capital Cost Fix	Capital Cost Var	Oper. Cost Fix	Oper. Cost Var
UTSTORAGEP	10.0	1000.0	0.00	0.00	0.00	0.00
UTSTORAGEF	10.0	1000.0	0.00	0.00	0.00	0.00
UTREACTOR	10.0	1000.0	0.00	10.00	0.00	0.00
UTBLENDER	10.0	1000.0	0.00	10.00	0.00	0.00
UTMIXER	10.0	1000.0	0.00	10.00	0.00	0.00

1 Page: 14

```
******************************************************************************************************************
*      gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)        Copyright Imperial College 1996        *
*      Running at Imperial College, London at 00:16:29     on Mon Dec 10 2001                                      *
******************************************************************************************************************
```

```
==================
***** U N I T S *****
==================
```

Problem involves 14 Equipment Units.

UnitName	#	Capacity	Type	T/S/U	Name	SizeFactors Minim	SizeFactors Maxim	Unavailable No.	Unavailable From	Unavailable To
TANKFA	1	T.B.D.	UTSTORAGEF	S	FEEDA	0.00	1.00			
TANKFB	1	T.B.D.	UTSTORAGEF	S	FEEDB	0.00	1.00			
TANKFC	1	T.B.D.	UTSTORAGEF	S	FEEDC	0.00	1.00			
TANKFD	1	T.B.D.	UTSTORAGEF	S	FEEDD	0.00	1.00			
TANKA1	1	T.B.D.	UTSTORAGEF	S	ADD1	0.00	1.00			
TANKA2	1	T.B.D.	UTSTORAGEF	S	ADD2	0.00	1.00			
TANKINT1	1	T.B.D.	UTSTORAGEF	S	INT1	0.00	1.00			
TANKP1	1	T.B.D.	UTSTORAGEP	S	PROD1	0.00	1.00			
TANKP2	1	T.B.D.	UTSTORAGEP	S	PROD2	0.00	1.00			
TANKP3	1	T.B.D.	UTSTORAGEP	S	PROD3	0.00	1.00			
TANKBA	1	T.B.D.	UTSTORAGEP	S	BLENA	0.00	1.00			
UREACTOR	1	T.B.D.	UTREACTOR	T	TREACTION	0.00	1.00			
UBLENDER	2	T.B.D.	UTBLENDER	T	TBLENDINGB	0.00	1.00			
				T	TBLENDINGA	0.00	1.00			
UMIXING	3	T.B.D.	UTMIXER	T	TMIXING3	0.00	1.00			
				T	TMIXING2	0.00	1.00			
				T	TMIXING1	0.00	1.00			

\#   : Multiplicity
T/S/U : Task/Storage/Utility
T.B.D. : To be determined

1 Page: 15

```
******************************************************************************************************************
*      gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)        Copyright Imperial College 1996        *
*      Running at Imperial College, London at 00:16:29     on Mon Dec 10 2001                                      *
******************************************************************************************************************
```

```
==========================
***** U T I L I T I E S *****
==========================
```

Problem involves 1 Utilities.

UtilityName	T/S	Name	Available From	Available To	Available Level	Available Cost
OPERATORS	T	TMIXING3	0	31	10.00	0.00
	T	TMIXING2				
	T	TMIXING1				
	T	TBLENDINGB				
	T	TBLENDINGA				
	T	TREACTION				

T/S : Task/Source

1 Page: 16

```
******************************************************************************************************************
*      gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)        Copyright Imperial College 1996        *
*      Running at Imperial College, London at 00:16:29     on Mon Dec 10 2001                                      *
******************************************************************************************************************
```

1 Page: 17

****
*   gBSS, general Batch Scheduling System vs. 1.2A (1 May 1996)   Copyright Imperial College 1996   *
*   Running at Imperial College, London at 00:16:29   on Mon Dec 10 2001   *
****

===
***** M A T H E M A T I C A L   M O D E L *****
===

Time horizon	:	32.00 real units
Time discretisation interval	:	1.00 real units
Intervals in time horizon	:	32 discrete units
Problem type	:	Short-Term Scheduling

=======================================
***** M I L P   S T A T I S T I C S *****
=======================================

Number of Integer Variables	:	454
Total Number of Variables	:	1335
Number of Constraints	:	2346
Number of Non-zero Elements	:	7066
Total Number of Search LP Calls	:	0
Number of Simplex Iterations	:	0
Total CPU Requirement (seconds)	:	35.000
Total LP CPU requirement (seconds)	:	35.000
CPU per LP (ms)	:	
CPU per LP Iteration (ms)	:	
Fully Relaxed LP Solution	:	0.00
Optimal Value of the Objective Function	:	-2137.51
Integrality Gap	:	1.0000
Relative Margin of optimality	:	0.100

1 Page: 18

****
*   gBSS, general Batch Scheduling System vs. 1.2A (1 May 1996)   Copyright Imperial College 1996   *
*   Running at Imperial College, London at 00:16:29   on Mon Dec 10 2001   *
****

===
***** O P T I M A L   S O L U T I O N *****
===

Selected Equipment

Unit	Type	Capacity	Fixed Cost	Variable Cost
UREACTOR	Proc/Store	49.35	0.000	493.51
UBLENDER	:a Proc/Store	24.70	0.000	247.00
UBLENDER	:b Proc/Store	24.70	0.000	247.00
UMIXING	:a Proc/Store	30.00	0.000	300.00
UMIXING	:b Proc/Store	25.00	0.000	250.00
UMIXING	:c Proc/Store	20.00	0.000	200.00
Total:			0.000	1737.51

Total Capital Cost : 1737.506

Processing/Storage Equipment Utilisation

Unit	Init.Condition	Task/[State]	Starts	Ends	Batchsize	Condition Time/Capacity	Utilisation (%)
UREACTOR		TREACTION	0.00	5.00	49.35		
		.	7.00	12.00	49.35		
		.	12.00	17.00	39.66		
		.	18.00	23.00	24.98		
		.	23.00	28.00	36.66		
						0.00/ 0.00	
UBLENDER	:a	TBLENDINGB	5.00	7.00	24.70		
		.	12.00	14.00	24.70		
		TBLENDINGA	17.00	19.00	15.00		
		TBLENDINGB	23.00	25.00	24.70		
		.	28.00	30.00	12.00		
						0.00/ 0.00	
UBLENDER	:b	TBLENDINGB	5.00	7.00	24.70		
		.	12.00	14.00	24.70		
		.	17.00	19.00	24.70		
		TBLENDINGA	23.00	25.00	0.30		
		.	28.00	30.00	24.70		
						0.00/ 0.00	
UMIXING	:a	TMIXING3	8.00	10.00	30.00		
		TMIXING2	11.00	13.00	30.00		
		TMIXING1	14.00	16.00	25.00		
		TMIXING3	20.00	22.00	10.00		

1 Page: 19

****

```
 *         gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996          *
*         Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                          *
*************************************************************************************************************************

                    TMIXING1            25.00     27.00          5.00
                    TMIXING2            30.00     32.00         30.00
                                                                   0.00/ 0.00
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
UMIXING      :b           TMIXING2              8.00      10.00           25.00
                    TMIXING3            14.00     16.00         15.00
                    TMIXING1            16.00     18.00         25.00
                    TMIXING3            18.00     20.00         25.00
                    TMIXING2            20.00     22.00         10.00
                    TMIXING3            25.00     27.00         20.00
                    TMIXING1            27.00     29.00         25.00
                                                                   0.00/ 0.00
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
UMIXING      :c           TMIXING1              8.00      10.00           20.00
                    TMIXING3            14.00     16.00         20.00
                    TMIXING2            19.00     21.00         20.00
                    TMIXING3            27.00     29.00         20.00
                    TMIXING2            29.00     31.00          5.00
                                                                   0.00/ 0.00
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


1 Page:  20

*************************************************************************************************************************
 *         gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996          *
*         Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                          *
*************************************************************************************************************************

                              --------------------------------
                              Dedicated  Storage  Utilisation
          ------------------------------------------------------------------------------------------------------------
          State        Capacity       From        To       Added     Removed      Storage    Wasted
          ------------------------------------------------------------------------------------------------------------

          FEEDA          1000.00        0.00      0.00     200.00
                            0.00      7.00                            150.65
                            7.00     12.00                            101.30
                           12.00     18.00                             61.64
                           18.00     23.00                             36.66
                           23.00     32.00                              0.00
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          FEEDB          1000.00        0.00      0.00     200.00
                            0.00      8.00                            200.00
                            8.00     14.00                            187.60
                           14.00     16.00                            172.10
                           16.00     25.00                            156.60
                           25.00     27.00                            153.50
                           27.00     32.00                            138.00
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          FEEDC          1000.00        0.00      0.00     200.00
                            0.00      8.00                            200.00
                            8.00     14.00                            185.90
                           14.00     18.00                            169.45
                           18.00     20.00                            157.70
                           20.00     25.00                            153.00
                           25.00     27.00                            143.60
                           27.00     32.00                            134.20
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          FEEDD          1000.00        0.00      0.00     200.00
                            0.00      8.00                            200.00
                            8.00     11.00                            185.00
                           11.00     19.00                            167.00
                           19.00     20.00                            155.00
                           20.00     29.00                            149.00
                           29.00     30.00                            146.00
                           30.00     32.00                            128.00
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          ADD1          1000.00        0.00      0.00      20.00
                            0.00     17.00                             20.00
                           17.00     23.00                             19.98
                           23.00     28.00                             19.98
                           28.00     32.00                             19.96
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          ADD2           1000.00        0.00      0.00      20.00
                            0.00      5.00                             20.00
                            5.00     12.00                             19.95
                           12.00     17.00                             19.90
                           17.00     23.00                             19.88
                           23.00     28.00                             19.85
                           28.00     32.00                             19.84
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          INT1        1000.00        0.00      7.00                             0.00
                            7.00      8.00                             49.40
                            8.00     11.00                             15.90
                           11.00     14.00                              3.90
                           14.00     16.00                             25.25

1 Page:  21

*************************************************************************************************************************
 *         gBSS, general Batch Scheduling System  vs. 1.2A  (1 May 1996)          Copyright Imperial College 1996          *
*         Running at Imperial College, London at 00:16:29       on Mon Dec 10 2001                                          *
*************************************************************************************************************************

                           16.00     18.00                             15.75
                           18.00     19.00                              2.50
                           19.00     20.00                             19.20
                           20.00     25.00                              9.90
                           25.00     27.00                             22.10
                           27.00     29.00                              2.00
                           29.00     32.00                              0.00
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          PROD1          1000.00       10.00     10.00             20.00
                           16.00     16.00             25.00
                            0.00     18.00                              0.00
                           18.00     22.00                             25.00
```

	22.00	22.00		25.00		
	22.00	27.00			0.00	
	27.00	29.00			5.00	
	29.00	32.00			30.00	
	32.00	32.00		30.00		
	32.00	32.00			0.00	
PROD2	1000.00	10.00	10.00		25.00	
	0.00	13.00			0.00	
	13.00	16.00			30.00	
	16.00	16.00		30.00		
	16.00	21.00			0.00	
	21.00	22.00			20.00	
	22.00	22.00		30.00		
	22.00	31.00			0.00	
	31.00	32.00			5.00	
	32.00	32.00		35.00		
	32.00	32.00			0.00	
PROD3	1000.00	10.00	10.00		30.00	
	16.00	16.00		35.00		
	0.00	20.00			0.00	
	20.00	22.00			25.00	
	22.00	22.00		35.00		
	22.00	27.00			0.00	
	27.00	29.00			20.00	
	29.00	32.00			40.00	
	32.00	32.00		40.00		
	32.00	32.00			0.00	
BLENA	1000.00	0.00	19.00			0.00
	19.00	22.00			15.00	
	22.00	22.00		15.00		
	22.00	25.00			0.00	
	25.00	30.00			0.30	
	30.00	32.00			25.00	
	32.00	32.00		25.00		
	32.00	32.00			0.00	

Utility Utilisation Profiles

Utilisation

1 Page: 22

****
* gBSS, general Batch Scheduling System vs. 1.2A (1 May 1996) Copyright Imperial College 1996 *
* Running at Imperial College, London at 00:16:29 on Mon Dec 10 2001 *
****

Utility	From	To	Amount	%	Cost
OPERATORS	0.00	1.00	1.00	10.00	0.00
	5.00	6.00	1.00	10.00	0.00
	7.00	8.00	2.00	20.00	0.00
	8.00	9.00	1.50	15.00	0.00
	11.00	12.00	0.50	5.00	0.00
	12.00	13.00	2.50	25.00	0.00
	14.00	15.00	1.50	15.00	0.00
	16.00	17.00	1.50	15.00	0.00
	17.00	18.00	1.00	10.00	0.00
	18.00	19.00	1.50	15.00	0.00
	19.00	20.00	1.00	10.00	0.00
	20.00	21.00	1.00	10.00	0.00
	21.00	22.00	0.50	5.00	0.00
	23.00	24.00	2.00	20.00	0.00
	25.00	26.00	1.50	15.00	0.00
	27.00	28.00	1.00	10.00	0.00
	28.00	29.00	1.50	15.00	0.00
	29.00	30.00	1.00	10.00	0.00
	30.00	31.00	0.50	5.00	0.00
	Totals :		24.00	7.50	0.00

1 Page: 23

****
* gBSS, general Batch Scheduling System vs. 1.2A (1 May 1996) Copyright Imperial College 1996 *
* Running at Imperial College, London at 00:16:29 on Mon Dec 10 2001 *
****

Order Deliveries Information

Order	Objective	Material	Amount	Time Window Priority	Unit Value/ Amount	Deliveries Amount	Deliveries Time	Deliveries Value/Penalty
ORDER001	V	PROD1	20.0	10	20.0	20.00	10.00	400.00
ORDER002	V	PROD1	25.0	16	20.0	25.00	16.00	500.00
ORDER003	V	PROD1	25.0	22	20.0	25.00	22.00	500.00
ORDER004	V	PROD1	30.0	32	20.0	30.00	32.00	600.00
ORDER005	V	PROD2	25.0	10	20.0	25.00	10.00	500.00
ORDER006	V	PROD2	30.0	16	20.0	30.00	16.00	600.00
ORDER007	V	PROD2	30.0	22	20.0	30.00	22.00	600.00
ORDER008	V	PROD2	35.0	32	20.0	35.00	32.00	700.00
ORDER009	V	PROD3	30.0	10	20.0	30.00	10.00	600.00
ORDER010	V	PROD3	35.0	16	20.0	35.00	16.00	700.00
ORDER011	V	PROD3	35.0	22	20.0	35.00	22.00	700.00
ORDER012	V	PROD3	40.0	32	20.0	40.00	32.00	800.00
ORDER013	V	BLENA	15.0	22	15.0	15.00	22.00	225.00
ORDER014	V	BLENA	25.0	32	15.0	25.00	32.00	375.00

V : maximum value
D(L) : minimum delay, linear penalty
D(Q) : minimum delay, quadratic penalty
* : order not at maximum value

State Order Summary

Material Name	Opening Stock	Total Receipts	Total Deliveries	Total Production (Consumption)	Closing Stock
FEEDA	0.00	200.00	0.00	(200.00)	0.00
FEEDB	0.00	200.00	0.00	(62.00)	138.00
FEEDC	0.00	200.00	0.00	(65.80)	134.20
FEEDD	0.00	200.00	0.00	(72.00)	128.00
REACP	0.00	0.00	0.00	0.00	0.00
ADD1	0.00	20.00	0.00	(0.04)	19.96
ADD2	0.00	20.00	0.00	(0.16)	19.84
INT1	0.00	0.00	0.00	0.00	0.00
PROD1	0.00	0.00	100.00	100.00	0.00
PROD2	0.00	0.00	120.00	120.00	0.00
PROD3	0.00	0.00	140.00	140.00	0.00
BLENA	0.00	0.00	40.00	40.00	0.00

*** OBJECTIVE BREAKDOWN ***

Total
SubTotal

1 Page: 24

****
* gBSS, general Batch Scheduling System vs. 1.2A (1 May 1996) Copyright Imperial College 1996 *
* Running at Imperial College, London at 00:16:29 on Mon Dec 10 2001 *
****

Deliveries Value

Order	State	Amount	Value
ORDER001	PROD1	20.00	400.00
ORDER002	PROD1	25.00	500.00
ORDER003	PROD1	25.00	500.00
ORDER004	PROD1	30.00	600.00
ORDER005	PROD2	25.00	500.00
ORDER006	PROD2	30.00	600.00
ORDER007	PROD2	30.00	600.00
ORDER008	PROD2	35.00	700.00
ORDER009	PROD3	30.00	600.00
ORDER010	PROD3	35.00	700.00
ORDER011	PROD3	35.00	700.00
ORDER012	PROD3	40.00	800.00
ORDER013	BLENA	15.00	225.00
ORDER014	BLENA	25.00	375.00
		7800.00	

Receipts Value

Order	State	Amount	Value
1	FEEDA	200.00	(2000.00)
1	FEEDB	200.00	(2000.00)
1	FEEDC	200.00	(2000.00)
1	FEEDD	200.00	(2000.00)
1	ADD1	20.00	(100.00)
1	ADD2	20.00	(100.00)
			(8200.00)

Closing Stocks

State	Amount	Value
FEEDA	0.00	0.00
FEEDB	138.00	0.00
FEEDC	134.20	0.00
FEEDD	128.00	0.00
ADD1	19.96	0.00
ADD2	19.84	0.00
PROD1	0.00	0.00
PROD2	0.00	0.00
PROD3	0.00	0.00
BLENA	0.00	0.00

Cost of Utilities

Utility	Amount	Value
OPERATORS	24.00	(0.00)

-400.00

14. Appendix C, GAMS Input Files for Hydrolubes Plant

```
* RTN Example for Hydrolubes Plant (N.Shah, PhD Thesis, p.p. 69-75)
* Aggregated RTN (First order)
*
*
$onnestcom inlinecom {{ }} eolcom ##
$OFFSYMLIST
$OFFSYMXREF
OPTION LIMROW=10000;
OPTION LIMCOL=10000;
OPTION SOLPRINT=ON;
OPTION ITERLIM=1000000;
OPTION RESLIM=100000;
OPTION DECIMALS=8;
OPTION OPTCR=0.000001;
OPTION PROFILE=2;
*
*
* TIMING HORIZON 32 hrs (1 hour intervals)
*
* TASKS 1:REACTION  2,3:BLENDING A,B  4,5,6:MIXING 1,2,3
*
* RESOURCES 1:FEED A, 2:FEED B, 3:FEED C, 4:FEED D, 5:ReactionProduct,
*           6,7:AD1,AD2  8:Int1, 9,10,11:P1,P2,P3  12:B1A, 13:Reactor,
*           14 Blenders, 15:Mixers, 16:Operators
*
SETS
       K  TASKS             /K1*K6/
       R  RESOURCES   /R1*R16/
       T  TIME(INT)   /T1*T32/
       th THETA             /0*4/;

ALIAS (th,thp);
ALIAS (T,Tp);

SET    myth(th)       /1*4/;
SET    mythp(th)      /1*4/;

PARAMETER tk(K)        TASK DURATION (intervals)
               /K1  4
               K2   1
               K3   1
               K4   1
               K5   1
               K6   1/;
```

```
PARAMETER h AGGREGATE TIME PERIOD;
h=8;

PARAMETER NL(T)  NUMBER OF TIME INTERVAL BASED ON h  ;
NL(T)=CARD(T)/h;

****declaration of parameters mu(m) and ni(v).
PARAMETERS m(K,R,thp), v(K,R,thp);

****TASK1****
* Reaction
m('K1','R13','0')=-1.0;
m('K1','R13','4')=1.0;
m('K1','R16','0')=-1.0;
m('K1','R16','1')=0.8;
m('K1','R16','3')=-0.3;
m('K1','R16','4')=0.5;
v('K1','R1','0')=-1.0;
v('K1','R1','4')=0.0;
v('K1','R5','0')=0.0;
v('K1','R5','4')=1.0;
****TASK2****
* Blending A
m('K2','R14','0')=-1.0;
m('K2','R14','1')=1.0;
m('K2','R16','0')=-0.5;
m('K2','R16','1')=0.5;
v('K2','R5','0')=-0.999;
v('K2','R5','1')=0.0;
v('K2','R7','0')=-0.001;
v('K2','R7','1')=0.0;
v('K2','R12','0')=0.0;
v('K2','R12','1')=1.0;
****TASK3****
* Blending B
m('K3','R14','0')=-1.0;
m('K3','R14','1')=1.0;
m('K3','R16','0')=-0.5;
m('K3','R16','1')=0.5;
v('K3','R5','0')=-0.999;
v('K3','R5','1')=0.0;
v('K3','R6','0')=-0.001;
v('K3','R6','1')=0.0;
v('K3','R8','0')=0.0;
v('K3','R8','1')=1.0;
****TASK4****
```

```
* Mixing 1
m('K4','R15','0')=-1.0;
m('K4','R15','1')=1.0;
m('K4','R16','0')=-0.5;
m('K4','R16','1')=0.5;
v('K4','R9','0')=0.0;
v('K4','R9','1')=1.0;
v('K4','R2','0')=-0.62;
v('K4','R2','1')=0.0;
v('K4','R8','0')=-0.38;
v('K4','R8','1')=0.0;
****TASK5****
* Mixing 2
m('K5','R15','0')=-1.0;
m('K5','R15','1')=1.0;
m('K5','R16','0')=-0.5;
m('K5','R16','1')=0.5;
v('K5','R10','0')=0.0;
v('K5','R10','1')=1.0;
v('K5','R3','0')=-0.60;
v('K5','R3','1')=0.0;
v('K5','R8','0')=-0.40;
v('K5','R8','1')=0.0;
****TASK6****
* Mixing 3
m('K6','R15','0')=-1.0;
m('K6','R15','1')=1.0;
m('K6','R16','0')=-0.5;
m('K6','R16','1')=0.5;
v('K6','R11','0')=0.0;
v('K6','R11','1')=1.0;
v('K6','R4','0')=-0.47;
v('K6','R4','1')=0.0;
v('K6','R8','0')=-0.53;
v('K6','R8','1')=0.0;

PARAMETERS mA_0(K,R,th), vA_0(K,R,th), mA_1(K,R,th), vA_1(K,R,th);

mA_1(K,R,th)=sum(thp$(ord(thp) GE ord(th) and ord(th) LE tk(K)+1),(ord(thp)-
1)*m(K,R,thp));

vA_1(K,R,th)=sum(thp$(ord(thp) GE ord(th) and ord(th) LE tk(K)+1),(ord(thp)-
1)*v(K,R,thp));

mA_0(K,R,th)=sum(thp$(ord(thp) GE ord(th) and ord(th) LE tk(K)+1),m(K,R,thp));

vA_0(K,R,th)=sum(thp$(ord(thp) GE ord(th) and ord(th) LE tk(K)+1),v(K,R,thp));
```

```
DISPLAY mA_1;
DISPLAY vA_1;
DISPLAY mA_0;
DISPLAY vA_0;

****resource constraints****
PARAMETER Rmax(R);
* Feed A
Rmax('R1')=10000;
* Feed B
Rmax('R2')=10000;
* Feed C
Rmax('R3')=10000;
* Feed D
Rmax('R4')=10000;
* Reaction Product
Rmax('R5')=0;
* AD1
Rmax('R6')=3000;
* AD2
Rmax('R7')=3000;
* Int1
Rmax('R8')=100;
* Product 1
Rmax('R9')=6000;
* Product 2
Rmax('R10')=6000;
* Product 3
Rmax('R11')=6000;
* Blend A
Rmax('R12')=4000;
* Reactor
Rmax('R13')=1;
* Blenders
Rmax('R14')=2;
* Mixers
Rmax('R15')=3;
* Operators
Rmax('R16')=10;

****volume constraints*****
PARAMETERS Vmax(K);

* Reaction
Vmax('K1')=50;
* Blending A
```

```
Vmax('K2')=45;
* Blending B
Vmax('K3')=45;
* Mixing 1
Vmax('K4')=45;
* Mixing 2
Vmax('K5')=45;
* Mixing 3
Vmax('K6')=45;

****demand of products****
PARAMETERS DEL(R,Tp), DEL_0(R,T), DEL_1(R,T);
* Product One
DEL('R9','T10')=20;
DEL('R9','T16')=25;
DEL('R9','T22')=25;
DEL('R9','T32')=30;
* Product Two
DEL('R10','T10')=25;
DEL('R10','T16')=30;
DEL('R10','T22')=30;
DEL('R10','T32')=35;
* Product Three
DEL('R11','T10')=30;
DEL('R11','T16')=35;
DEL('R11','T22')=35;
DEL('R11','T32')=40;
* Blend A
DEL('R12','T22')=40;
DEL('R12','T32')=46;

DEL_0(R,T)=sum(Tp$(ord(Tp) GE ord(T)-h+1 and ord(Tp) LE ord(T)), DEL(R,Tp));
DEL_1(R,T)=sum(Tp$(ord(Tp) GE ord(T)-h+1 and ord(Tp) LE ord(T)), (ord(T)-
ord(Tp)+1)*DEL(R,Tp));

DEL(R,T) = 0.0;
DEL_0(R,T)=0.0;
DEL_1(R,T)=0.0;

****costs*****
PARAMETERS
  CF(R)        value of each material resource R at the end of horizon
  CAPC(R)      capital cost of each resource R;

* Feed A
CF('R1')=10;
* Feed B
```

```
CF('R2')=10;
* Feed C
CF('R3')=10;
* Feed D
CF('R4')=10;
* AD1
CF('R6')=5;
* AD2
CF('R7')=5;
* Product 1
CF('R9')=20;
* Product 2
CF('R10')=20;
* Product 3
CF('R11')=20;
* Blend A
CF('R12')=15;

* Feed A
CAPC('R1')=0;
* Feed B
CAPC('R2')=0;
* Feed C
CAPC('R3')=0;
* Feed D
CAPC('R4')=0;
* Reaction Product
CAPC('R5')=0;
* AD1
CAPC('R6')=0;
* AD2
CAPC('R7')=0;
* Product 1
CAPC('R9')=0;
* Product 2
CAPC('R10')=0;
* Product 3
CAPC('R11')=0;
* Blend A
CAPC('R12')=0;
* Reactor
CAPC('R13')=10;
* Blenders
CAPC('R14')=10;
* Mixers
CAPC('R15')=10;
* Operators
```

```
CAPC('R16')=0;

VARIABLES
  RS(R,T)       amount of excess resource R over time interval T
  ARS_0(R,T)    aggregated form of variable
  ARS_1(R,T)    aggregated form of variable (first order)
  ksi(K,T)      continuous variable characterize the task K as batchsize
  Aksi_0(K,T)   aggregated continuous variable (zeroth order)
  Aksi_1(K,T)   aggregated continuous variable (first order)
  Z1            objective function;

INTEGER VARIABLES
  N(K,T)        integer variable characterize the task K as batches
  AN_0(K,T)     aggregated integer variable (zeroth order)
  AN_1(K,T)     aggregated integer variable (first order)
  RSI(R)        maximum amount of resource R required;

N.UP(K,T) = 10000000;
AN_0.UP(K,T) = 10000000;
AN_1.UP(K,T) = 10000000;

RSI.LO('R1') = 200;
RSI.UP('R1') = 200;
RSI.LO('R2') = 200;
RSI.UP('R2') = 200;
RSI.LO('R3') = 200;
RSI.UP('R3') = 200;
RSI.LO('R4') = 200;
RSI.UP('R4') = 200;
RSI.LO('R5') = 0;
RSI.UP('R5') = 0;
RSI.LO('R6') = 20;
RSI.UP('R6') = 20;
RSI.LO('R7') = 20;
RSI.UP('R7') = 20;
RSI.LO('R8') = 0;
RSI.UP('R8') = 0;
RSI.LO('R9') = 0;
RSI.UP('R9') = 0;
RSI.LO('R10') = 0;
RSI.UP('R10') = 0;
RSI.LO('R11') = 0;
RSI.UP('R11') = 0;
RSI.LO('R12') = 0;
RSI.UP('R12') = 0;
RSI.LO('R13') = 0;
RSI.UP('R13') = 1000;
```

```
RSI.LO('R14') = 0;
RSI.UP('R14') = 1000;
RSI.LO('R15') = 0;
RSI.UP('R15') = 1000;
RSI.LO('R16') = 10;
RSI.UP('R16') = 10;

EQUATIONS ARB0_0(R,T), ARB0_1(R,T), ARB_0(R,T), ARB_1(R,T), CC1(R,T), CC2(R,T), CC3(R,T),
CC4(R,T), CC5(K,T), CC6(K,T), CC7(K,T), CC8(K,T), CC9(R,T), CC10(R,T), CC11(R,T),
CC12(R,T), OC1(K,T), OC2(K,T), OC3(K,T), OC4(K,T), OC5(K,T,myth), OC6(K,T,myth), REQ(R,T),
REQ2(R,T), OBJ2;

****MATHEMATICAL FORMULATION*****

****First Order Formulation****
****Excess Resource Balances (RB)****

ARB0_0(R,T)$(ord(T) EQ h+1)..
RS(R,T) =E= RSI(R) + sum(K,mA_0(K,R,'0')*AN_0(K,T)+vA_0(K,R,'0')*Aksi_0(K,T)) +
sum(K,sum(myth$((ord(myth) GE 1) and (ord(myth) LE tk(K))), mA_0(K,R,myth)*(-N(K,T+(1-
ord(myth)))) +  vA_0(K,R,myth)*(-ksi(K,T+(1-ord(myth)))))) - DEL_0(R,T);

ARB0_1(R,T)$(ord(T) EQ h+1)..
ARS_0(R,T) - h*RSI(R) =E= sum(K,mA_0(K,R,'0')*AN_1(K,T) - mA_1(K,R,'1')*AN_0(K,T) +
vA_0(K,R,'0')*Aksi_1(K,T) - vA_1(K,R,'1')*Aksi_0(K,T)) - sum(K,sum(mythp$(ord(mythp) GE 1
and ord(mythp) LE tk(K)), N(K,T-(ord(mythp)-1))*sum(myth$(ord(myth) GE ord(mythp) and
ord(myth) LE tk(K)),(ord(mythp)-ord(myth))*m(K,R,myth)))) -sum(K,sum(mythp$(ord(mythp) GE
1 and ord(mythp) LE tk(K)), ksi(K,T-(ord(mythp)-1))*sum(myth$(ord(myth) GE ord(mythp) and
ord(myth) LE tk(K)),(ord(mythp)-ord(myth))*v(K,R,myth)))) - DEL_1(R,T);

*Zeroth order resource balance*
ARB_0(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE h+1 and ord(T) NE 1).. RS(R,T) =E=
RS(R,T-(h)) + sum(K,mA_0(K,R,'0')*AN_0(K,T)+vA_0(K,R,'0')*Aksi_0(K,T)) +
sum(K,sum(myth$((ord(myth) GE 1) and (ord(myth) LE tk(K))), mA_0(K,R,myth)*(N(K,T-
(h+ord(myth)-1))-N(K,T+(1-ord(myth)))))) + sum(K,sum(myth$((ord(myth) GE 1) and (ord(myth)
LE tk(K))), vA_0(K,R,myth)*(ksi(K,T-(h+ord(myth)-1))-ksi(K,T+(1-ord(myth))))))-
DEL_0(R,T);

*First order resource balance*
ARB_1(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE h+1 and ord(T) NE 1).. ARS_0(R,T) -
h*RS(R,T-(h)) =E= sum(K,mA_0(K,R,'0')*AN_1(K,T) - mA_1(K,R,'1')*AN_0(K,T) +
vA_0(K,R,'0')*Aksi_1(K,T) - vA_1(K,R,'1')*Aksi_0(K,T)) + sum(K,sum(mythp$(ord(mythp) GE 1
and ord(mythp) LE tk(K)), N(K,T-(ord(mythp)+h-1))*sum(myth$(ord(myth) GE ord(mythp) and
ord(myth) LE tk(K)),(h+ord(mythp)-ord(myth))*m(K,R,myth)))) - sum(K,sum(mythp$(ord(mythp)
GE 1 and ord(mythp) LE tk(K)), N(K,T-(ord(mythp)-1))*sum(myth$(ord(myth) GE ord(mythp) and
ord(myth) LE tk(K)),(ord(mythp)-ord(myth))*m(K,R,myth)))) + sum(K,sum(mythp$(ord(mythp) GE
1 and ord(mythp) LE tk(K)),
```

```
 ksi(K,T-(ord(mythp)+h-1))*sum(myth$(ord(myth) GE ord(mythp) and ord(myth) LE
tk(K)),(h+ord(mythp)-ord(myth))*v(K,R,myth)))) - sum(K,sum(mythp$(ord(mythp) GE 1 and
ord(mythp) LE tk(K)), ksi(K,T-(ord(mythp)-1))*sum(myth$(ord(myth) GE ord(mythp) and
ord(myth) LE tk(K)),(ord(mythp)-ord(myth))*v(K,R,myth)))) - DEL_1(R,T);

****Zeroth order aggregate resource capacity constraint****
CC1(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. ARS_1(R,T) -2*ARS_0(R,T) + RS(R,T)
=L= 0.5*(h-1)*(h-2)*Rmax(R);

CC2(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. ARS_1(R,T) -2*ARS_0(R,T) + RS(R,T)
=G= 0;

CC3(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. h*ARS_0(R,T) - ARS_1(R,T) - (h-
1)*RS(R,T) =L= 0.5*(h-1)*(h-2)*Rmax(R);

CC4(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. h*ARS_0(R,T) - ARS_1(R,T) - (h-
1)*RS(R,T) =G= 0;

****First order continuous extent non-negativity constraints***
CC5(K,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. Aksi_1(K,T) -
(tk(K)+1)*Aksi_0(K,T) + sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (tk(K)+1-
ord(myth))*ksi(K,T+(1-ord(myth)))) =G= 0;

CC6(K,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. h*Aksi_0(K,T) - Aksi_1(K,T) -
sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (h-ord(myth))*ksi(K,T+(1-ord(myth))))
=G= 0;

CC7(K,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. AN_1(K,T) - (tk(K)+1)*AN_0(K,T) +
sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (tk(K)+1-ord(myth))*N(K,T+(1-
ord(myth)))) =G= 0;

CC8(K,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. h*AN_0(K,T) - AN_1(K,T) -
sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (h-ord(myth))*N(K,T+(1-ord(myth)))) =G=
0;

*Linking variable excess resource capacity constraints*
CC9(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. RS(R,T) =L= Rmax(R);

CC10(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. RS(R,T) =G= 0;

*Zeroth order aggregate resource capacity constraints*
CC11(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. ARS_0(R,T) - RS(R,T) =L= (h-
1)*Rmax(R);

CC12(R,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. ARS_0(R,T) - RS(R,T) =G= 0;

****Operational Constraints****
```

```
*First order aggregate operational constraints*
OC1(K,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. 0*(AN_1(K,T) - (tk(K)+1)*AN_0(K,T)
+ sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (tk(K)+1-ord(myth))*N(K,T+(1-
ord(myth))))) =L= Aksi_1(K,T) - (tk(K)+1)*Aksi_0(K,T) + sum(myth$(ord(myth) GE 1 and
ord(myth) LE tk(K)), (tk(K)+1-ord(myth))*ksi(K,T+(1-ord(myth))));

OC2(K,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. Vmax(K)*(AN_1(K,T) -
(tk(K)+1)*AN_0(K,T) + sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (tk(K)+1-
ord(myth))*N(K,T+(1-ord(myth))))) =G= Aksi_1(K,T) - (tk(K)+1)*Aksi_0(K,T) +
sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (tk(K)+1-ord(myth))*ksi(K,T+(1-
ord(myth))));

OC3(K,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. 0*(h*AN_0(K,T) - AN_1(K,T) -
sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)) , (h-ord(myth))*N(K,T+(1-ord(myth)))))
=L= h*Aksi_0(K,T) - Aksi_1(K,T) - sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (h-
ord(myth))*ksi(K,T+(1-ord(myth))));

OC4(K,T)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1).. Vmax(K)*(h*AN_0(K,T) - AN_1(K,T) -
sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)) , (h-ord(myth))*N(K,T+(1-ord(myth)))))
=G= h*Aksi_0(K,T) - Aksi_1(K,T) - sum(myth$(ord(myth) GE 1 and ord(myth) LE tk(K)), (h-
ord(myth))*ksi(K,T+(1-ord(myth))));

*Linking variable operational constraints*
OC5(K,T,myth)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1 and ord(myth) GE 1 and ord(myth)
LE tk(K)).. 0*N(K,T-(ord(myth)-1)) =L= ksi(K,T-(ord(myth)-1));

OC6(K,T,myth)$(ord(T)$MOD(ord(T)-1,h) = 0 and ord(T) NE 1 and ord(myth) GE 1 and ord(myth)
LE tk(K)).. Vmax(K)*N(K,T-(ord(myth)-1)) =G= ksi(K,T-(ord(myth)-1));

****Demands
REQ(R,T)$(ORD(T) EQ CARD(T) and ORD(R) LT 13).. RS(R,T) =L= 1000;
REQ2(R,T)$(ORD(T) EQ CARD(T) and ORD(R) LT 13).. RS(R,T) =G= 0;

****Objective Function****
OBJ2.. Z1 =E= RS('R9','T32') + RS('R10','T32') + RS('R11','T32') + RS('R12','T32') -
sum(R,CAPC(R)*RSI(R));

MODEL ARTN1 /ARB0_0, ARB0_1, ARB_0, ARB_1, CC1, CC2, CC3, CC4, CC5, CC6,   CC7, CC8, CC9,
CC10, CC11, CC12, OC1, OC2, OC3, OC4, OC5, OC6, REQ, REQ2, OBJ2 /;
OPTION MIP=CPLEX;
SOLVE ARTN1 USING MIP MAXIMIZING Z1;
DISPLAY RSI.l;
```