A PRESENTATION OF THE SIMULATION LANGUAGE SEA DESIGNED FOR THE USE OF NON-COMPUTER ORIENTED PERSONNEL IN THE SOLUTION OF SCHEDULING PROBLEMS

A Thesis

Presented to

the Faculty of the School of Industrial Engineering

The University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

by

Kevin D. Moodie

January 1969

A PRESENTATION OF THE SIMULATION LANGUAGE SEA DESIGNED FOR THE USE OF NON-COMPUTER ORIENTED PERSONNEL IN THE SOLUTION OF SCHEDULING PROBLEMS

An Abstract of a Thesis

Presented to

the Faculty of the School of Industrial Engineering

The University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

by

Kevin D. Moodie

January 1969

ABSTRACT

Many waiting line type problems can only be solved by simulation. Simulation, although very effective, must invariably be programmed and run on a high speed digital computer. To reduce programming time, a number of simulation languages have been written.

One of these, MHSS, was written by Dr. C. E. Donaghey for application to materials handling problems. This is a very easy program to use and proved effective in numerous instances. So many non-material handling situations, however, were visualized to which with little or no modification MHSS could apply, that it was decided to generalize MHSS and demonstrate its power.

The result is SEA - a service effectiveness analyzer, applicable to practically any waiting line or scheduling type problems and requiring no computer experience of the user. The paper provides a complete guide for using SEA and includes an application to a real and non-industrial system in a Houston hospital.

TABLE OF CONTENTS

I INTRODUCTION

A remarkable number of industrial problems involve some type of queuing, i.e., a number of facilities servicing a number of events. For example, aircraft moving troops, or parking spaces at a stadium. The question is usually either one of the number of facilities necessary to handle certain events, or one of the number of events a set of facilities can conveniently handle. If the facility happens to be one in a series, an assembly line for example, the analysis becomes particularly vital as a small error at some point may be compounded to drastically effect many subsequent events. As queuing systems are usually non-deterministic, their analysis is by no means trivial and much time has been devoted to studying them.

Since the turn of the century, a great deal of work has been done on the analytical approach to waiting line problems and a whole queuing theory developed. Although highly sophisticated mathematical techniques have been applied to deriving a complex set of analytical tools, there are still very many situations for which they cannot be used. Other cases exist for which the analytical approach is valid but far too complicated for reasonable solution. In these cases simulation is an invaluable standby.

Simulation, made truly effective by the high speed computer, essentially models the system under consideration and then "operates" it through a number of cycles. The outcomes are averaged and the results printed out. It has proven to be a remarkably effective and extremely powerful tool, but has the disadvantage of requiring specialized computer programming ability and the inevitable "debugging" and running time.

The problem has been reduced by the introduction of simulation languages such as "SIMSCRIPT" and IBM's "GPSS". However even these languages must be learned, and this calls for a certain amount of effort and experience. In addition they use a relatively large amount of computation time and can, of course, only be run on machines equipped with the appropriate compilers.

The Service Efficiency Analyser (SEA) presents a compromise. It requires no familiarity with programming and may be learned completely in a matter of hours. It is also very fast and is written in the commonly available FORTRAN language. Although not as general as the complete simulation languages, SEA does have broad applicability and its simplicity is believed to be a valid compensation for any lack of power.

This is a presentation of SEA. The content is divided into four sections:

II. SEA - A DESCRIPTION - This should give the reader insight into the powers of SEA while at the same time describing how the program goes about building the model.

III. USER INPUT - Here a detailed description is provided of precisely how to prepare the data cards for SEA. An example problem is included.

IV. SEA OUTPUT - A discussion of SEA output describing user control, format, and data interpretation. Also includes a complete list of SEA error messages with hints on their cure.

V. APPLICATION IN A HOUSTON HOSPITAL - A demonstration of an application of SEA in an actual hospital bed prediction problem, from data collection to final recommendations.

Equipped with this presentation and a SEA program deck the industrial engineer should have a very useful tool at hand and be able to save many hours of tedious calculation and/or programming.

II SEA - A DESCRIPTION

This section describes the type of information SEA can handle and very broadly how it does it. This is essentially a paraphrasing of Section II of Dr. C. E. Donaghey's dissertation. (See bibliography (1)).

Any waiting line problem involves some number of events which in order to occur require the services of one or more facilities. An event might be the check out of customers in a supermarket and the facility a checkout stand, or the event might be the repair of a TV set and the facilities a TV serviceman with truck. In order to utilize SEA the user must first answer questions concerning the events and the facilities required to execute each event.

The event questions are as follows:

1. During what period of time is it possible for the event to occur?
2. What is the time interval between occurrences of the event?
3. How long does the event take?
4. What facilities does the event require?
5. Does the occurrence of an event interfere with the occurrences of any others?
6. Are there alternate ways of performing the event?

7. Does the event generate subsequent events?

8. Has the event any priority over other events?

Each one of these event questions will now be examined in more detail.

1. During what period of time is it possible for the event to occur?

   Initially a cycle length for the simulation must be established. This may be an eight hour shift, a thirty day month, twelve month year, or any other desirable period. To answer this question it must be decided when in the cycle the event can occur. For example the breakdown maintenance might occur at any time during a normal eight hour shift, but routine maintenance could be scheduled to occur only from say 7:00 a.m. to 7:30 a.m.

2. What is the time interval between occurrences of the event?

   The time between event occurrences may be constant, drawn from built in SEA statistical distributions, or read from user provided histograms.

3. How long does the event take?

   The length of time facilities are occupied satisfying one event. Again, as in question 2, this time may be constant or drawn from distributions.

4. What facilities does the event require?

Facilities necessary for the occurrence of an event. For example an operation in a hospital might require a surgical team, an operating theatre, and a transplantable organ before it can occur.

5. Does the occurrence of an event interfere with the occurrences of any others?

An event might, by its very nature, prevent the occurrence of some other event despite the fact that the latter has all facilities available. For example a system in which TV and power supply repair are two possible events may be under consideration. In this case it might be impossible for the TV to be repaired until power is restored - even if the TV repairman and his equipment were available. The power supply in this case would be the blocking event and might of course block more than one event simultaneously.

6. Are there alternate ways of performing the event?

Are there optional, perhaps less desirable facilities for executing an event? An automobile instead of a delivery truck, or a stairway instead of an elevator.

7. Does the event generate subsequent events?

Some events trigger following events. The following event might occur immediately (departure of a loaded truck) or at some later time (3,000 mile checkup following pre-delivery preparation of a new automobile). This time may be drawn from a distribution, and any number of following events may be triggered.

8. <u>Has the event any priority over other events?</u>

   Certain events may have priority over others - feeding an assembly line before removing scrap.

The facility questions are as follows:

1. During what period of time is the facility available?
2. For what proportion of the available time is the facility able to execute events?
3. When a facility becomes unavailable - for how long is it not available?
4. How many facilities of this type are available?

These will now be discussed as follows:

1. <u>During what period of time is the facility available?</u>

   A facility might not be available for a full cycle. A fork lift may for example be available only during afternoons. This time period must be specified.

2. <u>For what proportion of the available time is the facility able to execute events?</u>

   The proportion of the above available time for which the equipment is actually available must be specified. The facility might be subject to fatigue or breakdown at random periods during an available period leading to some percentage effective availability.

3. <u>When a facility becomes unavailable - for how long is it not available?</u>

   This must be specified to inform the program whether the facility is away for long periods of time or shorter periods more frequently.

4. <u>How many facilities of this type are available?</u>

   The user may specify any number of identical facilities. For example 250 baskets in a supermarket or 12 automobiles in a rental pool.

The data obtained from the answers to the event and facility questions are the parameters to the SEA model which operates as described in the following paragraphs.

SEA uses three main categories of events. They are primary events, optional events, and following events. A primary event occurs entirely on its own during the cycle at specified times. The times may of course be specified non-deterministically. A following event is triggered by some other event and an optional event occurs only when a primary or following event cannot take place. Initially SEA schedules the first occurrence of all primary events. It also schedules the first report (printed output) and the start of the availability period of each piece of equipment.

SEA now withdraws the earliest chronological occurrence from the schedule and examines it to see whether it is a report, an event, or a change in facility status. If more than one item occurs at the same time, changes in facility status are handled first, events second and finally reports. Facility status changes are handled first as often these will represent facility breakdown. For simplicity SEA will not allow facilities to break down while handling an event. This should not affect the realism of the model unless some event is scheduled to last for some large proportion of cycle time. For this case the event should be broken into smaller following segments thus allowing the facility to fail. Reports are given lowest priority as they only occur at the end of simulation cycles, and all activities should be completed before ending the cycle. Event priorities are also recognized, if assigned, otherwise they are handled on a first in first out basis.

Once SEA has determined an event is to occur, it first investigates to see if any other event is in progress blocking the event.

If the event that is to occur is blocked by a blocking event already in progress, SEA looks ahead to see when the blocking event will be completed. If the event that is to occur has no optional event, the event will be put back in the schedule to occur at the time the blocking event is completed. If the event does have an option, however, SEA must determine if an attempt should be made to execute the option.

In preparing data for SEA, the user must specify how long the event should wait before initializing the optional event. When a demand for a primary event occurs and the event cannot immediately be executed, perhaps becuase of blocking, SEA will not always at once attempt the optional event. For example, a primary event might require an overhead crane as its facility. An option for this primary event might require a fork lift truck. It would not be too realistic to assume that a machine operator will begin to unload his machine with a fork lift truck when the overhead crane, which can much more easily execute the move, will be available in three minutes. It would be a different situation if the crane was going to be blocked or otherwise unavailable for an hour. In that case he would probably use the truck to unload his machine.

The SEA user must specify to the model how long an event should be allowed to wait rather than use an optional event. During execution SEA assumes that the time a blocking event will be completed or facility will become available is known. In the above example assume the user had specified the primary event could wait as much as five minutes rather than immediately execute the fork lift truck event. Assume a demand for the primary event occurred at 10:00, and it was found the event was blocked until 10:20. In this situation, SEA would initiate the optional event at 10:00. If the blocking event was due for completion at or any earlier than 10:05, then the primary event would be delayed until then, the option is only used if it is determined the primary event must wait more than five minutes.

If no blocking is found for an event, SEA next checks facility availability. Each facility required for the event must be free. If any one is not free and the event has no option, the event will immediately be rescheduled to occur when the facility becomes free. If the event has an option and the time until the facility will be free is longer than the time the user specified the event should wait, the option is scheduled to occur immediately.

When it has been found that the event is not blocked and the facilities are free, the event is started. The length of time the event is to take will be determined from the specifications furnished by the user. The tags on the required facilities will be changed to show when the event will be completed, and the facilities again become available to execute other events. Tags and tallies connected with the event will be changed and updated to keep track of the number of times the event occurred, the length of time the event was delayed, the length of handling time, and other data needed for reporting and operation of the model.

If the event has any following events that are to occur immediately upon its completion or some time after its completion, they are then added to the schedule along with the time they are to occur. Primary, optional, and following events themselves can all have following events.

If the event just started is a primary event, the next occurrence of the event will be added to the schedule. The parameters which the user furnished to SEA concerning the time between demands for each of the primary events are used in determining when the next occurrence of a given primary will take place.

The second type of occurrence SEA can find in the schedule is a change of facility status. There are two ways that a facility can change its status. It can become available to the system being simulated or, the second way, it can become unavailable to the system. A facility in the process of executing an event for the system being simulated does not have its status changed until it completes the event. While in the process of executing an event for the system, it is still considered by SEA to be available to the system. A change in facility status could be caused by such things as equipment breaking down, equipment returning from a breakdown, personnel going to another department not included in the simulation, and personnel returning from another department. If the event is a facility becoming available to the system, the tag for the facility which records when the facility will be available is changed to contain the current time. If the event is a facility becoming unavailable to the system, SEA must determine when it will next become available, and record this value in the availability tag for the facility.

Each time a facility changes its status, the time for the next change is determined and added to the schedule. For a facility that is becoming unavailable to the system for the remainder of the cycle, no new change of facility status need be added to the schedule.

The third type of occurrence SEA can find in the schedule is a report. These conclude each cycle. There are two types of reports - end of cycle, and summary. If desired, end of cycle reports may be suppressed by the user in which case only a final summary of the output is printed. Output is discussed in greater detail in a later section.

III USER INPUT

This section describes in detail the input data necessary to run SEA. The data falls into four distinct categories.

1. System parameters
2. Facility information
3. Event information
4. Distribution specification

Before discussing each of these separately, it is necessary to outline the method of calling built in statistical distributions. This is done by specifying three parameters as described in Table I.

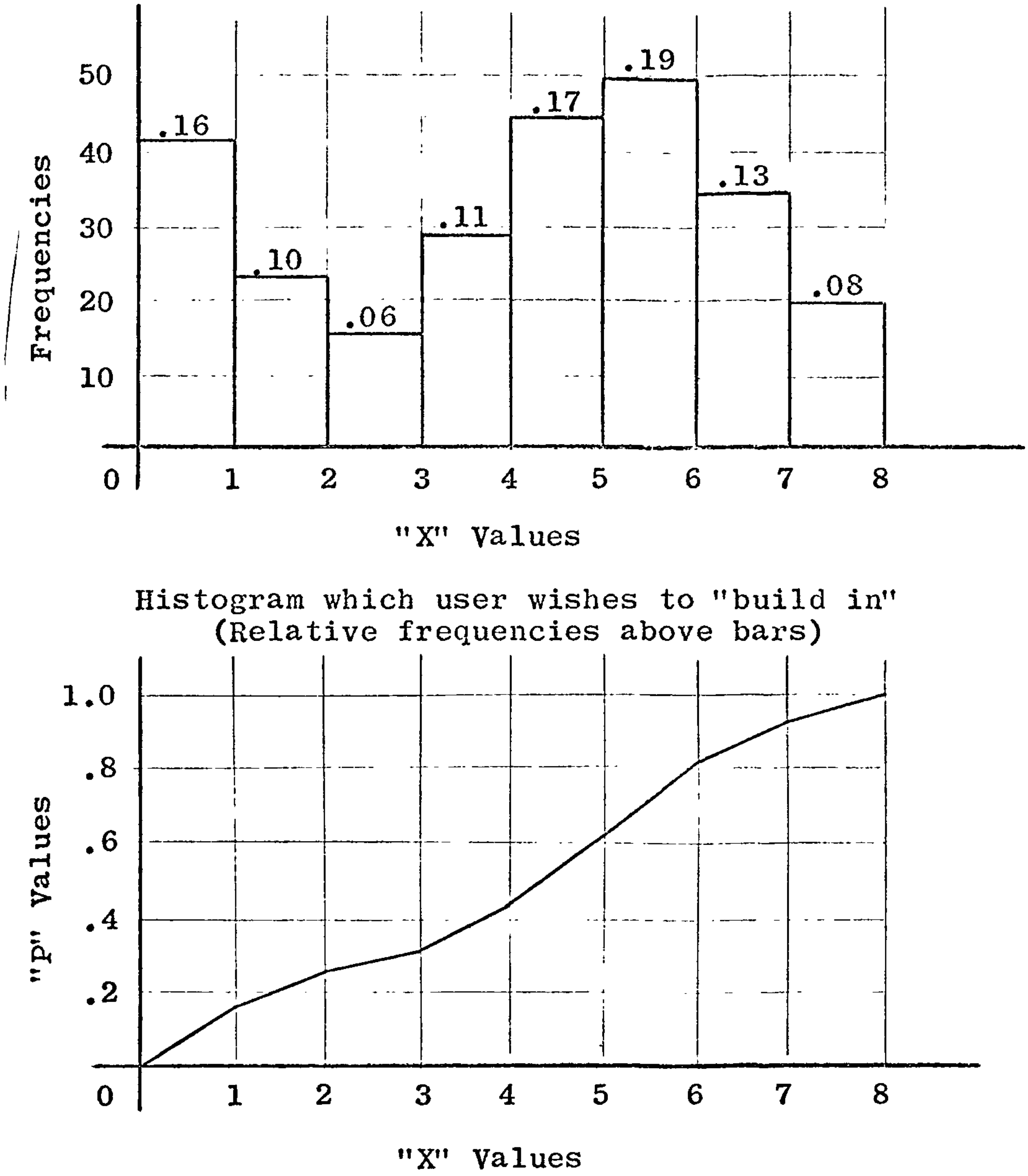
Note that the third parameter is always an integer for selecting the type distribution required. For example, a normal distribution with mean 10 and standard deviation 2.5 would have the three parameters respectively 10, 2.5, 3.

Whenever a parameter is zero it may be left blank.

For calling user specified distribution, only the third parameter is required.

Having specified a distribution the program will thereafter select random values from this distribution whenever necessary.

With the answers to the relevant questions in Section II, and the preceding information, data cards may be prepared. They fall into the four categories listed above.



Histogram which user wishes to "build in"
(Relative frequencies above bars)

Cumulative distribution of above histogram

Note: The eight pairs of corresponding "P" and "X" values, from the cumulative distribution, must be punched onto cards as per Table (5)

There may be fewer than eight but "P"must range from zero to one, inclusive.

FIGURE (1)

METHOD OF DERIVING USER DISTRIBUTIONS FROM USER DATA

TABLE (1)

PARAMETERS FOR CALLING VARIOUS DISTRIBUTIONS

Distribution	1st Parameter	2nd Parameter	3rd Parameter
Rectangular	Min. Value	Max. Value	1
Exponential	Mean	Min. Value	2
Normal	Mean	Std. Dev.	3
Constant	Value	Blank	6
Zero	Blank	Blank	9
Erlang Class 1 to 10	Mean	Min. Value	11 to 20
User	Blank	Blank	21 to 99

1. System Parameters

This consists of a single card containing information on length of cycle, random variable seed, etc. as listed in Table (2) and fully discussed below:

NOSSM - Number of cycles may be any reasonable number but should be kept as small as practical to reduce computer time.

HRSH - Time period for each cycle may be in any units but the units used here specify time units throughout the program.

IHOW - If left blank, end of cycle reports will be printed. Any non-zero value here suppresses all but the summary report with consequent reduction in computer time.

NSEED - Any positive integer, preferably prime, serves as the seed in the random number generator. The same problem solved with the same NSEED should give identical results. Varying NSEED gives similar and valid, but different results.

WTM1 - SEA will print out at the end of the simulation the percentage of events waiting longer than this time, which may take any value from 0 to WTM2.

TABLE (2)

SYSTEM PARAMETER CARD PUNCHING INFORMATION

Col.	Variable	Description	Range	Spec.
1-5	NOSSM	Number of cycles	1-1,000	I5
6-10	HRSH	Time per cycle	0-99,999	F5.0
11-15	IHOW	End of cycle report spec.	0 or 1	I5
16-20	NSEED	Random number seed	1-99,999	I5
21-23	JXX	Alphanumeric blank	Blank	A3
24-28	WTM1	1st Waiting index	0-HRSH	F5.0
29-33	WTM2	2nd Waiting index	WTM1-HRSH	F5.0

WTM2 - Any value greater than WTM1, but less than HRSH. For example, WTM1 might be set equal to .5 days and WTM2 3 days. SEA tabulates the percentage of each event type waiting longer than these times.

2. Facility Parameters

One card is required for each different type of facility. Twenty trucks and ten operators for example would constitute thirty facilities of two types and require two cards. The information is listed in Table (3) and discussed below.

JEQ - Facility code - any three alphanumeric characters.

TMEQS - The time at which the facility becomes available to the system. If it becomes available at the beginning of the cycle this may be left blank. It must obviously be less than the end of cycle time.

TMEQN - The time at which the facility becomes unavailable. Must be greater than TMEQS but less than (or equal to) HRSH. If a facility becomes available twice in one cycle a card with the same facility code and data but new TMEQS and TMEQN must be provided. SEA will treat this as two separate facilities,

TABLE (3)

FACILITY PARAMETER CARD PUNCHING INFORMATION

Col.	Variable	Description	Range	Spec.
1-3	JEQ	Facility code		A3
4-7	TMEQS	Start of facility availability	0-HRSH	F4.0
8-11	TMEQN	End of facility availability	0-HRSH	F4.0
12-15	PEQAV	Proportion of available time effective	0-1.00	F4.0
16-19	TEQGN	1st parameter for time facility gone		F4.0
20-23	TEQ2P	2nd ditto parameter		F4.0
24-26	JEQGD	3rd parameter (distribution)		I3
27-31	NEQSPC	Number identical facilities	0-750	I5

but the realism of the model will not be affected. This case might arise with an operator taking a lunch break, or a machine being used in another department in the middle of a cycle.

PEQAV - Fraction of time a facility is effectively available to the system during an availability period. It can be any number between .001 and 1.000. If PEQAV is 1.0, TEQGN, TEQ2P and JEQGD should be left blank.

TEQGN - First parameter for time facility is gone - that is the time the facility is away during an availability period.

TEQ2P - Second parameter for time gone.

JEQGD - Third parameter, an integer describing the distribution.

NEQSPC - Number of identical facilities of this type. Total number of facilities must be less than 750 and there can be no more than 50 types. For example 700 trucks and 50 service ramps would fill the program with 750 facilities and two types. The program can easily be redimensioned to increase this. If NEQSPC is left blank SEA assumes a value of 1.

3. Event Paramters

As in the facility parameters, one card is required for each event. The maximum is 250 events which should be adequate for the majority of applications. The program can of course be redimensioned as required. Refer to Table (4) for a summary of the event parameters discussed below.

MC - A three character alphanumeric code assigned each event.

MOWN - If this event (i.e., the one described on this card) is an option for some other event, termed "owning event", MOWN will be the MC code of the owning event.

MFPON - If this is a following event set equal to 1. Otherwise leave blank.

JEQR1 - Alphanumeric code for the first facility required to execute the event. The facility must be described on the facility list and JEQR1 may not be blank.

JEQR2 - Alphanumeric code for the second required facility. This may be blank or any facility from the facility list. It may be the same as JEQR1 if for example two facilities, say workmen, are required to service this event. If left blank SEA will not check for the third facility.

TABLE(4)

EVENT PARAMETER CARD PUNCHING INFORMATION

Col.	Variable	Description	Range	Spec
1-3	MC	Event code		A3
4-6	MOWN	Owning event code		A3
7	MFPON	1 if following event	1 or 0	I1
8-10	JEQR1	1st required facility		A3
11-13	JEQR2	2nd required fac.		A3
14-16	JEQR3	3rd required fac.		A3
17-20	TMLST	Start period for event	0-HRSH	F4.0
21-24	TMLFN	Finish period for event	0-HRSH	F4.0
25-28	TMMIN	1st param for event duration		F4.0
29-32	TMMAX	2nd param for event duration		F4.0
33-34	MDIS	3rd param for event duration		F4.0
35-38	TNMMN	1st param time between events		F4.0
39-42	TNMMX	2nd param time between events		F4.0
43-44	NMDIS	3rd param time between events		F4.0
45-47	MCOP	Code of optional event		A3.
48-51	TMCOP	Time rather than option	0-HRSH	F4.0
52-54	MCFOL	1st following event		A3
55-58	T1FL1	1st param time to 1st fol.		F4.0
59-62	T1FL2	2nd param time to 1st fol.		F4.0
63-64	JD1FL	3rd param time to 1st fol.		I2
65-67	MCFL2	2nd following event		A3
68-71	T2FL1	1st param time to 2nd fol.		F4.0
72-75	T2FL2	2nd param time to 2nd fol.		F4.0
76-77	JD2FL	3rd param time to 2nd fol.		I2
78	MPRI	Event priority	0-9	I1
79	MBLK	Blocking list specification	0 or 1	I1

JEQR3 - Alphanumberic code for the third required facility. This may be blank or any facility from the list. It may be the same as JEQR1 or JEQR2 or both.

TMLST - Start period for the event - any value between zero and end of cycle time (HRSH).

TMLFN - Finish period for the event. This value must be greater than TMLST and less than or equal to HRSH. This event will only occur during the period between TMLFN and TMLST. If an event can occur at two or more different periods - for example from 8-11 then from 3-5 during an 8 to 5 day simply give it a new code and vary TMLST and TMLFN keeping all other parameters constant. These may then easily be interpreted as one event although regarded as two by SEA.

TMMIN
TMMAX
MDIS - Parameters for specifying the length of time an event may take.

TNMMN
TNMNX
NMDIS - Parameters specifying time between occurrences of events. If the event occurs on the average of 8 times during an 8-hour shift, then TNMMN would be set equal to 1 hour - the time from one occurrence to the next.

MCOP - Optional event code. If this event has an alternate, MCOP will be the alphanumeric code for the optional event specified; which must be listed and must have an MOWN identical to the MC of this event. For example the event being described on this card might be the stay in the hospital of some patient, coded PIH. If the hospital happened to be full, an optional event (coded PIN), might be treating the patient in a nursing home. MCOP for PIH would then be PIN and event PIN would have to have MOWN equal to PIH.

TMCOP - Time to wait rather than use the option. If zero, the optional event will immediately be started upon finding the owning event obstructed in some way (blocked or lacking a facility). Any value given TMCOP is interpreted as the time to wait before using the option. PIH above might prefer to wait one day for a vacancy. If a bed is to become available within the next day, SEA will ignore the option and schedule the patient for admittance. If however a bed will not be available on the next day, SEA immediately checks the patient into the nursing home.

MCFOL - Alphanumeric code for the first following event, i.e., any event triggered by this one to occur either immediately or at any later time.

TIFL1
TIFL2
JD1FL - Parameters for time to first following event. The following event will be scheduled to occur by these parameters, based on the time from the completion of this event until start of the first following event. Setting JD1FL = 9 would cause the following event to be scheduled immediately upon completion of this event.

MCFL2 - Alphanumeric code for second following event. If more than two following events are described MCFL2 would be a dummy event, i.e., a move taking no time but with in turn two more following events. This may be repeated any number of times.

T2FL1
T2FL2
JD2FL - Parameters for time to second following event. This time would be from completion of initializing event until start of second following event.

MPR1 - Event priority. An integer from 0 to 9. Nine will be given top priority ranging down to zero as lowest priority. Events of the same priority will be handled on a first in first out basis.

MBLK - If this event blocks any other or others, set MBLK = 1. Otherwise leave blank.

4. Distribution Specifications

There are two cards required per distribution. The first names the distribution - the second gives the points as shown in Table (5) and Figure (1).

NDIS - Some integer greater than 20 which will be interpreted as the name of a user furnished distribution.

X,P - Points from the user distribution curve.

The preceding cards, described under 1, 2, 3, and 4, comprise the program data. The data deck must be assembled as follows:

(1) System description card.

(2) All facility parameter cards - order immaterial.

(3) Blank card signifying end of facility list.

(4) Event parameter cards - order is immaterial but any events blocking other events, i.e., with MBLK = 1, must be followed by a listing of all the events blocked. The list is made up by punching the code of each blocked event in the first three columns of separate consecutive cards. A blank card must be inserted to signify the end of the blocking list.

TABLE (5)

DISTRIBUTION PARAMETER CARD PUNCHING INFORMATION

Col.	Variable	Description	Range	Spec
1-5	NDIS	User assigned number for a distribution to be read	21-99	I5
1-80	X,P	Points from user distribution (see Fig(1))	P/0-1.0 X/none	16F5.0

(5) A blank card signifying the end of the event listing.

(6) A card with any user specification followed by the points specified. Any further user distributions would follow. The final distribution must be followed by a blank card. If there are no user distributions (6) will merely be a blank card.

NOTE: It might be possible to end the deck with three blank cards if the last event was a blocking event and no user distributions were furnished. The first blank card would terminate the blocking list, and the second the event list. The third indicates no user distributions. Omitting any one of these three cards would result in an immediate error.

Any number of blocks of data may be consecutively stacked. SEA will execute them all in their turn.

Any zeros may be left blank as may any non-applicable values. For example if an event has no option, simply leave MCOP and TMCOP blank.

In order to clarify the input, the following problem was considered. A supermarket manager has gathered data on his customers and now wishes to test the effect of increasing baskets, converting one checkout counter into an express counter, or even increasing the number of checkout counters.

The customers fall into three categories:

(1) Weekly shoppers doing a full weeks family shopping.

(2) Large order shoppers buying less than the weekly shopper, but nevertheless purchasing large quantities.

(3) Short order shoppers needing only a few items and often in a hurry.

Shopping times and checkout times were found to be normally distributed while arrivals at the store are Poisson. The store remains open ten hours per day and has at present 200 shopping baskets and seven checkout stands.

Table (6) shows an input to SEA testing the effect of having 250 baskets and converting one checkout counter into an express booth during the second half of the day only. Notice that each shopper creates two events, "shopping" and "checking out". "Shopping" is treated as a primary event with "checkout" an immediately following event. The short order shoppers are assumed to use the express counter when it is open, but are given the option of using any regular counter if ever it becomes necessary to wait more than six minutes in the express line. Notice also that large order shoppers do not arrive during the first two hours of the day, and that only short order shoppers are permitted to use the express booth.

TABLE(6)

DATA DECK REPRESENTATION FOR SUPERMARKET EXAMPLE

```
   /     1111111111222222222233333333334444444444555555555566666
1234567890123456789012345678901234567890123456789012345678901234

  ' 2 600.    75477

BST    600. 1.0             250

COT    600. .95 5. 1.5   3    6

CEX300.600. .95 5. 1.5   3

          BLANK   CARD

WK1    BST          600.60. 10.0 3 3.       2       WK2         9

WK2   1COT          600.10.0 2.0 3

LO1    BST      120.600.20. 7.00 3 .80      2       LO2         9

LO2   1COT          600.8.002.00 3

SO1    BST          600.5.003.00 3 3.0      2       SO2         9

SO2   1CEX          600.4.001.00 3          SOP 6.0

SOPSO2 COT          600.4.001.00 3

          BLANK   CARD

          BLANK   CARD
```

Note: BST - shopping basket
COT - regular checkout counter
CEX - express checkout counter
WK1 - weekly shopper shopping
WK2 - weekly shopper checking out
LO1 - large order shopper shopping
LO2 - large order shopper checking out
SO1 - short order shopper shopping
SO2 - short order shopper using express counter
SOP - short order shopper using regular counter

Each line represents one data card
1 represents card column 18 (typical)

Working through Table (6) in conjunction with Tables (1) through (4) should clarify the use of SEA to the new user. A discussion of output appears in the next section. This describes output format and presents certain hints on interpretation of the data. A complete listing of error messages is also included to aid in any necessary debugging.

IV SEA OUTPUT

SEA has three standard output possibilities. They are:

1. End of cycle reports
2. Summary report
3. Error messages

The user has only the option of suppressing end of cycle reports. He has no control over output format. This has, to date, proved to be no particular disadvantage. As all output, however, is controlled in a single subroutine, modifications and/or additions are relatively easy to make. This section discusses the standard output formats and includes an interpretation of error messages.

1. End of cycle reports

Unless suppressed these appear at the end of each cycle. The averages of each value are stored and printed at each end of cycle report.

Event data is tabulated first. For each event the following values are printed opposite the event code name.

(a) WAIT TIME - The time this event waited for facilities during the last cycle. It is not the time per occurrence, but total time this event waited. To find the average wait per occurrence divide WAIT TIME by the number of occurrences.

(b) NO. OCCURS - The number of times this event occurred.

(c) HANDLING TIME - The time spent satisfying this event. As with WAIT TIME this is total handling time not time per occurrence.

(d) AVERAGES TO DATE - The values of (a), (b), and (c) averaged over the number of cycles completed at time of printout.

Similarly for facilities the data on each is printed opposite the respective facility codes in columns headed as follows:

(a) WORK - The time spent working or occupied.

(b) IDLE - The time spent idle, i.e., available but without any demands for service.

(c) AWAY - The time spent away, i.e., not available to the system.

(d) AVERAGES TO DATE - Averages of all completed cycles are printed out with each report.

This completes the end of cycle report. Of probably greater interest and worth is the Summary Report.

2. Summary report

This again writes first all the event information in columns opposite event codes.

(a) AV. WAIT - The wait time per event per cycle averaged over all cycles. This is time for each coded event not for each occurrence of this event, so it must be divided by AV. NO. OCCURS if average wait per occurrence of this event is desired.

(b) AV. NO. OCCURS - The average (over the cycles) number of times the occurrence of this event was initiated. As SEA will not build a queue but generates a new occurrence only after initializing the previous one, this number approaches the number of completed events.

(c) AV. HAND. TM. - The average time spent satisfying this event.

(d) The next six columns the maximum and minimum values of (a), (b), and (c) extracted from end of cycle results. These values give a good idea of the expected range.

(e) LGR THAN "WTM1" - The number of times the event waited longer than the value of WTM1 specified on the system parameter card. The value is expressed as a percentage of the average number times the event occurred.

(f) LGR THAN "WTM2" - As in (e) except based on WTM2.

Facility data follows in the next block arranged in similar columns.

(a) AV. WORK - The average (over the cycles) time the facility was in use.

(b) AV. IDLE - The time the facility was available but not in demand.

(c) AV. AWAY - The average time the facility was away during an availability period.

(d) The maximum and minimum values in the right hand six columns give the maximum and minimum values recorded on the cycles.

(e) PCT TM IDLE - Here the total time idle is printed out and expressed as a percentage of total facility time.

This completes the discussion of the normal output from SEA.

3. Error messages

In case of an error in input, the program will stop and print an error message. These are interpreted below.

(a) BLOCKING CODE ERROR EVT

This indicates that the event EVT listed as a blocked event is not listed as an event on the event parameter list.

(b) 7 EVT OPTIONAL EVENT NOT ON LIST

The event listed 7th on the event parameter list calls for an option EVT. EVT has, however, not been listed as an event on the event parameter list.

(c) 2ND FOL BAD EVT

EVT, described as a second following event, does not appear on the event parameter list as an event.

(d) OWNING EVENT NOT ON LIST EVT

Some optional event specifies EVT as its owning event but EVT is not listed.

(e) NO FOLLOWING EVENT EVT

Event EVT calls for some following event which does not exist.

(f) NO FACILITY OR OPTION EVT

Event EVT has no facility defined and no option - events must have at least one facility to run on SEA.

(g) BAD DISTRIBUTION

Some distribution has been called for which specification does not exist. Check the third parameter on all distribution definitions. Most probably a value greater than 20 has been used without a user distribution being specified.

(h) BAD DISTRIBUTION 23

User distribution number 23 has been used incorrectly. Check that all "P" values range from zero to one. "0" and "1" must be included as the first and last "P" values.

(i) EXCESSIVE MOVES

The schedule has been overfilled. Either reduce the size of the problem or increase the dimensioning on subroutine FILE.

This concludes the list of SEA error messages indicating user mistakes and also concludes this section, and the user guide. The sample application of Section V should clear up any further gray areas, and at the same time provide an interesting demonstration of the power of SEA.

V APPLICATION IN A HOUSTON HOSPITAL

A certain specialized hospital in Houston is at present operating at maximum capacity. It is, in fact, overloaded as it currently diverts certain patients to various annexes. As this situation is liable to worsen due to increasing numbers of patients, the hospital authorities are considering a building addition. They require a prediction, as scientific as possible, of the size (measured in number of beds) addition to specify.

Treating beds as facilities and patient's stays in hospital as events, this is an elementary problem for SEA. As described later, however, it cannot be solved analytically. The approach to the project from raw data to final recommendation is discussed as follows:

1. Data collection and treatment
2. Approach to problem
3. Results and conclusions

Each of these is enlarged upon below:

1. Data collection and treatment

Data was collected from the following sources -

(a) The monthly summary sheet from 1963 to 1968.

(b) The inpatient reports for May through October, 1968.

(c) The daily reports for July through October, 1968.

The monthly sheet provided only the average percentage occupancy for the last five years - 83.4% - and the average number of admissions for the same period - 502 per month. An attempt was made to predict future trends, but due to the fact that maximum hospital capacity effectively suppresses any increasing trend in patient admission rate, the attempt proved futile.

The inpatient reports provided a breakdown of the 500 odd arrivals into patient types (patients typed according to treatment to be administered) as listed in Table (7). As patients requiring surgery are given priority where possible, an attempt was made to predict future increases by plotting their arrivals only. This gave an increase of about 1% per month, but was considered unreliable as it covered only the brief period from May to October, 1968. Inpatient reports were not maintained prior to May, 1968.

Figures for patient time in hospital at time of dismissal were extracted from the daily reports. These were grouped according to patient type and run on the computer to determine means, variances, and distribution types. The results are included

TABLE (7)

DATA FOR HOSPITAL PROBLEM

Patient	# per month	mean stay	Distribution
GAS	14.75	16.95	Erlang Class 4
URY	35.00	10.00	Erlang Class 2
GYN	103.80	8.19	Erlang Class 1
HAN	88.80	16.50	Exponential
HEM	20.50	18.54	Exponential
GMD	37.30	21.86	Exponential
MDB	14.00	15.34	
NER	10.00	20.64	Erlang Class 4
PST	6.80	9.29	Erlang Class 2
PED	20.50	20.08	Exponential
RAD	5.30	9.03	Erlang Class 2
GSR	80.20	14.52	Erlang Class 2
TSR	25.00	15.93	Normal (Std. - 10.8)
IED	9.30	11.44	Exponential
ORT	11.00	9.33	Normal (Std. - 4.6)
OPT	9.00	9.18	Normal (Std. - 8.4)
DET	28.30	23.51	Exponential
RES	9.00	14.10	Normal (Std. - 10.0)

in Table (7). Distribution types were estimated by eye from the histograms printed out. The five major types were then confirmed by means of a Chi Square test. As may be seen, the stays vary considerably both in mean length and distribution. This alone was of interest to the hospital authorities, who had no idea how long patients spend in the hospital, apart from their awareness of the fourteen day overall mean.

The arrivals of GYN patients, the largest sample, were checked for Poisson arrival rates. The remainder were assumed to follow suit.

2. Approach to problem

The problem was to determine the number of beds (facilities) to add in order to handle an increased number of patients (events).

Beds were classified into three groups -

(a) PBD's - beds for pediatric patients

(b) RBD's - beds for research patients

(c) BED's - beds for all other patient types.

As there appears to be little classification or segregation of non pediatric or research patients, this rather general grouping seemed most realistic. However, in the event that more than one patient is waiting for a BED, surgical patients are given top priority followed by radiology patients and then those requiring medical treatment.

As there are a number of events with different arrival rates, days in hospital, and priorities, all using the same type facility (BED), an analytical solution becomes impossible using existing techniques.

Doctors had estimated they would need 610 BED's, 40 PBD's, and 50 RBD's, for a total of 700 beds. It was assumed that they, the doctors, were thinking in terms of number of patients rather than number of beds. Their estimates were used as shown below to arrive at future patient arrivals.

$$\begin{array}{l}\text{Factor of increase}\\ \text{in patient arrivals}\end{array} = \frac{\text{Doctor's bed estimates}}{\text{Current number of beds}}$$

The factors turned out the following predictions -

BED patients - 2.58 x 470 = 1,410

PBD patients - 1.82 x 20.5 = 37

RBD patients - 2.50 x 9.0 = 23

The actual number of patients predicted is not really the concern of SEA. Of importance is that given any prediction SEA can arrive at the number of beds required.

The cycle length was determined by making runs over a five year, a ninety day, and a thirty day cycle. Using percentage idle time as the criterion, the five year and ninety day cycles gave closely agreeing results. The thirty day cycle, however, showed the effects of the stabilizing period, and was therefore considered unreliable. The ninety day cycle was selected for all remaining runs.

These runs were then made using patient arrivals as estimated above, and gradually increasing the number of beds. Results are discussed below.

3. Results and conclusions

Percentage occupancy is the currently accepted criterion of hospital operating efficiency. Although this may well be valid in the general hospital, it would not appear to be so in the case of the specialized hospital considered, which operates in a near saturated condition.

Patient waiting time would seem to be far more critical, and significant. This is the expected time a patient requiring a bed waits for a bed to become available. Average patient waiting time is this time averaged over all arrivals. As there are many patients who do not wait at all, this turns out to be a very small number of no absolute value. It is, however, considered to be of significance as an index of system condition and capability.

In Figure (2) average waiting time and percentage occupancy are plotted against beds, (BED's) increasing from 200 to 650. The following should be noticed:

(a) Values are plotted for 2.58 x current number of arrivals. 2.58 x current number of BED's would be 602. Notice that, not surprisingly, with 602 BED's, the percentage occupancy would be as it was during the period from 1963-1968.

(b) The average waiting, however, with 602 BED's is practically zero. This indicates that the system is less saturated than previously and therefore could handle a greater load.

(c) Restoring the average waiting time to its 1963-1968 level would be done by reducing the number of BED's to 530.

(c) With 530 BED's percentage occupancy is raised to 88%.

This indicates that with no increase in scheduling effort or handling efficiency, a

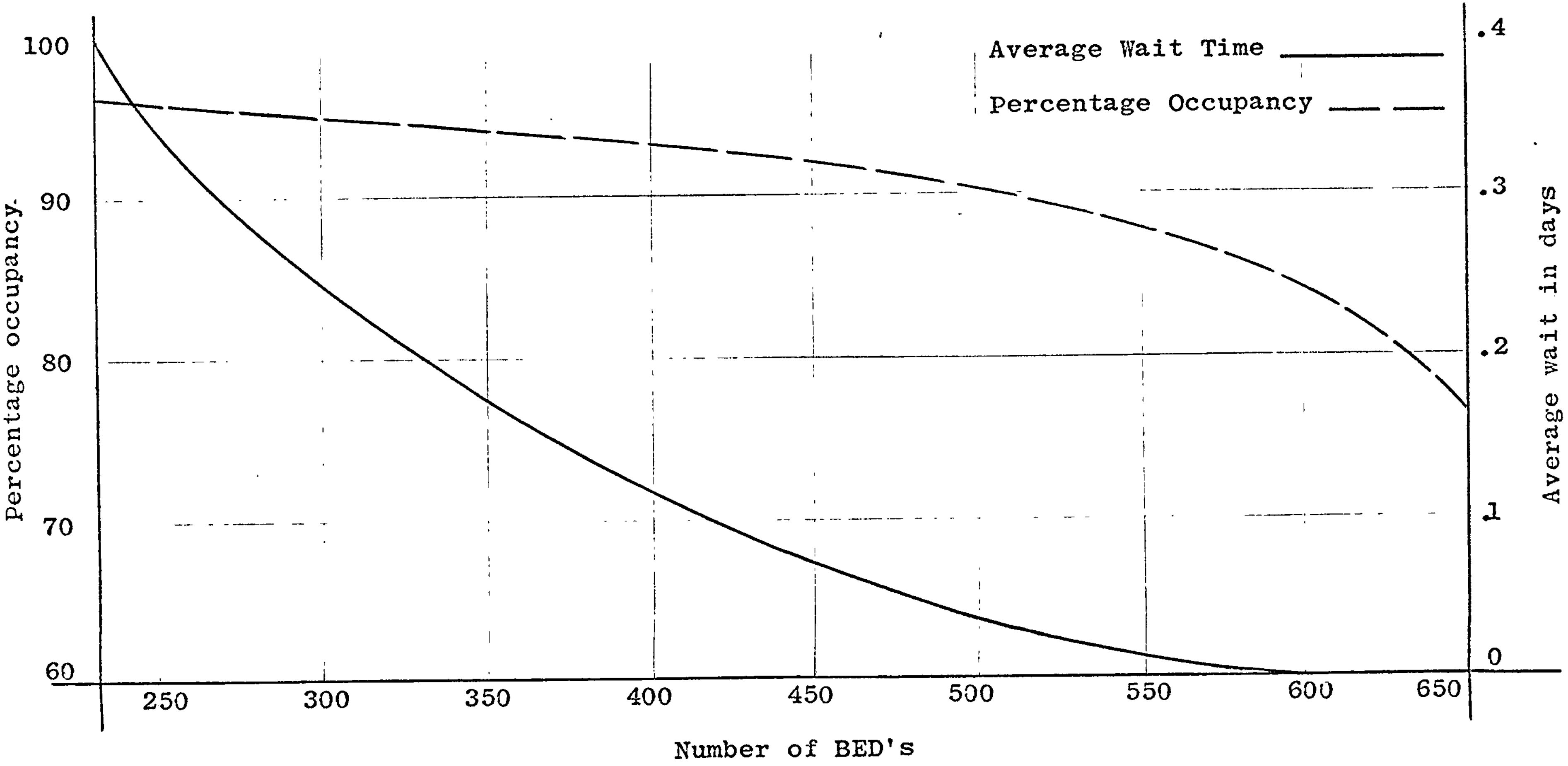


FIGURE (2)

DECREASE IN PERCENTAGE OCCUPANCY AND AVERAGE WAIT TIME

WITH INCREASING "BED" NUMBERS

(Arrivals increased to 1,410)

larger hospital is inherently more efficient than a smaller one in that it can operate at a higher percentage occupancy. This does of course assume that all bed services increase with beds, i.e., if there was one nurse for every ten beds previously there would still be one nurse for every ten beds in the enlarged hospital.

Using the output from SEA in this manner, it is easy to confidently recommend that, in order to handle the increased patients of the future, the following bed quantities be used:

(a) 550 BED's for 1,410 general patients per month

(b) 24 PBD's for 37 pediatric patients per month

(c) 18 RBD's for 23 patients per month on research

This application demonstrates the usefulness of SEA. A real problem, impossible to solve analytically, was succesfully handled by SEA. Data input was completely straightforward and computing time averaged about four minutes per situation tested. In addition the results should be very accurate because the model was initially run simulating existing conditions and agreement was excellent.

VI CONCLUSION

This paper should serve as a complete guide to anyone needing to use SEA. As it can be learned in a matter of hours by non-computer oriented personnel, it should be of great value in many fields solving tedious and often analytically insoluble problems.

Probably it will be of greatest worth to industrial engineers unfamiliar with the other more complex and powerful simulation languages. All that is needed is the ability to punch data onto IBM cards.

It is hoped that it will be put to some use and prove of value in as many fields as possible.

BIBLIOGRAPHY

1. Donaghey, C. E., A Generalized Material Handling Simulation System. Pittsburgh, Penn.: The University of Pittsburgh, 1968.

2. Elmaghraby, Salah E., The Design of Production Systems. New York, NY: Reinhold, 1966.

3. Meyer, P. L., Introductory Probability and Statistical Applications. Reading, Mass: Addison-Wesley, 1966.

4. Smalley, H. E., Freeman, J. R., Hospital Industrial Engineering. New York, NY: Reinhold, 1966.

5. Spiegel, M. R., Theory and Problems of Statistics. New York, NY: Schaum, 1961.