Examples of Discrete System Simulation: ①

* A CHECKOUT COUNTER PROBLEM :-

Consider the checkout procedure at supermarket. After customers have selected the items they wish to purchase, they then proceed to the checkout counters. Except during the busiest periods of the day, some of these counters are not staffed. There always seems to be just enough checkers so that a customer must wait in line for a while before receiving service. The reason for this, in many cases, is a store policy that if the checker has no one to serve, the counter is closed and the checker leaves the area. When the waiting lines reach a certain size, the checker returns and opens a counter.

From the customer's point of view, this policy is an annoyance since it means they will, almost every time, have to wait in line before checking out. But from the point of view of the store's mgt, it is a desirable policy because if a customer does not have to wait to check out, then for some period of time before that customer arrives at the counter, the checker was not doing any useful work. The mgt. wants to make effective use of the checker's time, either at the checkout counter or in some other activity.

Assume that a manager of a store which has operated with this policy wishes to consider a change. She would like to reduce the customer's waiting time and needs to know how much it would cost in terms of increased checkers' time at the customers. This manager believes that reducing customer waiting time will ultimately result in more customers.

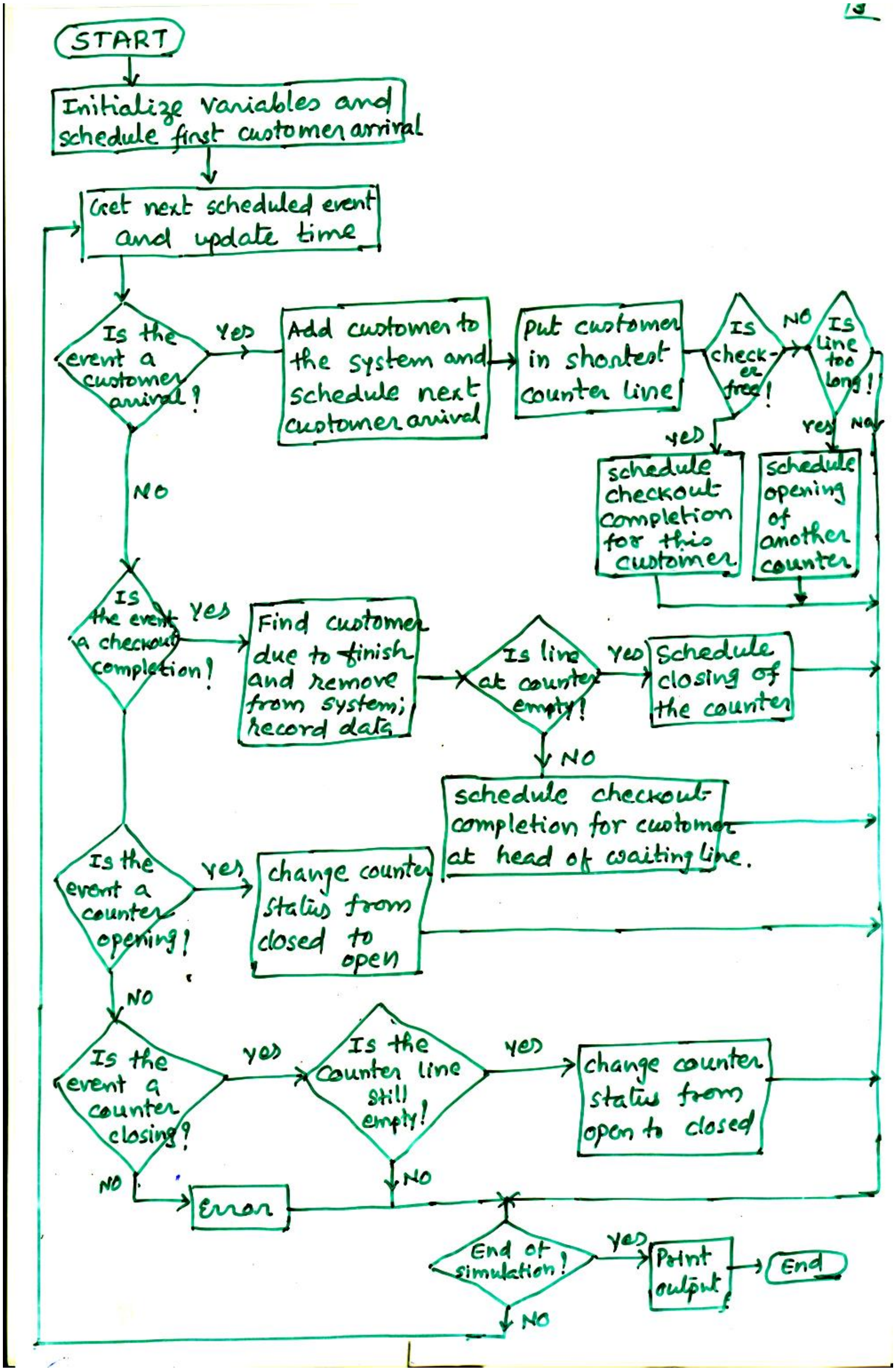
she also fears that if she improves service for a short trial period and then, because costs are too high, must go back to the old policy, there is a danger that customers may be lost. Rather than risk this, the manager wishes to determine the effect of a change in policy without changing the actual operation.

We will simulate this process of a customer entering the checkout line, waiting if necessary, and being checked out. We also simulate the activities of the checkers. ie. checking out customers, opening and closing the counters, and is to compute the time checkers spend away from the counters.

The flowchart for a computer program to simulate the checkout operation shown in figure:

It is an example of the discrete-event scheduling method in which the program executes the actions that occur when a particular event which alters the status of the systems takes place.

In the problems the scheduled events are:
(i) customer arrives (ii) customer finishes checkout, (iii) a counter opens, and (iv) a counter closes. There are other events which occur in the process, but they occur as the results of and at the same time as the scheduled events. Counters are scheduled to be opened when the size of the line at the already-open counters exceeds some specified value, but the opening actually occurs after a time delay. Similarly, counter closings are scheduled when a line becomes empty, but the closing occurs after a time delay.


START
Initialize variables and schedule first customer arrival
Get next scheduled event and update time
Is the event a customer arrival?
Yes
Add customer to the system and schedule next customer arrival
put customer in shortest counter line
Is checker free?
NO
Is line too long?
yes
No
yes
schedule checkout completion for this customer
schedule opening of another counter
NO
Is the event a checkout completion?
Yes
Find customer due to finish and remove from system; record data
Is line at counter empty?
Yes
Schedule closing of the counter
NO
schedule checkout completion for customer at head of waiting line.
Is the event a counter opening?
yes
change counter status from closed to open
NO
Is the event a counter closing?
yes
Is the Counter line still empty?
yes
change counter status from open to closed
NO
Error
NO
End of simulation?
yes
Print output
End
NO

If a customer arrives at the counter before this delay is over, the customer receives service and the scheduled closing is cancelled.

The significance of these results is dependent upon a subjective evaluation by the management of the store. A decision to change the store policy will be made on the basis of an estimate as to what effect this would have on customer behavior in response to this change. This was not part of the simulation model, so far this case there is very little to be gained by determining precise values for simulation results.

The approach used to study the checkout-counter problem can be extended to a wide variety of systems. It is limited, however, to what are often called DISCRETE-EVENT SYSTEMS. These systems are represented by some set of data, called the system STATE, which remains unchanged until some EVENT occurs which causes a discrete change in the state. The state contains all the information required to characterize the system at one point in time.

The state representation for the checkout counter are : cust. no, Arrival time, counter no, position in line, Time due to complete, Time waiting ended.

Note that changes in the state occurred with no elapsed time because of the manner in which we choose to model the system, not because customer movement or counter changes are physically instantaneous.