PROCESS PLANT RELIABILITY AND CAPACITY ANALYSIS

William R. Lesyna
E. I. du Pont de Nemours & Co.
Wilmington, Delaware

ABSTRACT

Reliability and capacity analysis of large chemical process plants can be done via discrete-event simulation models. Because of the time and expense involved in building and using these models, we have built a general model which uses optimization techniques imbedded in a simulation framework. This model has been implemented as a programming system which has been used very successfully by engineers who are not trained in simulation or optimization methods. This paper will cover the simulation and optimization aspects of the model and will describe how a process plant is modeled with this tool.

INTRODUCTION

Du Pont's Operations Research Group provides company-wide consulting services and we have often seen the utility of discrete-event simulation models in improving the design of systems. In particular, simulation is a very useful tool to analyze existing or proposed process plants to answer the following types of questions:

- What are the reliabilities of the unit processes and the plant as a whole?
- What is the annual capacity of the plant?
- What should the rate capabilities of the unit processes be?
- How big should the tanks be?
- What are the bottlenecks to increasing the capacity of the plant?

Such questions are frequently difficult to answer without simulation because of the size and complexity of the plants, the random failures, the process interdependencies, and the different possible "operating policies" for the plant. Although simulation can overcome these difficulties, it is an expensive tool which requires skilled analysts and a good deal of time. A typical process plant simulation model might cost $30,000 and require 6 man-months to build. We felt that for this class of problems, it was possible to build a generalized model to make more effective use of the company's consulting resources.

One of our goals in building this generalized model was to enable people who are not experts in simulation or optimization to use it as an effective tool without extensive aid from Operations Research Consultants. We have met that objective. In this paper I will explain the technical underpinnings of this model and discuss some ways in which it is used.

OVERVIEW

Figure 1 shows a schematic representation of a process plant with 25 processes, 21 tanks, and several recycle streams. Analysis of such a system is done in two phases. The first phase, called "equipment analysis" is a separate simulation for each of the 25 processes. Each such process block consists of series and/or parallel combinations of "components" (typically, say 35 of them) which represent equipment pieces (but not tanks) at whatever level of detail is appropriate. Rate and failure data are provided for each of these components. The simulation of each process produces availability and production reports for that process considered in isolation, as well as a history of the process rate capabilities as a function of time.

These "time-rate histories" are used as input to the second phase, called "final analysis". This phase is also a simulation, but this time for all the process blocks and tanks at one time. The time-rate histories from the first phase are used to define when events occur, and what the current process rate capabilities are. Between events the actual process rates (and, therefore, the tank levels) are determined by the linear programming

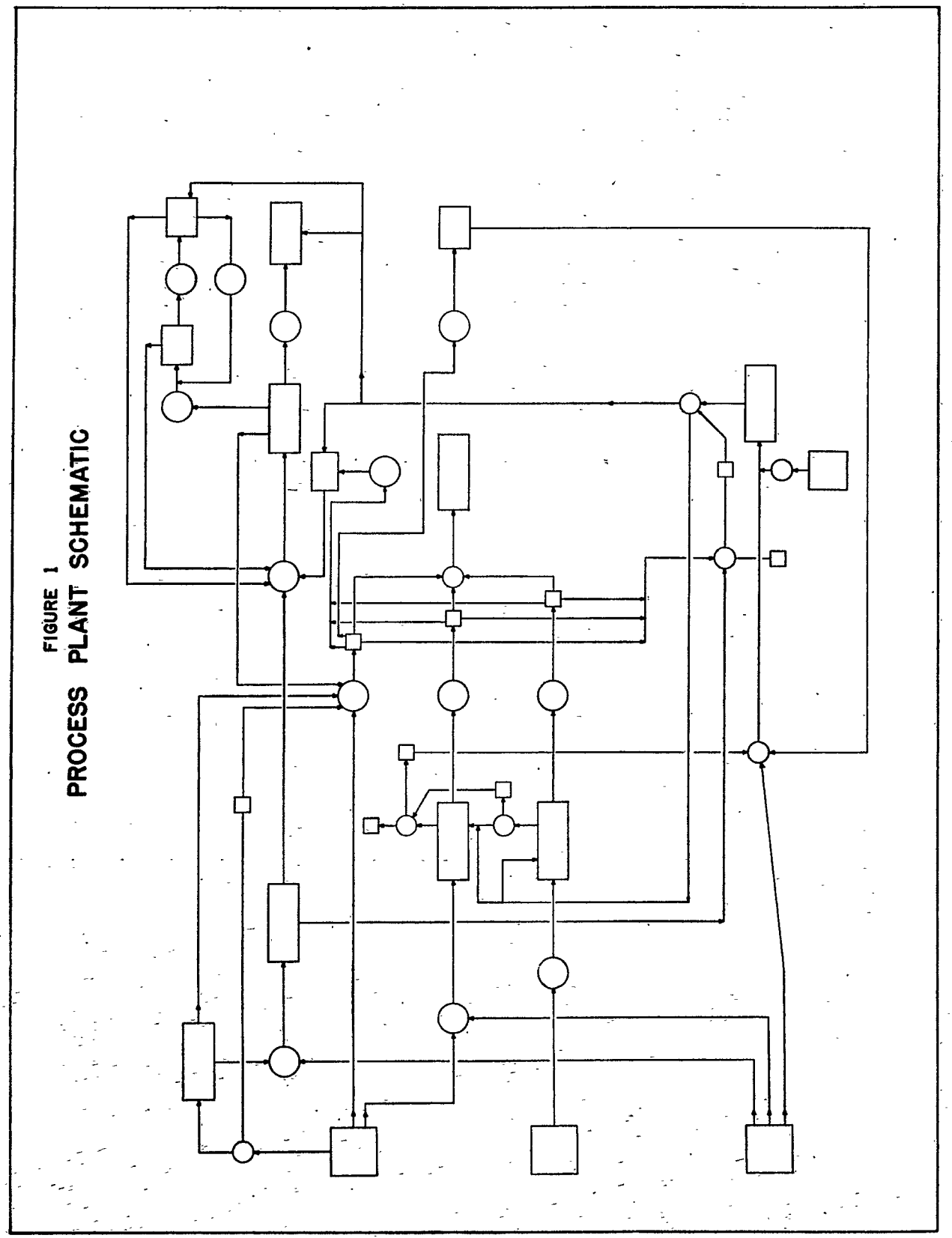
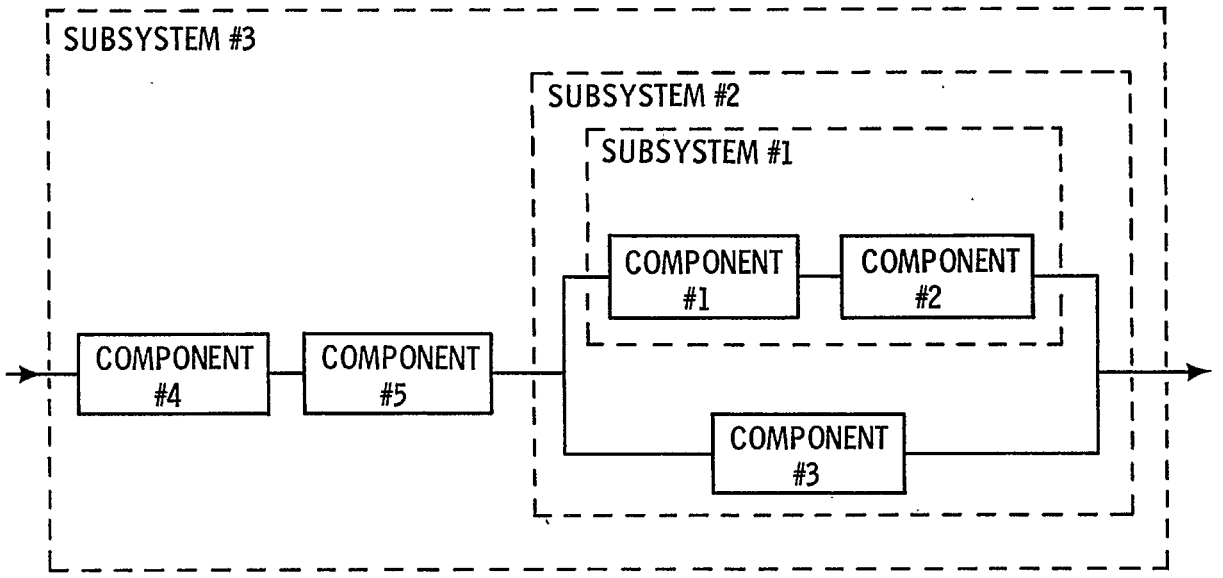


FIGURE 1
PROCESS PLANT SCHEMATIC

FIGURE 2
PROCESS EXAMPLE



optimization technique. The goal of this optimization is to maximize the throughput of the plant. Various reports are generated to document the performance of the process and plant models.

Some ideas similar to these can be found in Reference (1).

EQUIPMENT ANALYSIS SIMULATION

The most fundamental unit of equipment in our model is called a component. Rate and failure data are entered at the component level. Components which are in series with other components or in parallel with other components make up a "subsystem". A subsystem, in turn, can be in series or in parallel with other components and/or subsystems. The highest level subsystem is called a "system" or a "process". Note that tanks are not allowed within a process. Figure 2 shows a simple process consisting of five components. All subsystem boundaries have been indicated. Note that subsystem #3 is actually the process.

The data input for each component includes:

- Maximum and minimum rates
- Number of random failures per year
- Repair time distribution data
- Number and timing of scheduled shutdowns

Given the rate and failure data for each of these components, and the desired simulation time, the model proceeds by generating events for each component. Each event is either the start or the end of a repair. During each time interval when there are no events, the model determines the maximum and minimum rates at which material can flow through each subsystem as well as through the process as a whole. These rates and the times at which they change are used for two purposes:

- Availability and production for each component and subsystem, and for the process, are calculated and reported on.
- The rates and times, called a "time-rate history" are saved on a file for later use in analyzing the plant as a whole.

Figure 3 shows the simple structure of a time-rate history.

FIGURE 3
STRUCTURE OF TIME-RATE HISTORY

TIME (HOURS)	MAXIMUM POSSIBLE RATE	MINIMUM POSSIBLE RATE
0.0	100	50
10.3	75	50
11.3	100	50
17.4	0	0
20.4	100	50
⋮	⋮	⋮

As mentioned earlier, an event is either the start or the end of a component repair period. The number of pending events (on the "future events chain") is the same as number of components. These events are stored in multiple lists, each of which is ordered by event time.

The number of lists is chosen to optimize the list processing time. When inserting a new event, we scan the lists to find if the event can be added at the start or the end of any list. If not, we add the event (in time order) to the interior of the shortest list. When searching for the next event, we scan the lists to find the earliest of the events at the start of each list.

TWO MODEL VERSIONS

Some results from reliability theory (2) will illustrate why we have two different equipment analysis versions of our model. Suppose we are given two components in series, each with a reliability of .60. One estimate of system reliability is:

$$.60 \times .60 = .36$$

However, this is valid only if you assume that while one of the components is failed (being repaired), the other component continues to run and is, therefore, capable of failing at the same time. Since the failure of one component in this simple system causes the throughput of the system to be zero, it is possible that the non-failed component does not run during the repair time of the other component and is not, therefore, liable to failure during that time. Under this assumption, the reliability of the system is:

$$\frac{1}{1 + \left(\frac{1-.6}{.6} + \frac{1-.6}{.6}\right)} = .43$$

Notice this is significantly better than the previously calculated .36. Neither of these answers is more correct than the other. Which one to use depends on how the system in question actually operates. Although the equipment analysis calculations are not done analytically, the above results illustrate the ideas behind the different ways the simulations proceed in the two versions.

OUTPUT REPORTS

Various reports are produced for each equipment analysis:

- Input Data Log - This reports on the input data as read from cards.
- Detail Report - This report gives availability and production data for each component, each subsystem, and the system as a whole.
- Ranking Reports - These reports rank components by increasing availability, increasing maximum capacity, increasing maximum flowrate, increasing mean-time-to-outage, decreasing total mean down-time, and decreasing unscheduled failure time.
- Flowrate Histograms - These reports give tables and plots for flowrates of components and subsystems.

Equipment analysis results are valuable on their own, but usually we go on to do a final analysis, using the saved time-rate history from each equipment analysis.

FINAL ANALYSIS SIMULATION/OPTIMIZATION

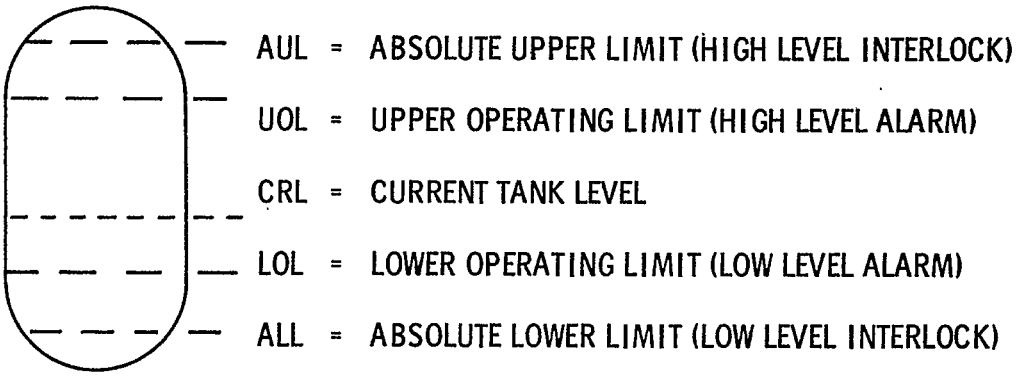
Final analysis consists of a simulation of the entire system. This system consists of processes, tanks, and flow-lines between processes and tanks.

PROCESSES, TANKS & FLOWLINES

Each process is a series/parallel combination of components as described by an equipment analysis.

Each tank is described by specifying its absolute upper limit, upper operating limit, initial level, lower operating limit, and absolute lower limit. Figure 4 shows these levels graphically.

FIGURE 4
TANK SPECIFICATIONS



Flowlines between processes and tanks are described from the process's point of view. A flowline either goes into a process from one or more tanks, or it goes out of a process to one or more tanks.

A process cannot be linked directly to another process. A "dummy tank" (with no capacity) is used to model this. A tank cannot be linked directly to another tank. A "dummy process" (with no failures) is used to model this.

The amount of flow on a flowline is assumed to be the same as the current rate of the process it is linked to, unless a linear scale factor is used to modify it. If a flowline goes to (or comes from) more than one tank, then the split among the tanks is determined by the optimization step discussed below.

After the system structure has been specified, events for all processes are merged into one "master events list", ordered by time. These events come from data saved during the equipment analysis phase. All processes are assumed to be up and running at time zero. That is, each process has an event at time zero giving its normal maximum and minimum rates, and the first event for each process after time zero is the start of a repair period. The final analysis simulation time must be no longer than the shortest of the equipment analysis simulation times.

From the master events list, we know the initial capabilities (maximum and minimum rates) of all the processes. From the tank specifications, we know the initial levels of all the tanks. Now we need to determine the "best" process rates to use until the next event (change in rate capabilities for some process) occurs.

LINEAR PROGRAMMING OPTIMIZATION

Between events the process rates are determined by the linear programming optimization technique (3). The objective of the linear program is to maximize the total amount produced by all the processes. The linear programming constraints are:

- For each process, production (rate x time) is less than or equal to the current process maximum rate times time.
- For each process, production is greater than or equal to the current minimum rate times time.
- For each flowline, production on the flowline equals the production of the process to which it is linked, times the specified scale factor.
- For each tank, the ending level equals the beginning level minus production on

all outgoing flowlines and plus production on all incoming flowlines.
- For each tank, the ending level is between the specified absolute upper and lower limits.
- The length of the time interval is between specified minimum and maximum "time horizons".

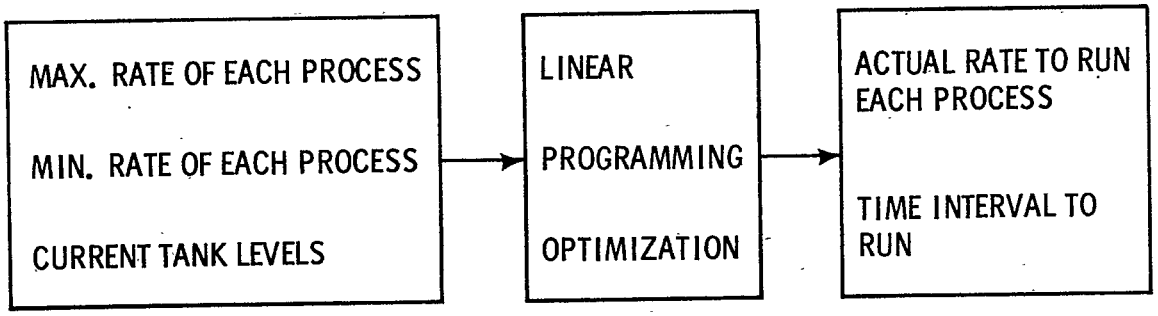
Additional linear programming equations are generated for each tank to represent the ending tank level minus the upper operating limit (if the difference is positive), and the lower operating limit minus the ending tank level (if the difference is positive). If one of these variables is positive, then a tank level is either above its upper operating limit or below its lower operating limit (but in no case can an absolute tank limit be violated). These variables are used in the objective function to give a penalty to violating an operating limit. As a result, the model will cause an operating limit to be violated only if the additional production obtained by doing so is sufficient to offset the penalty.

A process can have a minimum rate which is non-zero. This means that the process either runs between its non-zero minimum rate and its maximum rate, or it runs at exactly zero. This non-linearity is handled by introducing a variable for each process with a non-zero minimum rate to represent the processes' minimum rate times time minus the processes' actual production (if the difference is positive). These variables are given a very high penalty in the objective function. As a result, the model tries very hard to drive all of these variables to zero so each process will be operating within its allowed range of rates. If the variables cannot all be driven to zero, then the process which is furthest from its allowed minimum rate is shut down to zero rate and the linear programming problem is solved over again. In this way, the non-linearity is modeled by repeatedly solving linear problems. This will not necessarily find the globally optimum way to maximize production, but it has worked well and efficiently in practice.

The time variable in the linear programming problem is restricted to be between minimum and maximum time horizons. The minimum time horizon forces the rates to be calculated so that no process will have to shut down due to full or empty tanks in a shorter time. This provides some stability of operation. The maximum time horizon provides a planning horizon for calculation of rates. For example, if some tank is going to be filled because all of its output processes are currently down, the input process rates can be adjusted so the tank will just fill at the end of the time horizon. After that the linear program will reoptimize all rates for the following time interval.

When the linear program has determined the time interval to use, the model checks to see if the next event on the master events list is due to occur before the end of that time interval. If so, the rates determined by the linear program will be used only until that event occurs; then the new event (rate change) will be recorded and the linear program will re-optimize. If not, the rates determined by the linear program will be used for the time interval it determined. Figure 5 summarizes the information flow to and from the linear programming module.

FIGURE 5
LINEAR PROGRAMMING INFORMATION FLOW



MODELING IN FINAL ANALYSIS

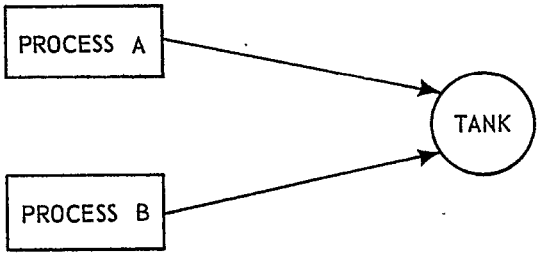
We have built a general model with features common to many process plants. If some particular feature of a plant is essential to a realistic representation but cannot be modeled with our general model, then a "custom built" simulation model will be necessary. However, we have found that the experienced model user can handle many "non-standard" situations. Following are some examples:

Process Priorities

Basically the objective of the optimization in final analysis is to maximize the total amount produced by all of the processes. Output processes (i.e., processes which do not feed any tanks), however, are given a higher priority since they normally represent finished products. In addition, users can change the priority given to any process by multiplying the default priority by a scale factor. This is useful for several different purposes. For example, in Figure 6:

FIGURE 6

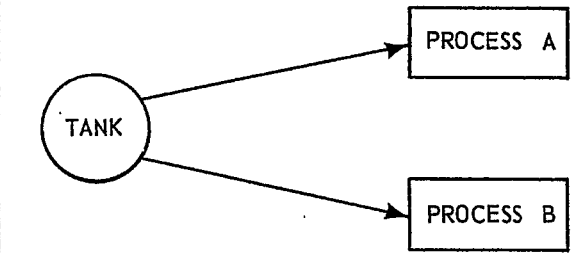
MULTIPLE PROCESSES FEEDING TANK



either Process A or B can feed the TANK. If the default priorities are not changed, the model will use one or both processes arbitrarily as needed to keep the TANK full. In many cases, it is preferred to use one of the processes, say A, to its full capability before Process B is allowed to run at all (e.g., if Process B is more expensive to run). This is easily accomplished by using a scale factor, say 2.0, to give Process A a higher priority than Process B. A similar example occurs when material in a tank can be used to make either of two final products - see Figure 7. If, say, Process A produces the more valuable product, then scaling its priority up by a factor of 2.0 will cause Process B to run only when Process A is using all the material it is currently capable of using.

FIGURE 7

TANK FEEDING MULTIPLE PROCESSES



Approximate Flow Split

Suppose the material in a tank can be used to make either Product A or Product B, and that normally it should produce these in a 25/75 split. This can be done as in Figure 8.

Suppose further that if either Process A or Process B is failed, then the other process should get all of the material. Failure of either Process A or B in Figure 8 causes all of the processes to shut down since the Dummy Tanks do not provide any buffer capacity and Process C cannot run without producing both of its outputs. Figure 8 can be modified as shown in Figure 9 to accomplish this.

If both Process A and Process B are running, then the material in Tank 1 can get to them either by going through or by going around Process C. Because the model attempts to maximize the production of processes, it will always choose to go through Process C and that will force the desired 25/75 split. If, however, one of the processes, say A, is failed, then the 25/75 split is not possible and Process C cannot run. In this case the model will allow as much material as can be used to go directly from Tank 1 to Process B.

OUTPUT REPORTS

Various reports, most of them optional, are produced for the final analysis phase:

- Input Data Log - This reports on the input data as read from cards.
- Master Events List - This report consists of the merged list of events for all processes from time zero to the specified simulation time.

FIGURE 8
FIXED FLOW SPLIT

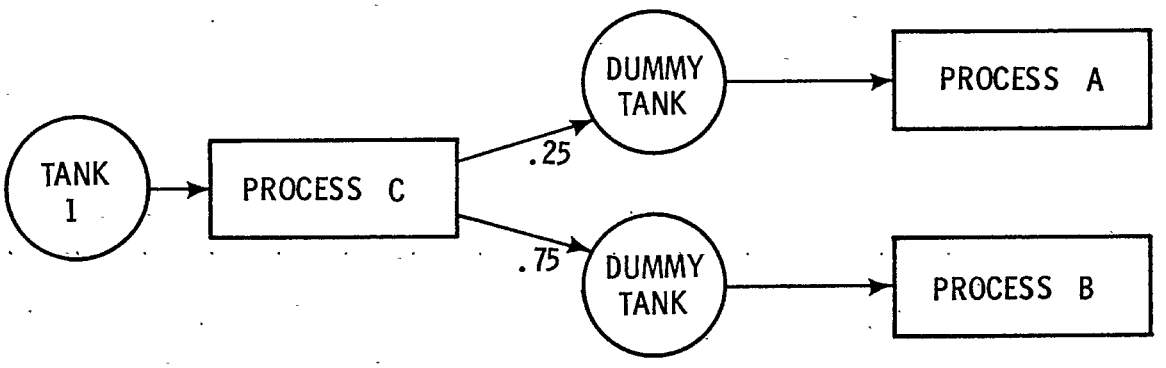
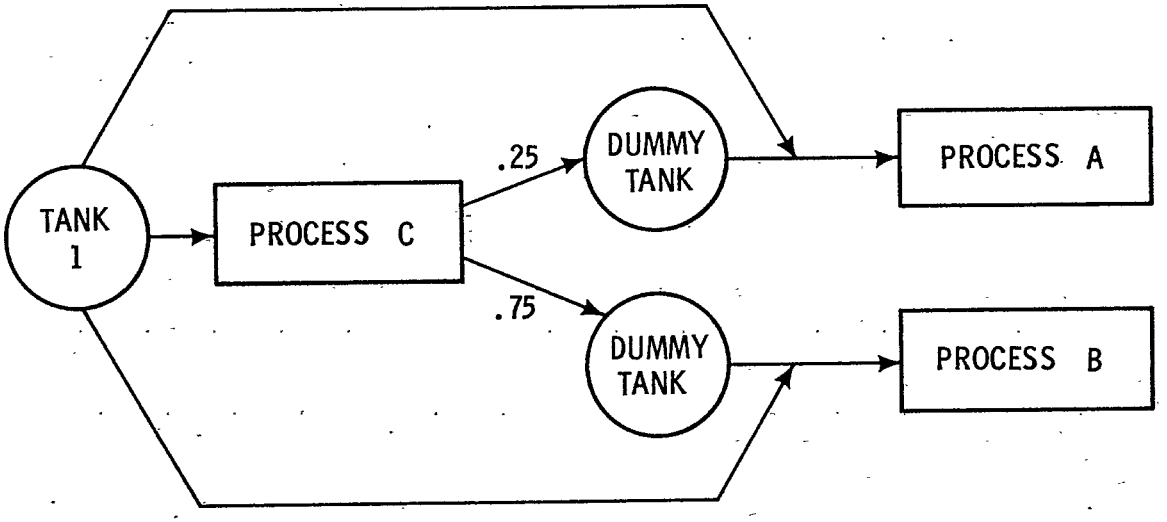


FIGURE 9
APPROXIMATE FLOW SPLIT



- Simulation trace - This report gives, for any specified time intervals, a chronological trace of the process rates and tank levels over time. These are the rates actually used rather than the rate capabilities as given in the master events list. This report is essential for complete understanding of the model dynamics. (We are now working on producing COM plots of these results.)
- Process Statistics - This report gives availability, production, and downtime statistics for each process.
- Tank Statistics - This report gives time spent above the upper operating limit and time spent below the lower operating limit, as well as the average tank level for each tank. This is very useful in deciding which tanks are larger than needed and which tanks are bottlenecks.
- Process Criticality Summary - This report gives the percent of time each process was running "unconstrained" (i.e., at its maximum current rate, even if that maximum is zero), running "throttled" (between maximum and minimum rates), and running "forced to zero" (by process/tank interactions). This is a very important report because processes which are bottlenecks are easily identified as those which are running unconstrained a large percent of the time.
- Process Rate and Tank Level Histograms - These reports give tables and plots of rates and levels using either relative or cumulative frequency.

IMPACT OF THE MODEL

The generalized model described in this paper is available within Du Pont as a proprietary Fortran program called "SAGE" ("Systems Availability Generalized Evaluator"). The SAGE program has been, and continues to be, heavily used for studies involving many different plants and departments. During a recent 6-month period, monthly usage of SAGE averaged 150 computer runs requiring 13 hours of Univac 1100/21 CPU-time. Individual run times vary from seconds for runs with short equipment analyses to about 1 hour for runs with equipment and final analyses of complex systems. The program is used primarily by Maintenance Engineers providing company-wide consulting services, but it is also used by engineers at plants who use Du Pont's Remote Job Entry System to access computers at Engineering Department headquarters.

ACKNOWLEDGEMENTS

The programming system described in this paper has been under development for over 10 years. Many members of Du Pont's Operations Research Group and Maintenance Engineering Group have contributed to its continued improvement over the years. Particular mention should be made of Paul A. Zitlau of the Operations Research Group, and Robert E. Bley of the Maintenance Engineering Group for their crucial efforts in designing, programming, and implementing the model.

REFERENCES

1. Wood, D. R., Muehl, E. J., and Lyon, A. E., "Determining Process Plant Reliability", in Chemical Engineering Progress, Vol. 70, No. 10, (October 1974).

2. Fox, R. J., and Zerbe, D. R., "Some Practical System Availability Calculations", in AIIE Transactions, Vol. 6, No. 3, (Sept. 1974).

3. Hadley, G., Linear Programming, Addison-Wesley Publishing Company, Reading, Massachusetts, 1962.