Simple Simulation

Richard C. Wilson

The University of Michigan
Ann Arbor, Michigan

Abstract

This paper describes an example of a network flow modelling approach for "quick and dirty" analysis of traffic dynamics in material handling systems. The example is based on a real installation. It serves to demonstrate potential for quickly comparing the design feasibility of alternate conveyor system layouts under dynamic loading before using specially programmed simulations.

INTRODUCTION

Material handling systems for manufacturing plants or distribution centers are increasingly costly and complex. The installed cost of a computer-controlled distribution center storage and handling system may exceed $5 million. The total (1977) sales of U.S. material handling products exceeds $5 1/2 billion, with more than 30% in conveyors and storage machines. Because most major installations are unique and specialized to the customer's projected needs, simulation is often used by both purchaser and vendor to assess the feasibility of a particular handling proposal.

Before programming a simulation of a new proposed material handling system, alternative system concepts must be developed. Only the most attractive of these can then be considered and selected for study by simulation. Time pressures and cost usually preclude simulation of all reasonable options. Consequently, the merits of systems to be simulated depend upon the experience and skills of the concept designer. The elapsed time from conception to installation of a complex handling system may be several years or more. Furthermore, systems are seldom identical. Learning gained from experiencing an installation from concept to installation may not be useful to the next generation of systems or applications. The process can be accelerated through simulation studies, but the inferences gained from simulation may be difficult to codify and translate into generally applicable knowledge.

A new unit load handling system is designed on the basis of uncertain forecasts of future demands, product mixes, and supplier or customer behavior. Performance requirements are often given in terms of required throughput with a "typical" mix and a maximum or peak capacity. Explicit definitions of the characteristics of "peak," as well as distributions of failures, repair times, or arrival intervals are unusual. Hence, design performance must often await installation and startup for verification. Performance sometimes must be demonstrated using customer personnel under local conditions perhaps unknown to the system vendor at the time of contracting.

The implications are fundamental to the industry's design practices and use of system simulation. Since simulation validation is often limited to qualitative assessment, it is primarily useful for comparison of the key alternatives, or for verification of design feasibility under clearly stated quantitative modelling assumptions. A conveyor hardware designer may calculate the maximum throughput capacities for each device in the system. If he is unable to assure himself that a particular segment of his system design concept will not be a critical bottleneck, he often will redesign the uncertainty out of his concept. Often a conveyor engineer's calculations involve a second by second timing analysis of the position of case goods as they move through a series of sensors and diverters or merges. Inventory accumulation space and flow inventory are calculated by repeated use of Little's formula $L=\lambda W$ in every conceivable interpretation. Typically these analyses are time-consuming, detailed, and prone to numerical errors typical of paper and pencil simulation. The designer is motivated to provide an extra margin of capacity to minimize the need for such detailed hand analyses. At the same time, unnecessary capacity margins increase the cost of the system and may make the engineering price quotation non-competitive.

Simple Simulation (continued)

These observations lead to the conclusion that there is a need for a general modelling capability at the concept stage (at least in the material handling field) which would serve as a precursor to simulation. Such a tool would need to satisfy a number of requirements:

- be able to model the flow and queueing characteristics which arise from non-stationary loading and scheduling of activities.

- require no programming or debugging skills of the user. Terminology and model construction must fit the methods used by the project engineer in pursuit of his normal tasks.

- permit stochastic modelling, e.g. arrivals, failure times, and explicit schedule and flow control rules.

- permit application to a wide range of arbitrary configurations of a variety of handling components.

- permit easy tradeoff analyses of the interplay among changing the time dynamics of the loading and scheduling, the consequences of limited storage and queueing space, the choice of movement velocities and flow rate capacities, and the impact of overflow due to blocking.

- should be consistent so that one modelling scheme applies for any choice of time unit. Thus, it should be possible to use the same model and easily compare and verify the results of steady-state analyses, day by day loading changes, or second by second dynamics.

- provide optimization (normative) capability to reliably determine the best performance which can be obtained from a particular design. If this performance is barely adequate to meet the system requirements at the concept phase of aggregated modelling, simulation of a more detailed system model with additional constraints will be unnecessary.

- must be easy to translate the user's problem from his terms to that required by the model.

- must be easy to explain and understand the model so that it has credibility with non-simulation specialists, and inspires confidence in the user.

- must be able to compute results at low cost per trial run and responsive enough for interactive time sharing utilization.

- must be compact enough to run on small computer systems.

- provide analytic performance characteristics which are sufficiently valid and detailed to provide acceptable comparisons among alternate system configurations.

- should be modular to allow adding elements and capability.

This utopian wish list is beyond the reach of any single tool available for today's material handling engineer. It is mind boggling however to reflect on the amount of effort and dollars expended annually in the simulation evaluation of industrial material handling systems. Most require the development of the simulation model de novo. For operating installations, a simulation model can be used repetitively for study of protocols for improving operations or adapting to changing conditions. Examples of general purpose problem-oriented manufacturing simulation packages are found in [1], [2], and [3]. Only a few analytic schemes seem to be available for checking conceptual designs before simulation.

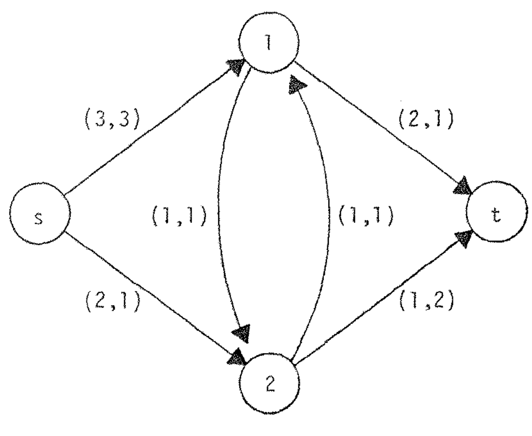
Recent developments in queueing models of stochastic networks have broadened their potential as aggregate models of material handling flow systems [4]. They have been applied to estimation of the steady-state throughput capacity of flexible manufacturing systems, where an infinite queue capacity assumption is tenable. The negative exponential service time assumption however seems incompatible with the deterministic and/or correlated flow processes in many automated handling systems.

Graphical techniques have been advocated by Newell [5] as important for engineering evaluation of flow characteristics in traffic design, especially when the effects of non-stationary processes must be treated. We have found this approach to be useful in evaluating the non-stationary behavior of a series of server activities under deterministic assumptions of input rates and processing times. However, the graphical method becomes extremely complex and difficult to use when multiple products, recirculation flow, or transit time delays must be considered. Furthermore, methods which can be used as characterizations of specific material handling devices must be invented for each application.

Mathematical programming offers the potential for development of mathematical models of material handling flow with considerable scope. Briefly, a material handling system is represented in a network flow model by a set of nodes connected by directed arcs. An arc represents the potential for passage of unit loads from one point to another. The rate of loads which flow across the arc may be constrained by upper and lower bounds. A cost may be assigned to each item which flows across an arc. Each arc has a travel time for an item. A simple dynamic network is shown in

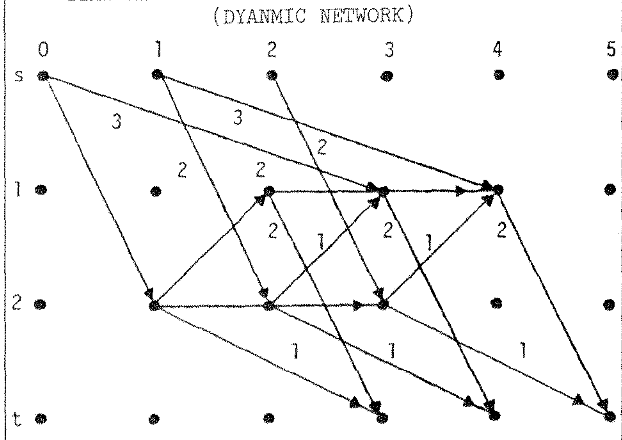
Figure 1. In the figure, a pair (F,Δ) is associated with each arc. F represents the maximum rate of flow items over the arc, and Δ represents the travel time.

Figure 1
A SIMPLE DYNAMIC NETWORK



The dynamic network can be transformed into an equivalent time-expanded static network with time intervals represented horizontally and the space nodes of the original network represented vertically (Figure 2). A network flow algorithm can be used to find the maximum value of the flow of items from node s to node t over the arcs of the time-expanded static network in a specified number of time periods. Using such networks, we are able to model a large number of conveyorized material handling configurations. We have developed a software system which we call DYNAFLO to explore this potential. Details of the system software are given in [6].

Figure 2
TIME-EXPANDED STATIC NETWORK IN FIGURE 1
(DYANMIC NETWORK)

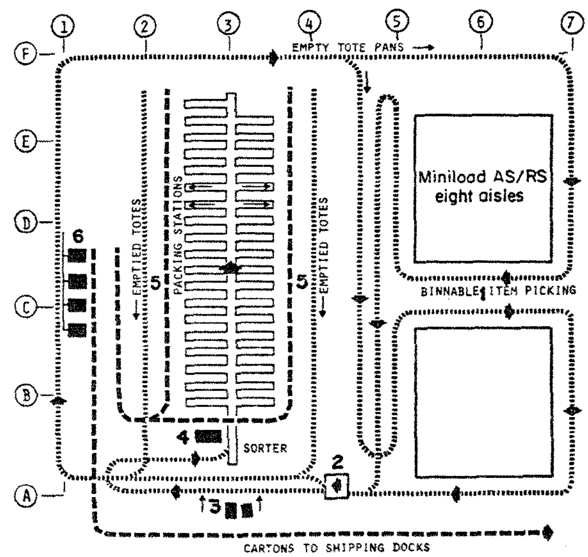


EXAMPLE PROBLEM

To demonstrate the potential of simple and approximate mathematical models for concept analysis, we use a disguised example of a material handling system in a large distribution center. The system is to have a guaranteed capacity to pick and individually pack and ship 1020 randomly demanded items in each hour of an eight-hour shift. Items are of two categories: 45000 items of type A which are less than 200 cubic inches and 36 inches long. The A items are stored in a 16-aisle mini-load stacker system (AS/RS). The B items are stored in containers on shelving.

Figure 3 is a floor plan layout of the A item picking area, and the packing for A and B items.

Figure 3
FLOOR PLAN LAYOUT OF ORIGINAL CONVEYOR SYSTEM
(BAYS ARE 40'×40')



The concept of the system is to minimize the picking time on the A items by bringing the items to the picker at (1), and to pack both A and B items in the same pack areas at (5). Empty tote pans either 12"×16" or 36"×16" are delivered continuously to the mini-load AS/RS picker on moving conveyors. The picker selects the desired item from the AS/RS bin, and drops it into an empty tote, together with an order tag. At (2), all full totes are automatically separated from the empties, which continue to recirculate on the conveyor to (6) and back to the AS/RS pick area. Full totes are conveyed to a sort station (4). Here, an operator reads the part order tag, and by pushbutton, keys in the packing lane which is scheduled to pack that order in the next hour. Full totes accumulate

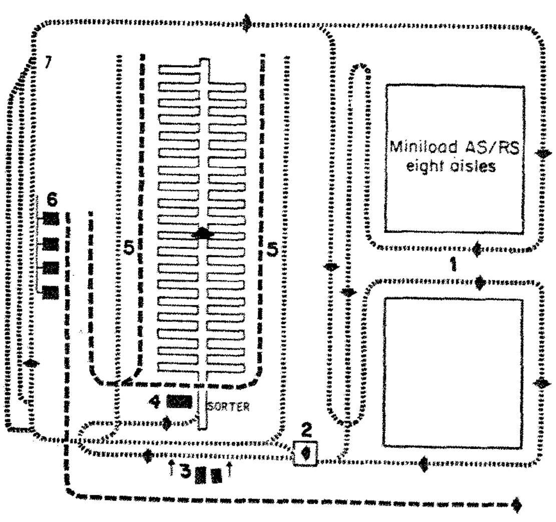
Simple Simulation (continued)

during each hour on one of the two accumulation lanes feeding each of the several dozen packaging stations. At the start of each hour, each packing station removes items from the lane of totes pre-scheduled for one hour of packing work. As the items are packaged in outgoing cartons and dispatched to the shipping dock for pickup by UPS or parcel post, the emptied totes are returned to the system at 5 for merging and recirculation. At 3, full totes from the pick trucks selecting B items from shelf storage are placed on the conveyor line of full totes. Empty totes for picking B items are obtained from the conveyors at 6.

In designing this system, the vendor needed to assure himself that the proposed concept would achieve a pick/pack rate of 1020 items per hour. Calculations were needed to determine that empty totes would be available to the mini-load and shelf picking operations as needed. Furthermore, adequate but not excessive lengths of accumulation conveyors must be provided where needed. The number and mix of totes required to maintain this throughput flow and accumulation must be estimated, and potential blockage or starvation of operations anticipated.

There was initial concern about possible congestion where the emptied totes from the two sets of packing stations merged with the recirculating empties from the mini-load pick conveyors through 2. An alternate design was proposed, Figure 4, which added over several hundred additional feet of tote accumulation conveyor before the new merge point at 7.

Figure 4
FLOOR PLAN LAYOUT OF REVISED CONVEYOR SYSTEM



Simulation was proposed to establish the need for this additional expenditure. We show that the use of a simple modelling technique such as DYNAFLO could have resolved the need for this expenditure.

CONVERSION TO DYNAFLO NETWORK MODEL

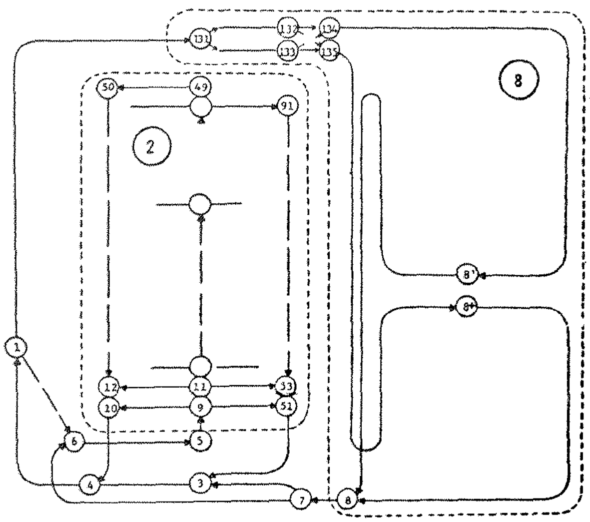
Analysis of a material handling system requires selection of the necessary level of detail. In simulation, unnecessary detail can produce large complex models, which are costly to program, debug, and execute, and often difficult to interpret. In DYNAFLO network modelling, levels of detail are limited to the set of available program modules, and program debugging is eliminated. However, one to one congruence between conveyor segments of the real system and the network models can produce a high number of space-nodes in the network. Therefore segments of the real system should be aggregated for modelling purposes. Similarly, the time step (sample interval) should be chosen to be as large as possible to minimize the number of item intervals required, but still to be compatible with the time dynamics of the system under investigation.

The following steps are necessary to develop a DYNAFLO model, given a physical layout of the handling system:

- draw a scaled layout network of the flow of items in the system.
- compute the lengths of each of the conveyor devices in the system.
- wherever consistent with the purposes of the analysis, aggregate clusters of adjacent conveyor devices into a single device arc with length and capacity equivalent to the original cluster.
- select a basic time-slice duration for the analysis and the number of time-slices needed in the dynamic model run so that the time varying behavior will be captured.
- compute the integer number of time-slices for a unit load to traverse a conveying device, given its velocity and length. Compute the bottleneck flow capacity per time-slice, and the total storage capacity for each of the aggregated devices.
- generate specific bounds on flow and storage for aggregated arcs which reflect exogenous schedules and flow control.
- select mode of analysis: minimum cost/maximum flow; wraparound or not; and control costs on run.

Figure 5 is a scaled network of the item flow in the system layout of Figure 3.

Figure 5
NETWORK REPRESENTATION OF CONVEYOR LAYOUT IN FIGURE 3
(Multiple nodes omitted for clarity.) Nodes in dotted lines are condensed into two new nodes: the packing area becomes new node (2); the AS/RS pick area becomes new node (8).



We note that already some sequential conveyor devices such as live roller and zero pressure accumulation have been combined into one arc. Where the distinction is important to the purposes of the study, additional nodes and arcs need to be retained. Table I identifies each of these arcs, by type of conveyor, its velocity, and length.

Table I
DEVICE TYPES FOR NETWORK OF FIGURE 5

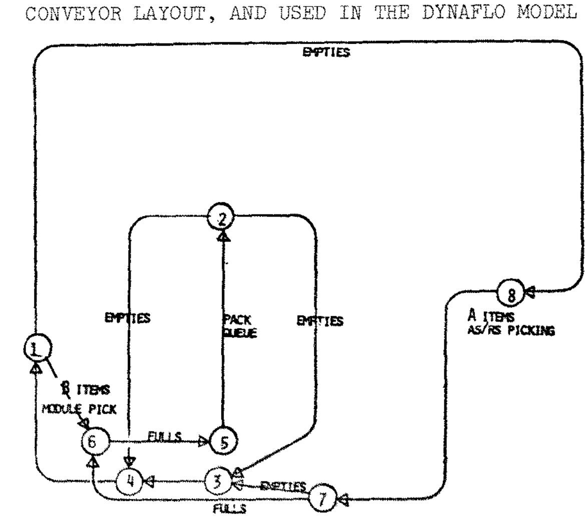
Arc Number	Device Type	Velocity (FPM)	Length(Feet)
1,6	Module picking delay	--	--
1,131	ACCUM	100	240
3,4	ACCUM	60	40
4,1	ACCUM	90	80
5,9 9,11 11,13 etc. 49	BELT (Chain Conveyor)	120	190 each
6,5	ACCUM	80	40
7,3	ACCUM	60	40
7,6	ACCUM	90	160
8,7	BELT	90	3
8',8	ACCUM	60	320
8+,8	ACCUM	60	220
9,10 9,51 49,50	QUEUE (Gravity)	--	50 each
10,4	ACCUM	60	30
50,10 91,51	BELT	100	120 each
51,3	ACCUM	60	80
131,132	ACCUM	55	40
131,133	ACCUM	55	40
132,133, 134,135	DIVERTER	--	--
134,8'	BELT	60	200
135,8+	BELT	60	320

We note that if we assume that each of the packing accumulations' arcs (9,10), (9,51), (11,12), etc. behave identically, an aggregate representation of the packing stations might be adequate for analyzing the queueing issues. A similar assumption about the two AS/RS mini-load picking conveyors enables us to aggregate these two conveyor lines also. We thereby greatly reduce the size of the network needed by replacing the nodes clustered within the dotted lines by new nodes. Figure 6 and Table II show the aggregated network.

The node-arc configuration for the packing area in the aggregated network represents the dual gravity storage conveyors feeding each packing station. During each hour ("flight"), the packer removes totes from a full gravity conveyor, while the other (empty) conveyor accumulates full totes dispatched from picking activities at a uniform rate. Thus, the gravity conveyor (5,2) is full at the start of the hour ahd is emptied at a dynamic rate determined by packing methods during the hour. The other gravity conveyor feeding the packer station must be full at the end of the hour. The packer's rate of release of empty totes into the conveyor system is fixed by the flow bounds on arcs (2,4) and (2,3). We assume the release schedules are identical for each half of the packing stations.

The choice of time-slice duration requires a compromise between exact travel times and network computation time. A time-slice of 0.6 minutes would provide a good least common multiple of all travel times, and require 100 time-slices to simulate an hour of operation. For this study we somewhat arbitrarily select a time-slice of 2

Figure 6
SIMPLIFIED NETWORK REPRESENTING THE ORIGINAL CONVEYOR LAYOUT, AND USED IN THE DYNAFLO MODEL



minutes, requiring 30 time-slices per hour and the coarse device travel times in time-slices in Table II.

Table II
DEVICE TYPES FOR AGGREGATED NETWORK OF FIGURE 6

Arc Number	Arc Nodes	Device Type	v Velocity (FPM)	d Length (Feet)	Travel Time (Min.)	Travel Time (Time-slices of 2 min.)	Flow Capacity (Units per Time-Slice)	Storage Capacity (Units)
1	1,6	(Module Picking Delay) QUEUE	--	--	0	0	∞	∞
2	1,8	BELT	60	600	10 (for travel time) (total length = 1200)	5	90	--
3	2,3	ACCUM	100	120	1.2	1	75	90
4	2,4	ACCUM	100	120	1.2	1	75	90
5	3,4	ACCUM	60	40	.67	0	45	30
6	4,1	ACCUM	90	80	.88	0	67	60
7	5,2	QUEUE	--	(600)	--	--	--	∞
8	6,5	ACCUM	80	40	.5	0	60	30
9	7,3	ACCUM	90	160	1.77	1	67	120
10	7,6	ACCUM	60	40	.67	1	45	30
11	8,7	QBELT	60	270	4.5	2	45	200
				Σ2070				

For the studies, we assume a mix of 3 foot and 1 foot long totes in the ratio of 1:5. Hence on the average, queued totes occupy $\ell=(5*1+3)/6=$ 1 1/3 feet. In general, the flow capacity is limited by v/ℓ, and the storage by d/ℓ. On zero pressure accumulation conveyors, we assume that moving totes will have a gap of at least one tote length between them. Hence the maximum flow capacity of an accumulation conveyor is half the storage capacity per time-slice. The resulting flow and storage capacities are given in Table II. The values will be altered as described later according to the specific requirements of each of the investigative analyses. In addition, since we assume that the picking and packing schedule for each hour is to be identical, a "wraparound" model is specified, where the terminal state at the end of each hour is the same as the initial state at the beginning of each hour.

THE EXPERIMENTS

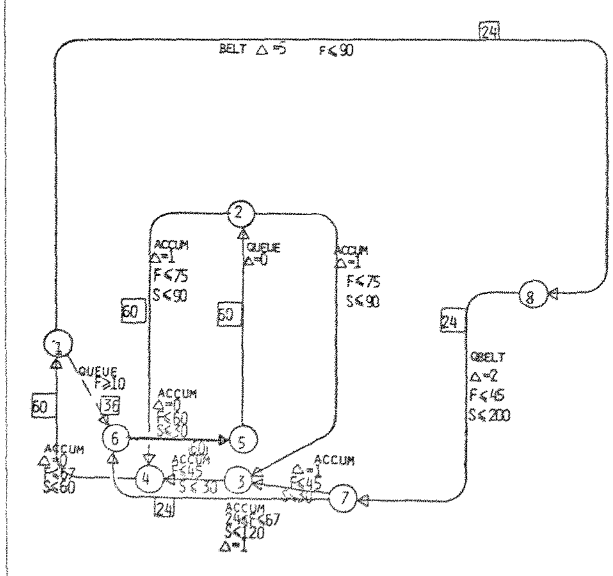
A series of analyses are conducted, with increasing constraints on system behavior. We explain the selection of parameters for each of these runs.

Run 1

The first run is a simple steady-state (one-period) run to determine the maximum capacity of the system. A large negative cost is placed on flow out of the queue arc (5,2) in order to drive the model to find the maximum capacity packing rate. A lower bound of 10 totes per time-slice is placed on the flow of full totes out of the module picking queue arc (1,6), and of 24 full totes per time-slice out of the AS/RS pick arc (7,5). Since there are 30 time-slices per hour, these bounds imply a total pick rate of 1020 totes per hour.

Run 1 indicates that the maximum capacity of the system is 1800 totes per hour, which is easily seen to be the capacity of 60 totes per time-slice (30 per minute) on conveyor (6,5) into the sorter. The flows are given in Figure 7. A total of 252 totes are required for "pipeline" inventory.

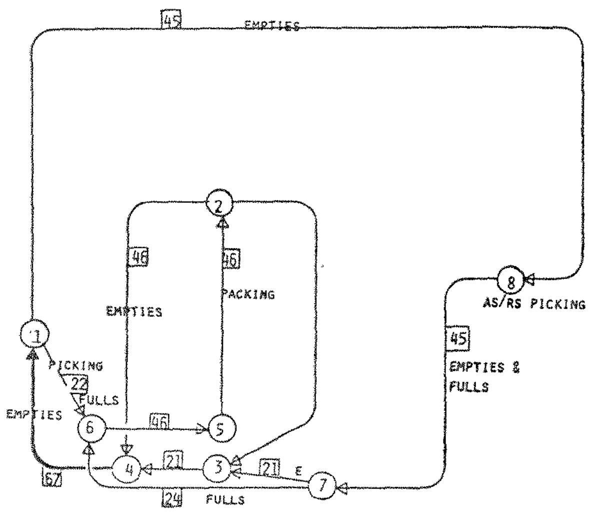
Figure 7
PARAMETER VALUES FOR STEADY-STATE CAPACITY ANALYSIS OF RUN #1. NUMBERS IN □ ARE MAXIMUM FLOW IN TOTES PER TIME-SLICE. HEAVY ARC (6,5) IS THE BOTTLENECK.



Run 2

The maximum capacity of 1800 totes per hour is achieved in Run 1 by flow required by full totes. Thus no excess empties circulate through the system. Suppose we want to assess the system steady-state capability to pick/pack at the required rate of 1020 totes per hour, while circulating an additional 780 empty totes per hour to use the flow capacity of 1800 totes per hour. If we add a lower bound to the flow of empties on the accumulation conveyor (7,3) of (780*2)/6=26 totes per time-slice (TPS), we discover that the required total flow of 26+24 exceeds the 45 TPS capacity of accumulation (8,7) and is therefore infeasible. We reduce the required flow rate of empties to 21 TPS on (7,3). The resulting maximum feasible flows are given in Figure 8.

Figure 8
FLOW VALUES OBTAINED FROM STEADY-STATE CAPACITY ANALYSIS WITH REQUIRED RATE OF 21 EMPTIES PER TIME-SLICE RECIRCULATION. BOXED NUMBERS ARE MAXIMUM FLOW IN TOTES PER TIME-SLICE. HEAVY ARC (4,1) IS THE BOTTLENECK.



We note that the consequence of forcing the circulation of empty totes saturates the capacity of empty accumulation conveyor (4,1). The rate at which empty totes must be recirculated limits the rate of use by the module picking activity (1,6). Consequently, the maximum achievable packing rate is reduced to 1380 totes per hour, because of the need to circulate empty totes! We note that the affect of intermediate rates of empty recirculation on maximum packing can be easily determined by varying the model parameters. For example, with a required empty tote recirculation of 300 per hour, the maximum packing rate is 1710 totes per hour.

Run 3

The preceding analyses assume that rates of both the packing and picking operations are constant. In reality of course they both fluctuate. The picking is executed continuously throughout the shift by a dozen or more operators. It is reasonable therefore to assume a constant mean rate of placement of inventory into empty totes by the pickers. The packers however operate on an hourly scheduled work load. Generally at the beginning of each hour most packers remove the items from a number of totes waiting on the gravity conveyors. The items are placed on the packing bench, and the emptied totes are placed on the empty return conveyors. Thus, a surge of empties of uncertain magnitude is imposed on the system at the start of each hour. The affects of this hourly transient in the mean release rate need to be assessed.

In order to capture the transient mean packing rate, we set the output of the packing stations into the takeaway conveyors (2,4) and (2,3) at 39 totes per minute for the first 20 minutes of each hour, and 6 totes per minute for the remaining forty minutes. Thus the total output of empties is at least 1020 per hour, corresponding to the minimum pick rate. We also initially set the accumulation capacities of all devices to be unbounded, thus avoiding infeasibility due to blocking or starvation in this trial run. With required empty recirculation of 300 totes per hour and the dynamic packing rate, the pack rate of 1710 per hour is the maximum feasible. However, we note that the dynamic packing rate requires at least 965 totes in the system for feasibility compared to 329 for the constant pack assumption.

Run 4

For this run, we reduce the storage capacities of all devices to the bounds appropriate to their physical size. Otherwise the model is identical to Run 2. The analyses indicate the model is now infeasible. Examination of the flows discloses the infeasibility occurs during the period 16 to 20 minutes after the scheduled hour. In Figure 9, we see that the accumulation conveyor (4,1) is the bottleneck throughout the hour. During these critical minutes it is unable to carry away the emptied totes. As a consequence the packers' discharge rate is blocked.

Run 5

The concern for the potential congestion from the preceding run resulted in a modified design on the output of the packing stations. The modified design was intended to provide additional queue space before the merge of empties, and to move the merge point beyond the point at which the module pickers picked up the empties for their picking tours. Figure 10 shows the network for the revised design of Figure 4 and Table III lists the device types and parameter values for the revision.

Simple Simulation (continued)

With the unbounded storage capacities, the new configuration permits a maximum pack rate of 1800 totes per hour. Through this run, a total of 472 totes are circulating in the system.

Figure 9
FLOWS DURING TIME-SLICE 10 FOR TRANSIENT RUN #4 WITH ACCUMULATION CONSTRAINTS. HEAVY LINES INDICATE FULLY LOADED FLOW AND STORAGE CAPACITIES, SHOWING BLOCKING OF PACKERS' EMPTY TOTE DISCHARGE FROM NODE 2. NUMBERS ON ENDS ON EACH ARC ARE NUMBERS OF TOTES INTO OR OUT OF THE ARC DURING TIME-SLICE 10.

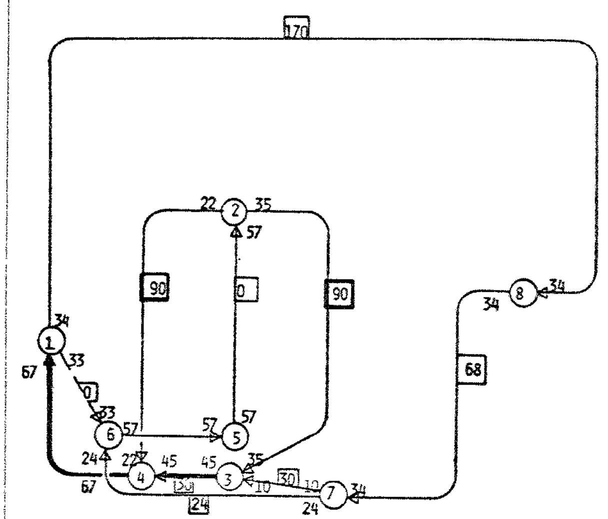


Figure 10
NODE-ARC NETWORK FOR REVISED HANDLING LAYOUT. Δ IS THE TRAVEL TIME IN TIME-SLICES, F IS THE FLOW BOUND IN TOTES-PER-TIME-SLICE.

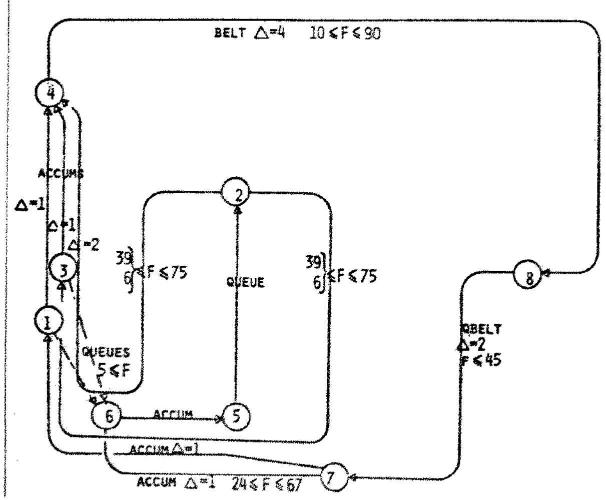


Table III
DEVICE TYPES FOR REVISED AGGREGATED NETWORK OF FIGURE 10

Arc Number	Arc Nodes	Device Type	v Velocity (FPM)	d Length (Feet)	Travel Time (Min.)	Travel Time (Time-slices of 2 min.)	Flow Capacity (Units per Time-Slice)	Storage Capacity (Units)
1	1,6	QUEUE	--	--	--	--	∞	∞
2	1,4	ACCUM	90	120	1.33	1	67	90
3	2,3	ACCUM	100	180	1.8	1	75	135
4	2,4	ACCUM	100	320	3.2	2	75	240
5	3,4	ACCUM	100	120	1.2	1	75	90
6	3,6	QUEUE	--	--	--	--	∞	∞
7	4,8	BELT	60	440	7.33	4	90	--
8	5,2	QUEUE	--	(600)	--	--	--	∞
9	6,5	ACCUM	80	40	.5	0	60	30
10	7,1	ACCUM	60	160	1.5	1	45	120
11	7,6	ACCUM	90	160	1.777	1	67	120
12	8,7	QBELT	60	270	4.5	2	45	200
				Σ2410				

Run 6 (Revised Design with Storage Constraints)

Run 5 is repeated except with physical storage limits imposed on the system. The result is a feasible pack rate of 1800 totes per hour. Under the same conditions the original design was unable to achieve the specified throughput of 1020 totes per hour!

We see that by increasing the total length of conveyor in the system from 2070 to 2410 feet, and by relocating the merge point of empties, a congested system is converted to one with substantial excess capacity. Figure 11 shows the flows in period 10 for this run. Comparison with Figure 9 shows how the congestion has been reduced. The number of totes in this system is 472.

Figure 11
FLOWS DURING PERIOD 10 FOR TRANSIENT RUN #6 WITH ACCUMULATION CONSTRAINTS ON REVISED LAYOUT. HEAVY LINES INDICATE FULLY LOADED FLOWS. NOTE ABSENCE OF CONGESTION ON ARCS (2,3) AND (2,4) ON DISCHARGE SIDE OF PACKING. NUMBERS ON ENDS OF ARC ARE NUMBER OF TOTES INTO OR OUT OF THE ARC DURING TIME-SLICE 10.

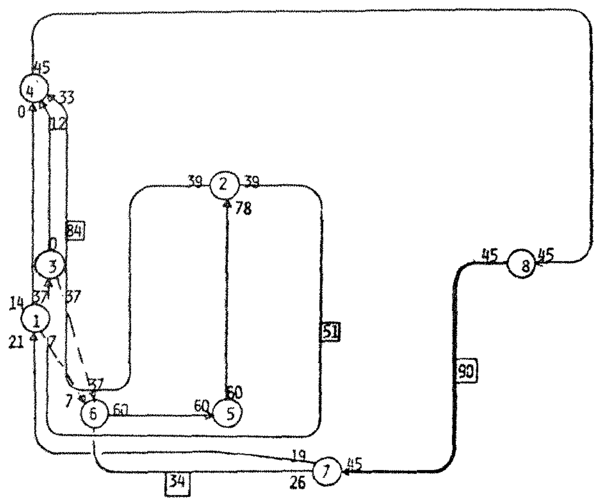
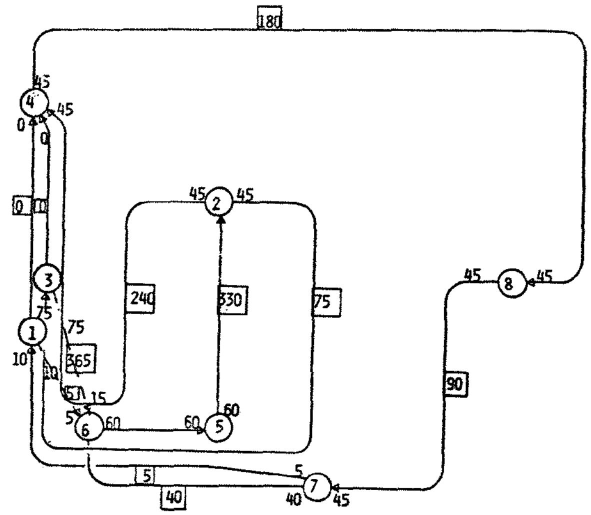


Figure 12
FLOWS DURING PERIOD 10 FOR TRANSIENT RUN #7 OF REVISED LAYOUT WITH ACCUMULATION CONSTRAINTS, AND INVENTORY OF 1020 TOTES IN THE PACKERS' LANES AT START OF EACH HOUR. NUMBERS IN BOXES ARE TOTES ON DEVICE AT END OF TIME-SLICE. NUMBERS ON ENDS OF ARCS ARE FLOW IN OR FLOW OUT DURING THE PERIOD.



Run 7

All of the preceding runs assumed that full totes are being delivered to the packers during the hour in which they perform the packing. In actual operation, the plan was to initiate the hourly packing load with an accumulation of one hour of full totes waiting on one gravity conveyor to be emptied. Thus, each hour should start with 1020 full totes in the queue arc (5,2). This is modelled by placing a lower bound of 1020 on the storage flow on arc (5,2) for only the first time-slice in each hour. The flow results of this operational change are shown in Figure 12. We note that the revised system is still feasible and can pack 1800 totes per hour. However, the new schedule requires at least 1330 totes in the system at any time.

SUMMARY

Our example shows that simple simulation using DYNAFLO can provide comparative evaluation of alternate concepts for complex configurations of material handling systems. Thus, it can eliminate detailed simulation of many inadequate layouts. The methodology is especially powerful in testing bounds on system flow capacity. Since the flows are the maximum possible under the given assumptions, additional complications (e.g., multiple products, stochastic loadings, etc.) imply additional constraints on the model, rather than relaxing existing constraints. Obviously, the maximum flows of more constrained models cannot be greater than those of the less constrained DYNAFLO model. A total Amdahl 470/V7 CPU time of 27.19 seconds was required for all runs, thus providing interactive response time for each run.

The dynamic network modelling technique is useful as a method for "quick and dirty" evaluation. Furthermore, the simplicity of the underlying models makes the results creditable and easy to understand. We believe that exploration of system configurations using network models may lead to general design principles which are obscured by more complex models. The approach however is also limited by its simplicity. It does require arbitrary discretization in time with consequent imprecise specification of flow capacities, and delay times. Furthermore, small time-slice models run over many time-slices may generate large models which, like simulation, can consume extensive CPU time. The present capability is limited to essentially one commodity systems amenable to pure network representation. If precision is required, detailed simulation can be used to confirm simple simulation implications.

We are currently investigating methods of modelling more complex handling system components, while retaining the conceptual simplicity of DYNAFLO. In addition, we are planning further evaluations of the system by participation at the conceptualization stage of proposed handling systems.

ACKNOWLEDGEMENT

The work described in this report was partially supported under NSF Grant APR76-16709.

REFERENCES

1. Heimburger, Donald A. "A Case Goods Conveyor Simulator," Proceedings of the 1973 Winter Simulation Conference (January, 1973), pp. 112-117.

2. Mitome, K., S. Tsuhara, S. Seki, and K. Isoda. "MAFLOS--A Generalized Manufacturing System Simulation," Proceedings of the 1973 Winter Simulation Conference (January, 1973).

3. Roggenbuck, R. A., and N. W. Hopwood, Jr. "Automated Model Building," Proceedings of the Tenth Annual Simulation Symposium (March, 1977), pp. 179-196.

4. Solberg, J. J. A User's Guide to CAN-Q, Report No. 11, Optimal Planning of Computerized Manufacturing Systems, School of Industrial Engineering, Purdue University, West Lafayette, Indiana (December, 1977).

5. Newell, G. F. Application of Queueing Theory, London: Chapman and Hall (1971).

6. Maxwell, W. L., and R. C. Wilson. Analyses of Dynamic Material Handling Systems by Network Flow, Working Paper No. 4, Manufacturing Flow Research Project, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan (February, 1979).

FROM THE EDITOR'S DESK

(continued from page 32)

offer my sympathy. Yet, I feel that we would all admit that you do not throw good money after bad. If ADA is more efficient than COBOL, should it not be used? All of us as taxpayers support government-contractors and if we can reduce expenditures by using more efficient programming, especially execution time, should we still oppose ADA?

However, I must note that at this time I do not have any reasonable comparison between ADA and COBOL. Maybe, before we attack the introduction of another computer language, we should look at the facts. Only last week I found that a close friend, Sue Solomon and former chairperson of SIGSIM, has a package for statistical validation of simulation data. Well, I have had one around for almost ten years. I have over the years found about three others that I consider worthwhile. Must we continue to reinvent the wheel in computing?

Part of our problem is the lack of communication. Part is also the feeling that what we have produced is not 'fine' enough to make public and exposing ourselves to possible criticism from our fellow practioners. It is difficult for many to change their thinking from programing in one language to programing in another. For some of us who feel exceedingly comfortable with a powerful language, we do not wish to start again with a new language.

Let us wait until we have full information. And let us approach this conflict reasonably. Maybe the best solution, if ADA does prove itself superior, is to phase the language in. We should not make obsolete existing programmers who do not wish to encumber themselves with a new language, but maybe the colleges should start teaching ADA to future programmers.

I cannot help, because of industry training, to look at cost-effectiveness. If ADA does prove itself, then in the best American tradition of the profit motive, we should go along with this new language. Yet, we must consider the human element.

Harold Joseph Highland

Harold Joseph Highland