A SIMULATOR FOR DESIGNING HIGH-RISE WAREHOUSE SYSTEMS

Kailash M. Bafna, Ph.D.

Assistant Professor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Abstract

This paper presents a simulator to evaluate alternative designs of high-rise warehouse systems using stacker cranes to permit system cost minimization while satisfying design constraints and specifications. There are no warehouse size restraints on its use. Furthermore it can consider single or multiple stacker crane systems. Resolution of problems associated with finding combinations of storage height and aisle length in combination with crane vertical and horizontal travel speeds is provided. Hardware and operating variables considered are more extensive than those found in earlier simulations of similar systems.

Introduction

Automatic stacker cranes are still in their infancy, being only about a decade old. However, in this short period of time, they have assumed an increasingly important role in the field of materials handling, especially in warehousing. It is predicted that the rate of growth of this industry over the next few years will approach an exponential rate and form a significant proportion of the materials handling industry.

Stacker installations involve high investment. In 1965, the average system cost $150,000; today it costs over $1 million; in 1975, it may range from $4 to $5 million.[1] At the time management makes a commitment for a stacker installation (which may include new racks in an existing warehouse, or even building a new warehouse), risks are involved in forecasting operating demands and capacities and it is not absolutely certain whether the desired rate of return on investment will be achieved.

Management makes an investment for which it expects a certain return to be realized, and

[1] "Up-to-Date Report: How Stacker Cranes Stack Up", Modern Manufacturing, June 1970, p. 56.

would like to minimize their risk in realizing such returns. Such a preview can only be attained if the engineers (who may have no prior experience with such an installation) have simple analytical and quantitative tools available by which they can study the characteristics of alternative systems. Such techniques may also be used for estimating realizable values of the principal parameters of a high-rise warehouse using stacker cranes prior to final design and installation.

This research has approached this objective by developing a discrete-event simulator, hereafter referred to as BASS (Bafna's Stacker Simulator). The simulator has been tested for different throughput requirements.

This paper very briefly describes the cost representation (the cost model for the various components of the warehouse is one of the most elaborate cost representations) of a high-rise warehouse. This is followed by a description of BASS. The paper concludes with a discussion and analysis of some sample runs using the simulator.

Principal Assumptions

Some of the major assumptions made in the analysis are:

1. The physical size, maximum height and weight of the unit-load to be stored in the system is known.

2. The system handles only one size of unit-loads.

3. An estimate of the total storage slot requirements in the warehouse is known.

4. The required throughput for the system (number of unit-loads to be handled--both stored and retrieved--by the proposed system) is known.

The Cost Model

A high-rise warehouse using stacker cranes consists of several elements such as land, building, racks, stacker cranes, transfer cars, fire protection equipment, peripheral equipment (such as conveyors, fork lift trucks, etc.), inventory control equipment, lighting and heating (if necessary), etc., a cost being associated with each. The costs for a few of these elements will remain somewhat constant irrespective of the warehouse design. However, the costs of the other elements will vary with the warehouse and stacker crane parameters. Hence, in order to select between alternate warehouse designs, it will suffice to study the costs that vary with variations in the warehouse and crane parameters.

The design variables used in the analysis are:

1. Number of slots along the height of rack.

2. Pairs of rows of racks in the system (also equal to the number of aisles).

3. Number of stacker cranes in the system.

4. Horizontal speed of stacker cranes.

5. Vertical speed of stacker cranes.

The variable costs which have been considered in the cost model are described below.

1. Cost of Floor Space. This is considered to be the cost of (a) area occupied by

racks, (b) area occupied by aisles, and (c) area for related warehouse services.

2. Cost of Building. This is the sum of three elemental costs:

1. Foundation and floor on which the racks are mounted.

2. Roof of warehouse.

3. Perimeter walls.

The costs of foundations and floor vary with the intensity of loading (the height of storage comes into play here). The cost of roof and the perimeter walls varies with the roof height.

3. Cost of Racks. The cost of racks is treated as the sum of four elements:

1. Cost of material and labor for load arms and ties.

2. Cost of material and labor for the pairs of columns required for each truss assembly. Changes in the cross-sectional area of the columns due to increased loadings when the rack height is increased have been considered here.

3. Cost of splicing if there are any splices.

4. Cost of installing the racks.

4. Cost of Stacker Cranes. The cost of each crane is analyzed as the sum of four elements:

1. Cost of stacker crane hardware which consists of the stacker base, the vertical mast and the lifting mechanism hardware (elevator). It provides for the variation in the cost of the mast due to increasing heights.

2. Cost of providing a specific horizontal speed.

3. Cost of providing a specific vertical speed.

4. Cost of the controls. Each stacker crane design will have its own control design. In addition, the control of several individual stacker cranes may be combined for central control. The cost of operators required depending upon the level of controls is also considered.

5. Cost of Transfer Cars. This includes the cost of all the transfer cars to transfer the cranes for the given size and load requirements.

6. Cost of Fire Protection. The protection against fire appears to be best accomplished by a well designed system of automatic sprinklers, or a combination of sprinklers with high expansion foam. It may also be preferable specifically to have one of these types due to the kind of material to be stored or other constraints. Both types have been considered in the analysis and it is possible to select either one or the cheaper of the two.

Annual Cost of the System

Having determined the individual elements of cost, the equivalent annual cost can be computed. The salvage values and the write-off periods for the various elements of the system are treated as variables.

The Simulator

The operation of a high-rise warehouse using stacker cranes has been programmed into the simulator, BASS. Since warehouses vary in their layouts, requirements, and operating rules, a basic layout of a high-rise warehouse and operating rules in a general warehouse have been modelled. If any deviations from these are required, it can be done by making changes in the corresponding subroutines of the simulator.

Each aisle in the warehouse is assumed to have two queues--a material queue and an order queue. In addition, there is a marshalling area where a "common material queue" is formed. Each material queue has a finite capacity. When all of these are full, the overflow is sent into the common material queue (unassigned materials). Flow of material from the common queue to the material queue of a specific aisle takes place automatically as soon as a vacancy occurs.

Arrival of material to be stored and orders to be retrieved from the system are input based upon user demands, or alternatively, according to any desired distribution. When more than one aisle is serviced by a stacker crane, operating decisions for transfer between aisles are made by a choice of criteria including maximum waiting time before servicing orders, maximum number of orders allowed in a queue, completion of a given number of cycles in an aisle, and the material and order queues for the aisles becoming empty. BASS keeps track of the full or empty condition of each storage location and reacts to the slot condition in arriving at storage or retrieval decisions.

The simulator has the capability of evaluating alternative operating policies related to scheduling storage and/or retrieval cycles to meet operating schedules. During the simulation run, the stacker crane performs single address (deposit only or retrieve only) or dual address (deposit and retrieve) cycles as necessary depending on the available entries in the material and order queues.

Language Used

The language used in BASS is Fortran IV. To facilitate the programming of standard simulation procedures, GASP II, a Fortran IV based simulation language is used. The purpose of selecting Fortran IV as a programming language for BASS is twofold. First, Fortran compilers are commonly available. Second, BASS may need minor changes of decision-rules, etc., to suit specific needs. Fortran makes this relatively easy since a high percentage of programmers are familiar with Fortran who, with some knowledge of GASP (an event-oriented simulation language), can make the necessary changes. Besides, greater portability in running the simulator has been achieved by having the maximum length of variable names limited to five letters. This allows BASS to be run on a variety of computers.

Design Variables

The design variables described above have been used in BASS. The names assigned to them are:

KHGHT -- number of slots along height.

NAILE -- number of aisles.

NSCR -- number of stacker cranes.

HOVEL -- horizontal speed of stacker.

VEVEL -- vertical speed of stacker.

Since the first three variables can be incremented in steps of one, the upper and lower limits to be considered by the designer are specified for each of these. All possible values of the two speeds are also specified.

Operation of BASS

The principal stages of BASS are shown in the system flow chart in Figure 1. The lower limits (as specified by the programmer) of KHGHT, NAILE, NSCR, HOVEL, and VEVEL as fed in by the data cards are taken as the initial system variables. The throughput is simulated with this set of variables until the steady-state conditions, as verified by programming procedures, have been reached. This calculated value of throughput is transferred to the MAIN program and compared with the last throughput (initially set as 0). If it has increased, a new series of variable value sets are generated (this is done by increasing in each set one of the five variables to its next possible higher value) and the set giving the lowest annual cost is transferred to GASP to simulate the next throughput with. If the throughput has not increased, another set of variables (the one giving the next higher annual cost) from the series generated earlier is transferred to GASP and the throughput is calculated. This cycle is repeated over and over again until the calculated throughput is greater than the required throughput. This throughput is then checked (using the same set of variables) under more stringent steady-state conditions than before. If the revised value of the throughput still equals or exceeds the required throughput, the simulation stops and the final values of the design variables are printed. If the revised throughput is less than required, the procedure is repeated until the final requirements are reached.

Outputs from BASS

For each set of variables, BASS collects statistics on waiting times, cycle lengths, times between transfers, crane utilization, times lost in transferring and travelling empty, material and order queue build-ups in each aisle, number of dual and single address cycles by each crane, and the throughput for the system. In addition, it provides histograms on waiting times and cycle lengths. It also computes the costs and the throughput for each set of variables.

Special Features of BASS

Special features built into the simulator, to suit the specific needs of individuals who may use it, include:

1. Any size warehouse can be simulated, the only limitation being available core storage in the computer system.

2. Depending on individual requirements, the start up of the stacker crane at the beginning of simulation may be handled in either of two ways. First, the cranes can be started

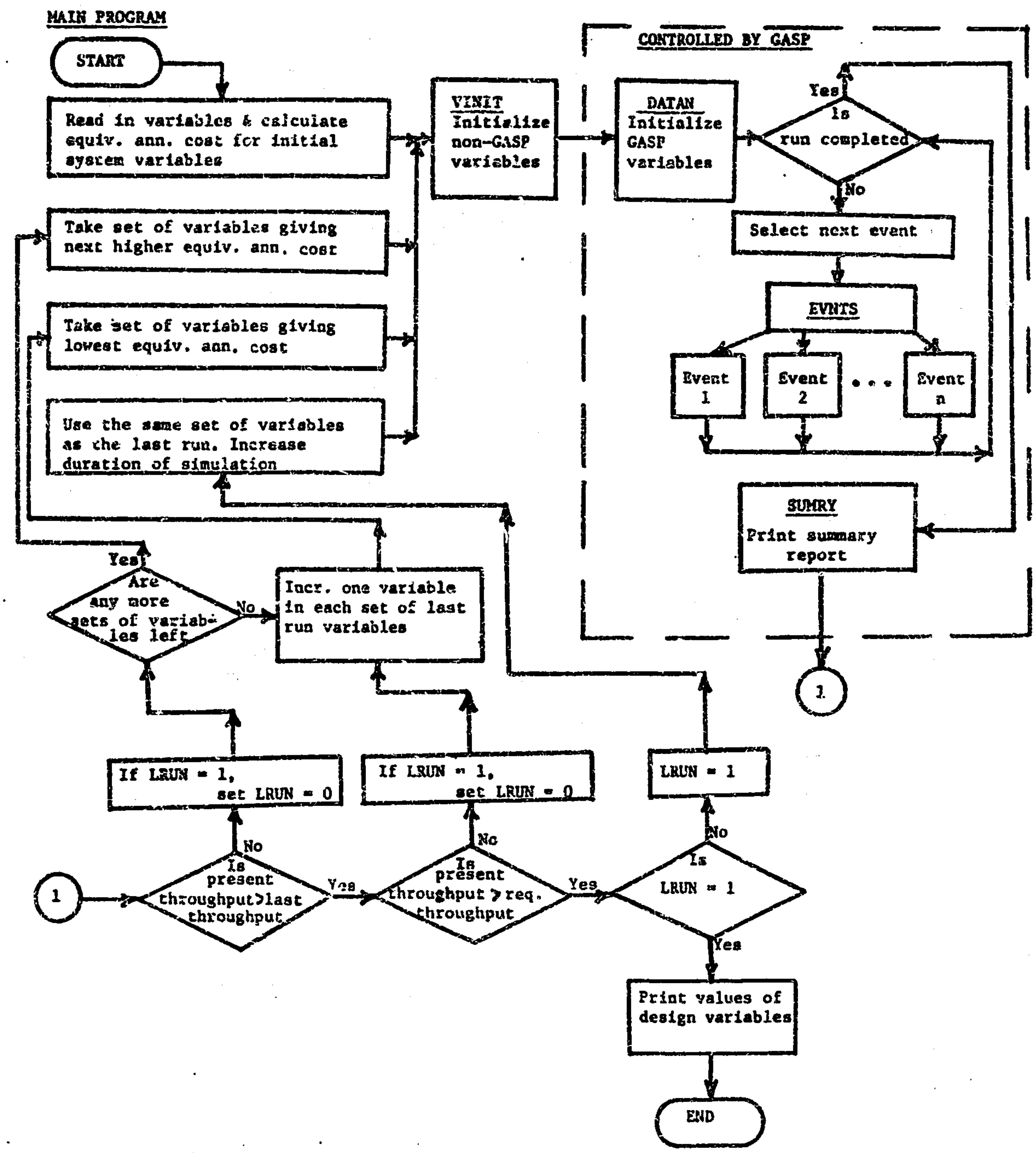

Figure 1. System flow chart showing the principal functions performed by the main routines.

whenever entries are available in the queues, i.e., the first arrivals can trigger the start of the cranes. Alternately, the cranes can start at any desired and predetermined time. All arrivals until that time are kept in queues. All that the user has to do is to specify start times of stacker cranes, such as start of shift, one-quarter hour after shift start, etc.

3. Also, the queues can be emptied at the end of the simulation before statistics are computed, or alternately, queues can be left in their end of simulation conditions and statistics collected without runout. This feature is available by assigning a value of 0.0 or 1.0 to a variable, QUEUE.

4. Since the value of QUEUE is checked (in

either case) at the end of the simulation, other analysis to occur at the end of simulation can be programmed into BASS, to be called when QUEUE has specific values. This will involve changing of a few cards in the end of cycle subroutine. This feature provides greater flexibility to users to add on special requirements without extensive program changes.

5. Depending on the amount of money or computation time available to the user, he can adjust the duration of each run to meet his specific needs of accuracy.

6. Flexibility is provided to handle situations where users do not have an accurate idea of the ranges in which the number of aisles, height of racks, and the number of cranes fall. The simulation can be done using 1 aisle, 1 crane, and 1 slot height (although this is an infeasible value, it is used to emphasize the point) as the starting point and working up until the required solution is reached. Of course, the user will have to pay for the lack of the required knowledge in increased processing time.

7. By setting the value of a variable LFIRE as 1, 2, or 3, it is possible to select the system having fire protection with sprinklers only, foam only, or the one with least cost.

8. The arrivals of materials to be deposited in the warehouse and the orders to be retrieved from the warehouse can be made to follow any given distribution with a minimum of change in the program. In addition, empirical data collected from an existing warehouse can also be used to generate the arrivals.

9. Great flexibility is provided by the fact that Fortran IV had been used as the programming language. This provides for ease in programming new decision-rules and events to suit individual requirements.

From the aforegoing, it can be concluded that BASS is a flexible simulator. The need for this flexibility arises from the fact that high-rise warehouses using stacker cranes are still in their infancy and numerous developments and changes are anticipated in the future. Because of the nature of BASS, these changes can be programmed into it without having to develop an altogether new simulator.

Validating the Simulator

In order to validate BASS, several runs have been made with different storage requirements, each with varying throughput requirements. The storage requirements selected were 600 slots, 5,900 slots, 13,500 slots, and 14,500 slots. The simulator ran for all of these sizes indicating that it could be used to design very small systems as well as large ones.

The results obtained at each step in one of these runs are summarized in Table 1. The run was made for a warehouse having approximately 5,900 storage slots and a throughput capacity of 90/hour. Exponential arrivals of materials for depositing and orders for retrieval were assumed. The values of the variables for each iteration

Table 1

Step-by-step results of a simulation run. Total storage required = 5,900 slots (approx.), and hourly throughput required = 90.0.

ITERATION	NO. OF AISLES (NAILE)	HEIGHT IN SLOTS (KHGHT)	LENGTH OF RACKS IN SLOTS (KLGTH)	NO. OF STACKER CRANES (NSCR)	HORIZ. SPEED (FT/MIN) (HOVEL)	VERT. SPEED (FT/MIN) (VEVEL)	EQUIV. ANNUAL COST ($) (EQACT)	ACTUAL THROUGH-PUT (PER HR) (HTHPT)
1	3	5	197	1	262	74	240,272	71.04
2	4	5	148	1	262	74	228,977	73.02
3	5	5	118	1	262	74	222,781	74.28
4	6	5	99	1	262	74	221,404	74.70
5	6	5	99	1	420	74	221,609	73.14
6	6	5	99	1	262	99	221,650	74.46
7	7	5	85	1	262	74	222,484	74.70
8	6	6	82	1	262	74	238,630	75.78
9	6	7	71	1	262	74	232,137	77.40
10	6	8	62	1	262	74	227,876	78.78
11	6	9	55	1	262	74	225,247	80.22
12	6	10	50	1	262	74	225,185	81.96
13	6	11	45	1	262	74	223,659	83.16
14	6	12	41	1	262	74	223,786	84.18
15	6	12	41	1	420	74	223,991	82.98
16	6	12	41	1	262	99	224,033	83.40
17	6	13	38	1	262	74	225,081	84.66
18	6	13	38	1	420	74	225,286	83.52
19	6	13	38	1	262	99	225,327	84.24
20	6	14	36	1	262	74	228,804	84.06
21	7	13	33	1	262	74	231,114	84.36
22	6	13	38	2	262	74	242,900	97.32
23	4	5	148	2	262	74	246,386	95.70
24	3	5	197	2	420	74	252,957	100.48

are shown in the table. It also shows the equivalent annual cost and the hourly throughput obtainable with the system for each run.

The results of the above run are plotted in Figure 2, which shows how, at each step, the simulator tries to increase throughput while minimizing the incremental cost. The plot has a zig-zag formation. Up to iteration 4, cost decreases with increases in throughput. Iteration 5, 6, and 7 do not show any increase in throughput. Finally, there is an increase in throughput in 8, but with a significant increase in cost. Once again cost generally keeps decreasing for increases in throughput up to iteration 17. Subsequent iterations fail to increase throughput and hence, again in iteration 22, the cost increases significantly for an increase in throughput.

Since the increased throughput in 22 is obtained by increasing NSCR from 1 to 2, the initial values of the variables are used with NSCR = 2 for 23 (Table 1). Since the throughput of 22 is better than that of 23, hence run 24 is made which yields a better throughput. However, EQACT of 24 is greater than that of 22 and hence iteration 22 is the final design.

Sensitivity Analysis

BASS is intended to aid the designer in his process of decision-making. The sensitivity of the cost to changes in the throughput at each iteration could be a valuable guide in the decision-making.

Table 2 shows the values of cost and throughput at each iteration for the data given in Table 1. The fourth column is the equivalent annual cost per unit increase in throughput at each iteration. This is plotted in Figure 3. Since this plot is for a low throughput system, it is found that the incremental cost per unit throughput is lower for higher throughputs. In iteration 22, the number of stacker cranes is increased from 1 to 2. This gives a sharp increase in throughput (about 13 units) for an increase in cost of approximately $18,000. Since a through-

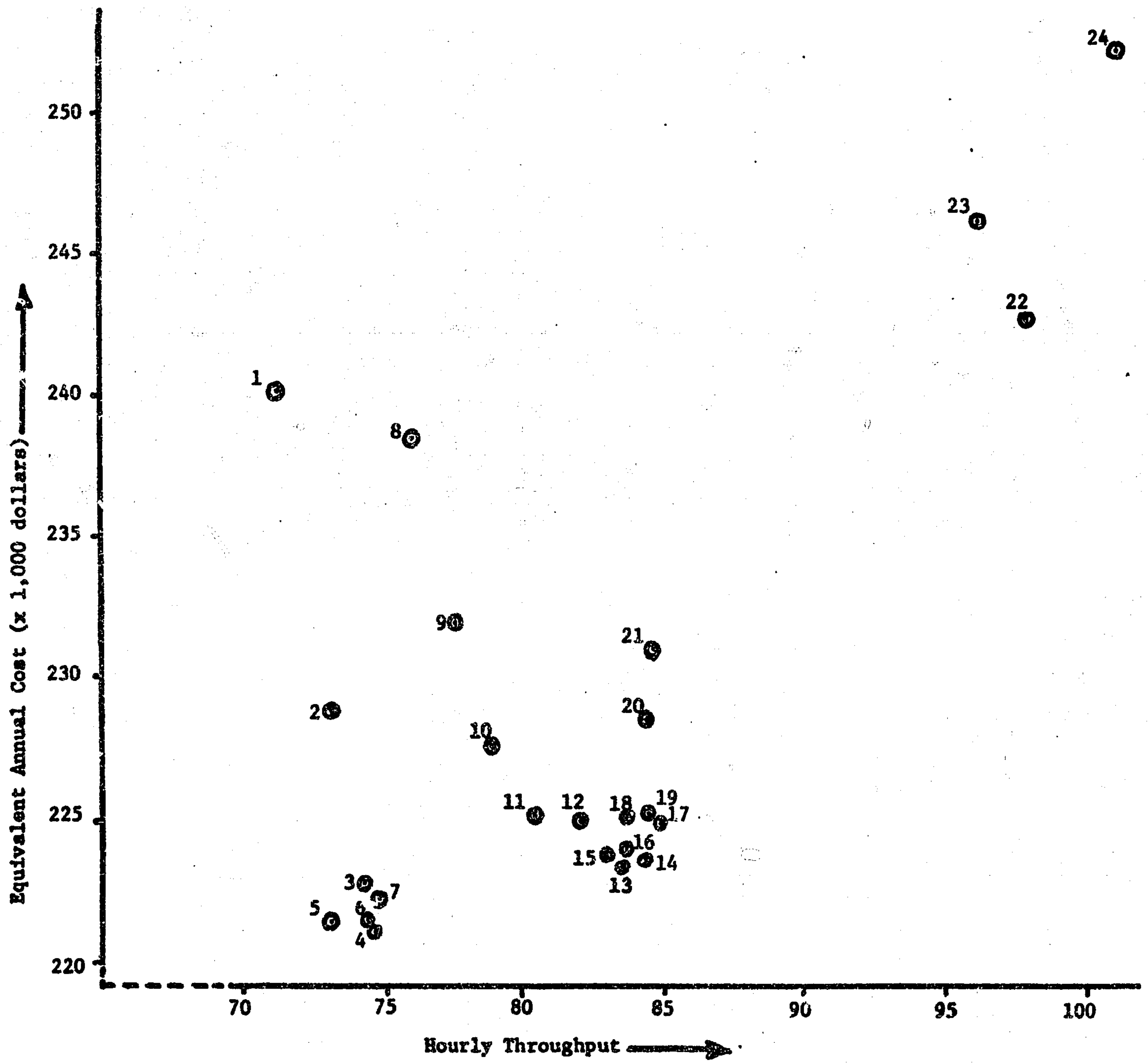

Figure 2. Plot of equivalent annual cost vs. throughput for each iteration.

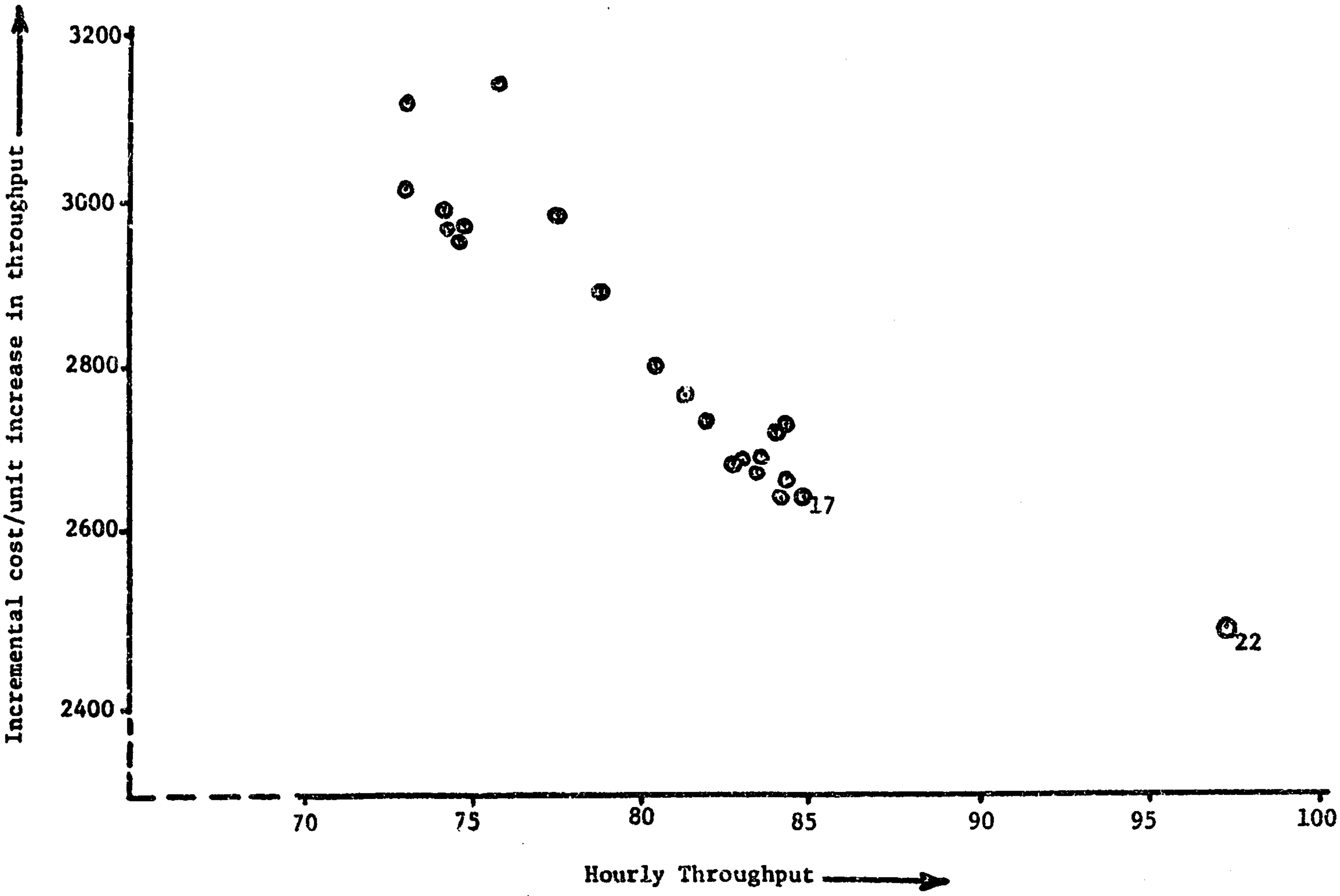

Figure 3. Incremental equivalent annual cost per unit increase in the throughput at each iteration.

Table 2

Values of equivalent annual cost and throughput at each iteration.

ITERATION	EQUIV. ANNUAL COST ($) (EQACT)	ACTUAL THROUGHPUT (PER HR.) (HTHPT)	EQACT / HTHPT	ITERATION	EQUIV. ANNUAL COST ($) (EQACT)	ACTUAL THROUGHPUT (PER HR.) (HTHPT)	EQACT / HTHPT
1	240,272	71.04	3,382	12	225,185	81.96	2,747
2	228,977	73.02	3,136	13	223,659	83.16	2,690
3	222,781	74.28	2,999	14	223,786	84.18	2,658
4	221,404	74.70	2,964	15	223,991	82.98	2,699
5	221,609	73.14	3,030	16	224,033	83.40	2,686
6	221,650	74.46	2,977	17	225,081	84.66	2,659
7	222,484	74.70	2,978	18	225,286	83.52	2,697
8	238,630	75.78	3,149	19	225,327	84.24	2,675
9	232,137	77.40	2,999	20	228,804	84.06	2,722
10	227,876	78.78	2,893	21	231,114	84.36	2,740
11	225,247	80.22	2,808	22	242,900	97.32	2,496

put rate of about 90 per hour is desired, selection could be on either side of it. The lowest cost per unit throughput for a throughput below 90 was in iteration 17. The value above 90 was in iteration 22. Because of the large difference in the incremental cost at these two points ($163 per unit throughput), it would be better to select the results of iteration 22. However, had the two points been fairly close, either could have been selected, the decision then being based upon how much capital could be made available for the warehouse.

In addition to making a decision on the basis of the sensitivity analysis, information printed out in the summary reports should be analyzed. More specifically, information such as statistics of material and order waiting times, crane utilization, statistics of aisle material queues, order queues, the common material queue, and the types of crane cycles should be studied. The waiting time, especially that of servicing orders, should not exceed that required by the system that the warehouse is to service.

Findings From Runs

The following are some of the main findings from the results of the sample runs made to test the simulator:

1. Whereas it may be necessary from an efficiency standpoint to have the ratio between the horizontal and vertical speeds conform to the ratio between the length and the height of racks, it is not absolutely necessary to have this relationship for a least cost solution.

2. The general belief among stacker manufacturers that higher warehouses are cheaper is validated by the results obtained from the sample runs.

3. Greater travel speeds are not necessary for low throughput systems and basically just add to the cost.

4. For a given system throughput, it is found that the larger the storage capacity of the warehouse, the lower is the cost per storage slot. Also, the cost per unit slot increases as greater throughputs are desired.

Conclusion

The results from the sample runs show that numerous alternatives confront the designer of high-rise warehouse and stacker crane systems and these differ from one another in small steps. It is, therefore, justifiable to conclude that the use of BASS, along with the sensitivity and report analysis described, will prove to be a very useful tool. Besides helping in design selection, it will provide useful information about the actual system which has been either impossible or impractical by earlier analysis techniques. The simulator can also serve as a tool to test different operating rules and management policies related to high-rise storage.