The Hong Kong Polytechnic University
Department of Applied Mathematics
AMA488 Simulation

Project
2018/19 Semester 1

Due date: Dec 7, 2018 (Friday) 5:30 p.m.

This project counts towards 15% of the final marks.
You may work in a group of **at most 5** members (The "group" can consist of 1 member).

Background

To transform raw cotton into a completed order, a four-stage process is required in a cotton textile mill. The four stages are spinning, weaving, finishing and packaging. The order of processing is represented by the following diagram:

Raw Cotton ➔ Spinning ➔ Weaving ➔ Finishing ➔ Packing ➔ Final Product

All the processes are done by machines.

The process time to spin sufficient yarn to produce a bale of finished cloth has a normal distribution with mean 240 and variance 120. The process time for the weaving stage follows a normal distribution with mean 480 and variance 200. The process times for the finishing stage and the packing stage follow exponential distributions with means 120 and 360, respectively. **All time unit expressed in seconds.**

Due to a lack of queueing spaces inside the factory, the queue for spinning is initially set to 25 units and one unit will be added to it when one unit of final product is produced.

The mill operates continuous production on a three shifts a day, seven days a week basis. That is, "one shift" is equivalent to an 8-hour operation.

Problem

The management realized that if the factory is producing at its most efficient level, the overall percentage of idle time of the machines has to be minimized. The management has decided that the maximum number of machines at each stage is 10, and wishes to know the number of machines, subject to this constraint, that the factory should use at each stage in order to minimize the idle time. In addition, the management wishes to assess the queueing costs for the units waiting in the queues, for the optimal solution obtained based on minimizing idleness.

As a consultant of the factory, you decide to develop a computer simulation model to solve the problem.

**** Specific instructions/requirements will be announced accordingly ****

Identify the states that the machines and queueing entities can be in at any time

In this case study, the queueing entities can be in one of the following three states:

- Waiting in a queue
- Being processed
- Not waiting in a queue nor being processed (either before arrival or fully processed)

For the machines, they can be in one of the following two states:

- Busy, providing the service
- Idle

Mechanics of constructing and executing the model

The first step in constructing a simulation model is to express the real system in terms of its key events. An event is defined as a point in time at which changes in the characteristics of the system take place. This normally happens when one or more activities are terminated and new ones begun. In order to gather the desired output information from the simulation, it would thus suffice to observe the system at the instances when events take place.

In a queueing simulation model, usually we are concerned with two events:

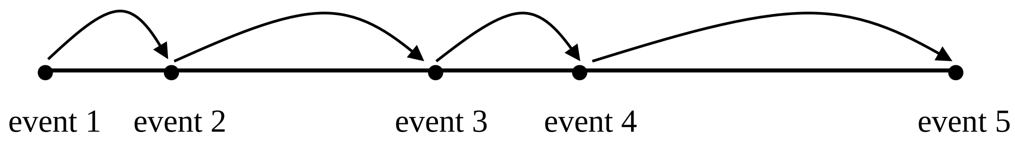
- A queueing entity arrives
- A service is completed

In this case study, when a queueing entity arrives, it can either join a queue or be processed (if any machine is available). On the other hand, when a service is completed, a unit in the queue can start to be processed; or if no one is waiting, the machine becomes idle. We can gather the necessary information by observing the various conditions that arise with the occurrence of each of the two events. For example, we can keep track of the length of the queue as follows: When an arrival occurs, the queue length is incremented by one whenever the facility is found busy. Similarly, the queue length is decreased by one when a service is completed and the waiting line is not empty.

We now show how a typical simulation model is executed. Suppose that we wish to simulate T time units of a system's operation. The simulator starts from a zero datum and locates the proper events on the time scale in chronological order. The execution of the model is achieved by "jumping" from one event to an immediately succeeding one. At each event, records are updated to reflect possible changes in the measure of performance.

Therefore, when we want to find out what the next event is, we need to find out which event gets the earliest time. In this case study, any arrival is governed by whether a service is completed, so we are only interested in the time when a service is completed. At any point of time, we also need to find whether there are any machines have been idle since last events. If there are any, the idle time of the machines are incremented by the period between last and present event.

The following diagram represents the execution process. Each event is represented by a "●" on the time scale.



The fact that the simulator "jumps" from one event to the next event shows that the execution process occurs at discrete points in time, giving rise to the name discrete (event-type) simulation.

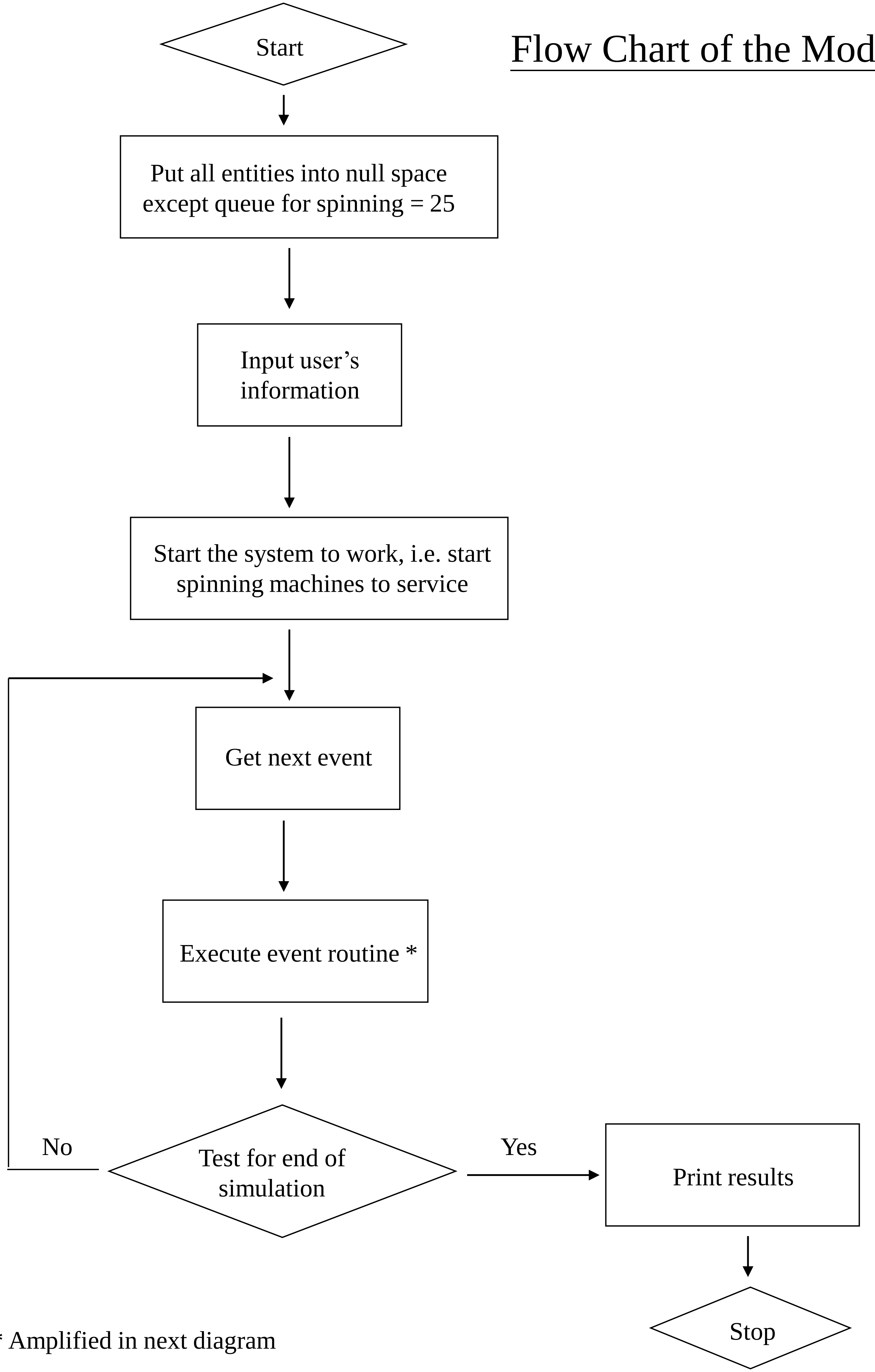
By the very nature of discrete simulation, we note that there is actually no relationship between the real time and the execution time of the model. Since the simulator is executed by a digital computer. The actual execution time is usually much smaller than the real time.

Assessment of queueing costs

The queueing costs are 100, 200, 300 and 400, in dollars per unit per unit time, for the four processes, respectively.

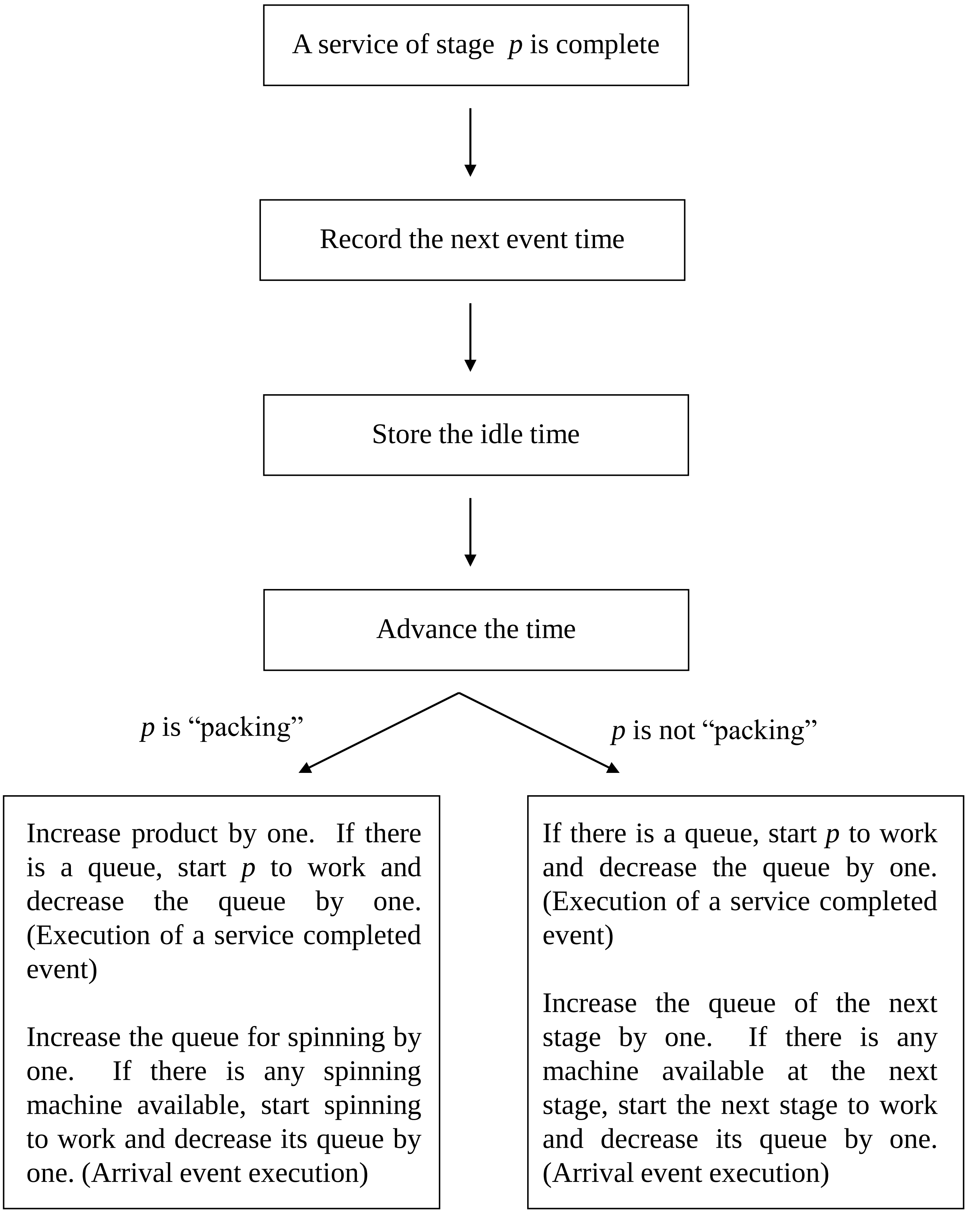
For example, if in the optimal solution the average queue length in the weaving process is 2.3 units, the average queueing cost for this process is found as: 200(2.3) in dollars per unit time. Perform similar calculations for the other three processes, and sum these costs to find the overall average queueing cost for the optimal solution.

Flow Chart of the Model

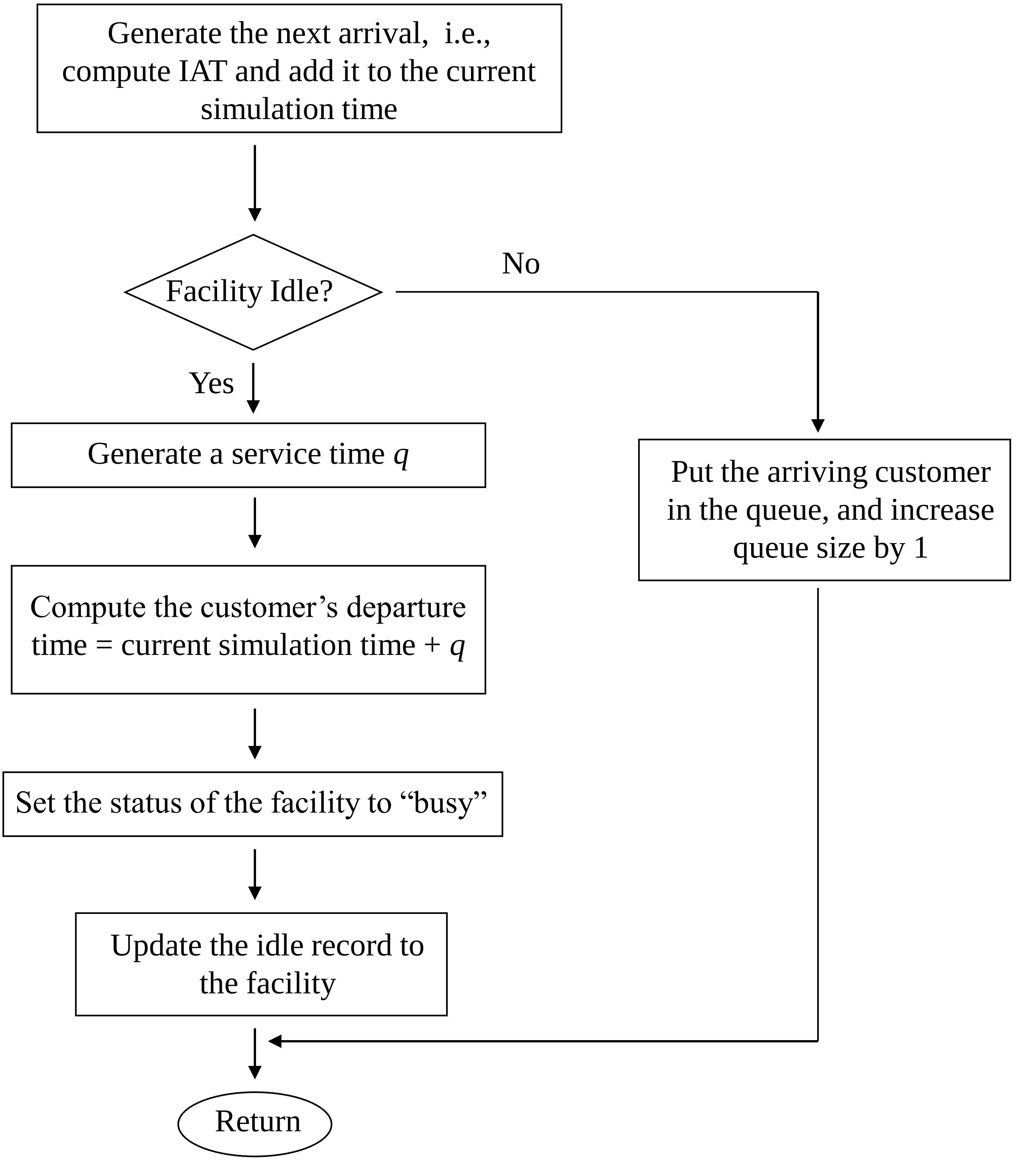


* Amplified in next diagram

Execute Event Routine



Arrival Event



Departure Event