Optimization Theory and Methods

2025 Autumn

同济经管
TONGJI SEM

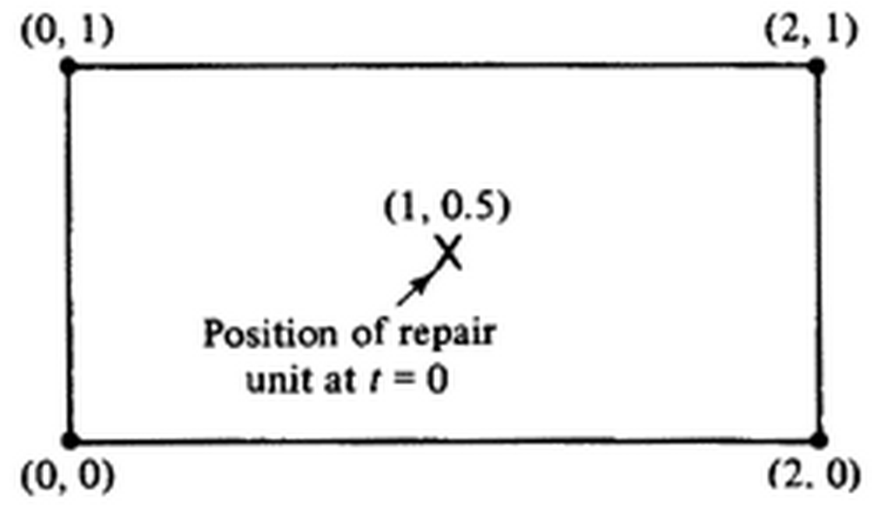
魏可佶 kejiwei@tongji.edu.cn
https://kejiwei.github.io/

Chapter 17 Simulation Case Study: Emergency Response

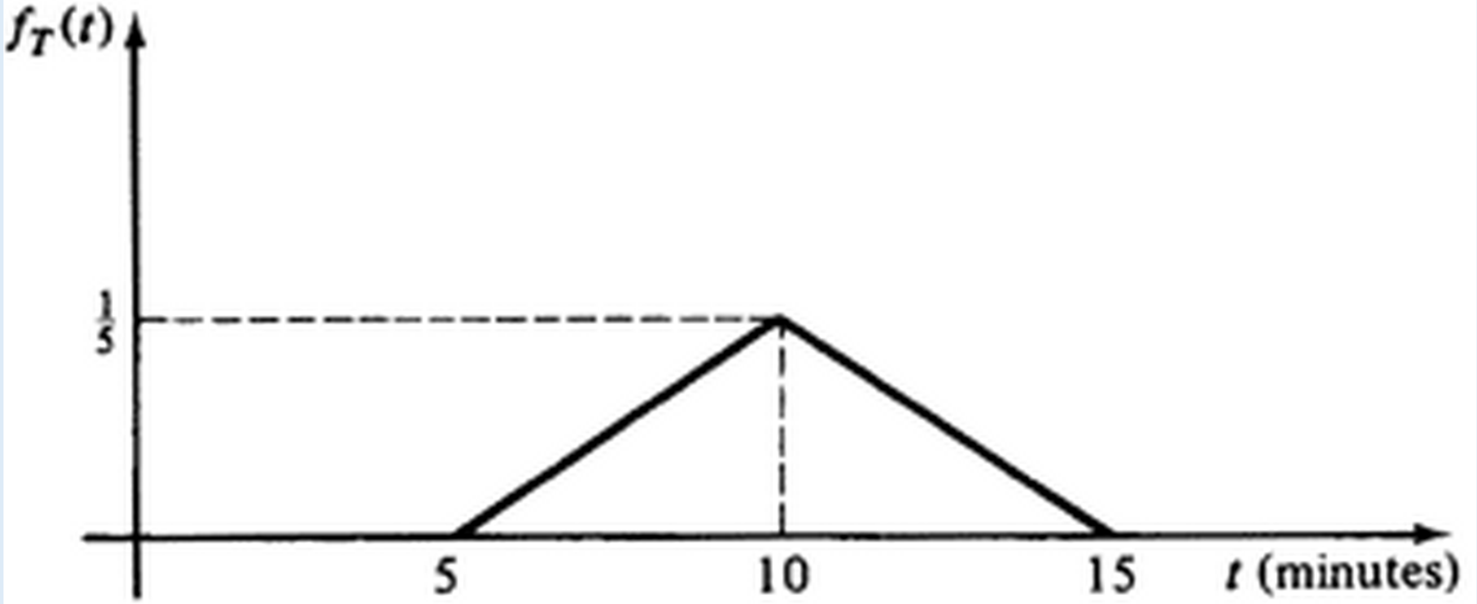
↳ Emergency response vehicle service

- Consider an emergency response vehicle (e.g. ambulance, fire engine, police car, etc.) serving a certain part of a city.
- Whenever there is no outstanding call for service, the vehicle is positioned at its home location.
- When the vehicle receives a call, it drives to the call location, provides necessary service and then reassesses whether there is another call waiting to be serviced.
- If there is no other call then it again positions itself at its home location.
- If there is one or more calls waiting when the vehicle completes a service, then it drives to the nearest call location (rather than to the location of the first call that came in).

- For simplicity, assume that the part of the city that is served by this emergency vehicle (the service region) is a rectangle with vertices at $(0,0)$, $(2,0)$, $(2,1)$, and $(0,1)$.
- Assume the home location to be at the geometric center of the rectangle, which is at $(1,0.5)$. All distances are in miles.
- Let the direction of travel be parallel to the rectangle's sides. So, distance between points (x_1, y_1) and (x_2, y_2) is given by $|x_1 - x_2| + |y_1 - y_2|$.



- Let the speed of the emergency vehicle be 20 mph.
- Let the call locations be independent and uniformly distributed over the rectangle.
- Let the calls be generated as per a Poisson process with rate $\lambda = 3$ per hour and the amount of service time spent on call location be given by the following PDF.



- Total service time of a call equals the time to travel to the call location + actual time spent on call location.
- Note that the next call to be served is the one that is closest from the previous call location.
- So the longer the service time of the previous call, the greater the number of outstanding calls waiting at the end of that service, and so the more likely it is to have a call in close vicinity of the previous call.
- Therefore, when service time is longer, the next service time is likely to be shorter.
- So consecutive service times are DEPENDENT.
- Thus, analytical solutions are extremely difficult to find.
- We have to resort to event-driven simulation.

- Let us plan to simulate the system for a period of 12 hours.
- We wish to collect data on
  - average wait time between a call and the vehicle arrival on call location (response time)
  - maximum observed queue length
  - average length of a busy period
  - variance of total service times, etc.
- Assume that at time $t = 0$, there are no calls in the system and the vehicle is located at home location $(1{,}0.5)$.
- We will measure time in minutes and distance in miles.
- We will need to keep track of several important variables as we proceed through the simulation. Let us define them first.

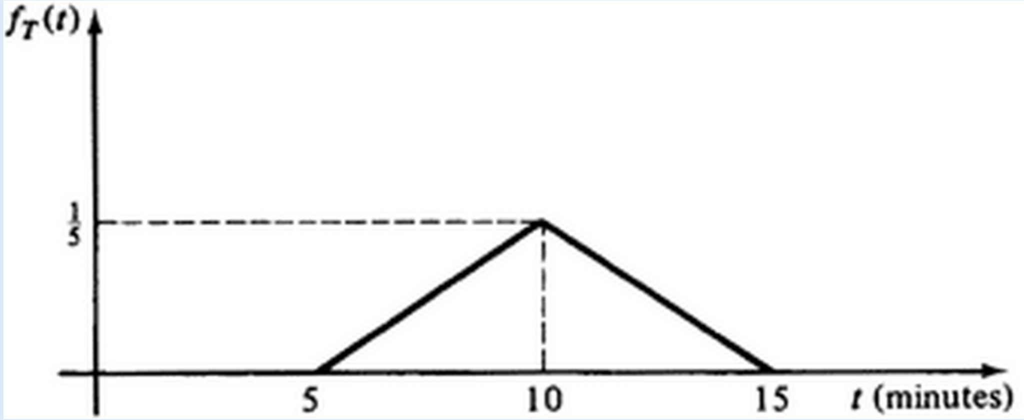
- Time variables:
  - t: Variable indicating the current time in the simulation.
  - t_{nc}: Time of the reception of the next call.
  - t_{ss}: Time when service to a call starts.
  - t_{sf}: Time when service to a call finishes.
  - t_l: Time when the simulation ends (last time). This is set to 12 hours or 720 minutes, as explained earlier.
- Indicator variables:
  - I: Number of calls in the queue.
  - $N = \begin{cases} 0, & if\ the\ vehicle\ is\ idle \\ 1, & if\ the\ vehicle\ is\ busy \end{cases}.$

- Variable j denotes the calls in the order in which they are received.
- t_j: Time of arrival of call j.
- (x_j, y_j): Location of call j.
- (x_0, y_0): Current location of the vehicle.
- τ: Travel time to reach a call from current location of the vehicle to the call location at 20 mph. $\tau = 3(|x_j - x_0| + |y_j - y_0|)$.
- Note that we are assuming the travel to happen along vertical and horizontal grid-lines. This is also sometimes called as Manhattan distance or the $L1$ norm.

- Let us simplify the random number generation process by considering two separate series of random numbers:
  - $r_{1,k}$ for $(k = 1,2,3,\ldots)$: For simulating probabilistic quantities related to time. These include times between consecutive calls for service and service times on call location.
  - $r_{2,m}$ for $(m = 1,2,3,\ldots)$: For simulating probabilistic quantities related to space. These include the X and Y coordinates of the call location.

- If the next random variable to be drawn is the time between consecutive call arrivals, then it is generated using the next uniform random variable from the first series $(r_{1,k})$ as $\frac{-\ln(r_{1,k})}{\lambda} = \frac{-\ln(r_{1,k})}{1/20} = -20\ln(r_{1,k})$. (Note: All times are measured in minutes).
- The second series of random numbers $r_{2,m}$ is used for generating the X and Y coordinates of the next call location.
- $X \sim U[0,2]$ and $Y \sim U[0,1]$. So the two numbers are generated as $X = 2 * r_{2,m}$ and $Y = r_{2,m+1}$.

- If the next random variable to be drawn is the service time on call location, then a convenient way of simulating it is to use the next two random numbers from the first series, namely, $r_{1,k}$ and $r_{1,k+1}$. We use the relationships method here.
- A Useful Fact: A triangular distribution is obtained for a random variable that is the sum of two independent uniformly distributed random variables (Can you guess their ranges?).



- $\mathrm{T} = 5 + V_1 + V_2$ where $V_1, V_2 \sim U[0,5]$ and V_1, V_2 are independent of each other.
- So the time spent on call location is generated as $5\left(1 + r_{1,k} + r_{1,k+1}\right)$.

同济经管 TONGJI SEM

- All distances are measured in miles and times in minutes.
- **Clock starts ($t = 0$)** when there are no calls in the system and the vehicle is located at home location $(1,0.5)$.
- We plan to simulate the system for a period of 720 minutes.
- There are four **types of events** that can occur:

1. A new call for service is received (at t_{nc}).
2. Service to a call starts (at t_{ss}).
3. Service to a call finishes (at t_{sf}).
4. Simulation ends (at t_l).

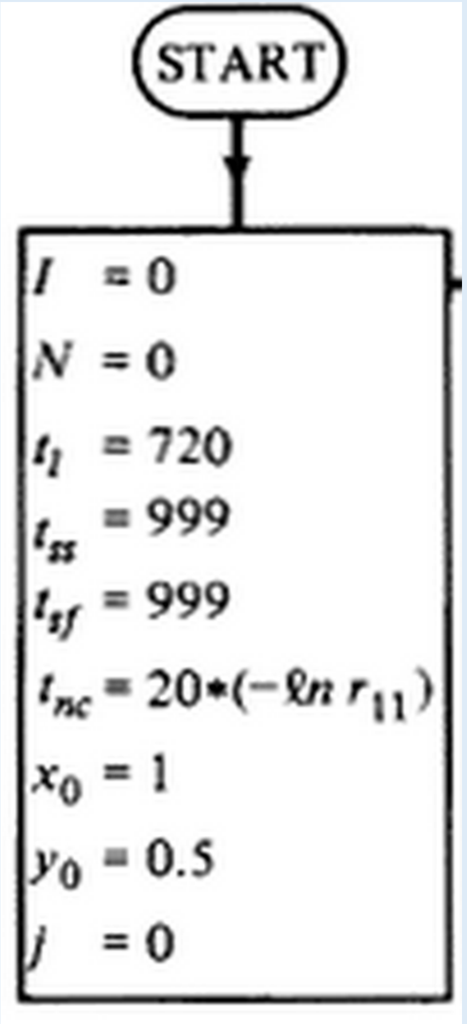
CAMEA 中国高质量MBA教育认证 AACSB ACCREDITED EQUIS ACCREDITED

↳ Next-event time advance (cont.)

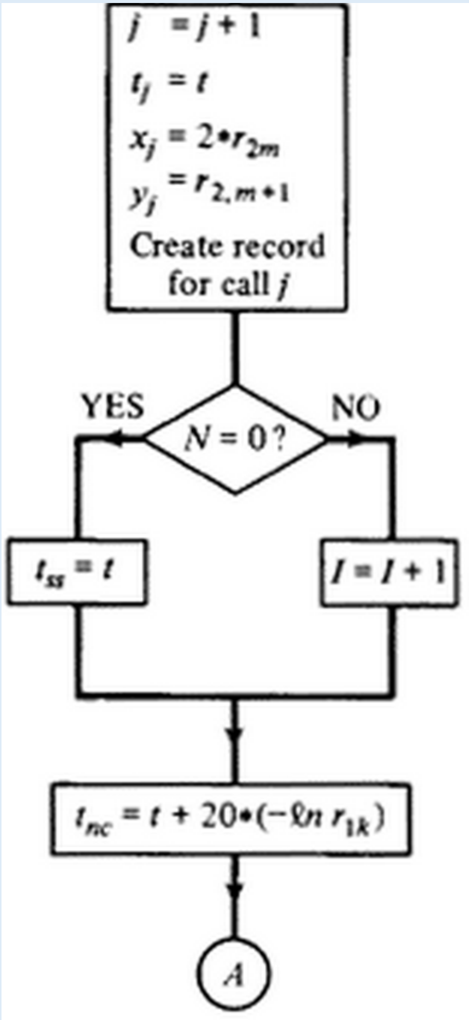
- Throughout the period of simulation, the program repeatedly identifies what event will happen next and responds to it.
- Thus, there are 5 main parts of the simulation program: initialization and response routines for each type of event.

↳ Initialization

- Initialize the call queue length $I = 0$.
- Initialize N to 0 indicating that the vehicle is idle.
- Simulation end time t_l is initialized to 720. It will stay constant at that value for the entire simulation.
- t_{ss} and t_{sf} are initialized to large enough values (999). Any value > 720 would work.
- Call index j is initialized to be 0.
- Arrival time of the first call t_{nc} is simulated from exponential distribution as $-20\ln(r_{1,1})$.
- Starting position of the vehicle is initialized at center of the region. So $x_0 = 1$, and $y_0 = 0.5$

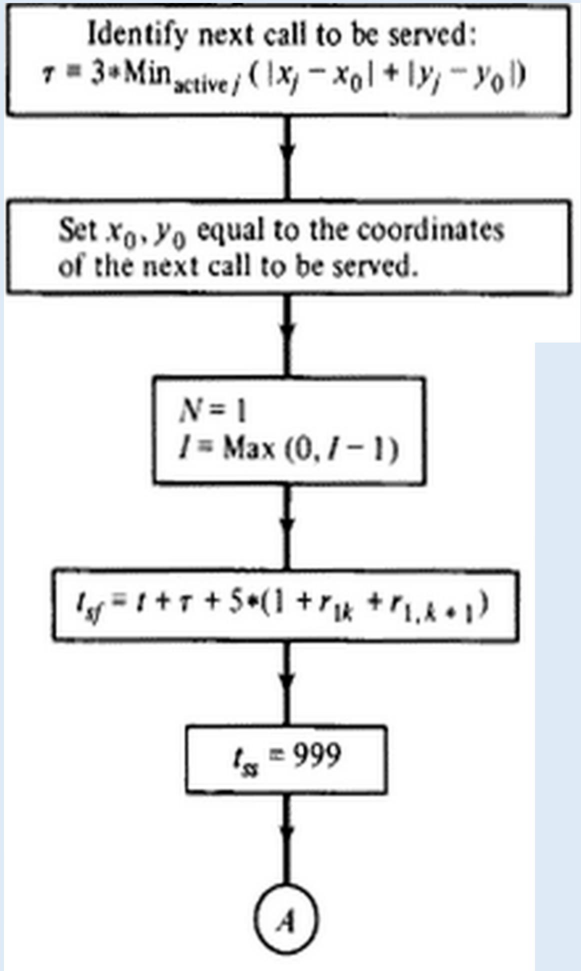


- When new call is received, call index is increased by 1, i.e., $j = j + 1$.
- The time of the call is set to current time $t_j = t$.
- Call location is simulated as $x_j = 2r_{2,m}$ and $y_j = r_{2,m+1}$.
- Add this call to the list of active calls.
- Consider two cases:
  - If the vehicle was idle when the call came in then set its service start time t_{ss} to t.
  - Else, add it to queue by increasing queue length I by 1.
- Finally, simulate the next call time. This has to be done right away because we need to decide which event to attend to next.



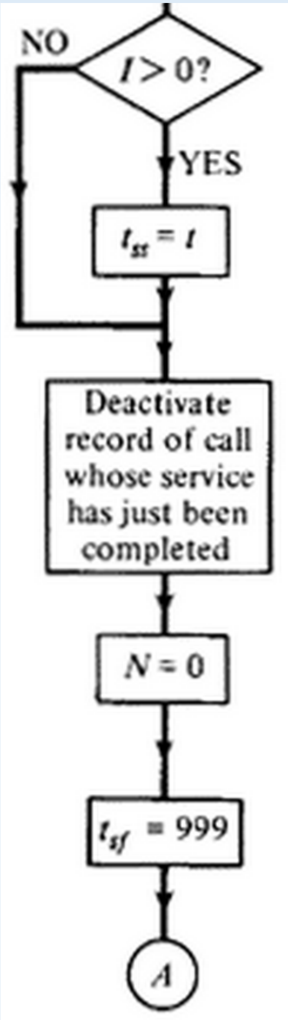
↳ Service to a Call Finishes

- We need to keep track of all active calls, defined as the calls that have arrived but haven't yet been served.
- That way we can identify the next call to be served as the call that is nearest from current position of the vehicle (denoted by (x_0, y_0)) from the list of active calls. So $\tau = 3 * Min_{active\ j}\left(\left|x_j - x_0\right| + \left|y_j - y_0\right|\right)$.
- Set (x_0, y_0) to the coordinates of this call.
- Set $N = 1$ and $I = I - 1$ indicating that server is busy and queue length is decreased by 1.
- Simulate service time. Set call completion time = current time + travel time + service time. $t_{sf} = t + \tau + 5\left(1 + r_{1,k} + r_{1,k+1}\right)$.
- Set t_{ss} to a large value so next event is not a service start.

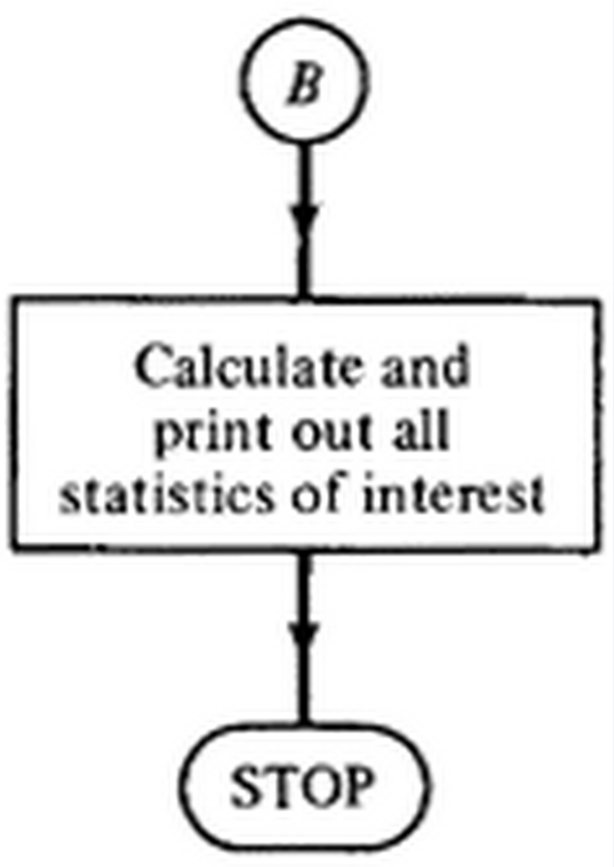


↳ Service to a Call Finishes

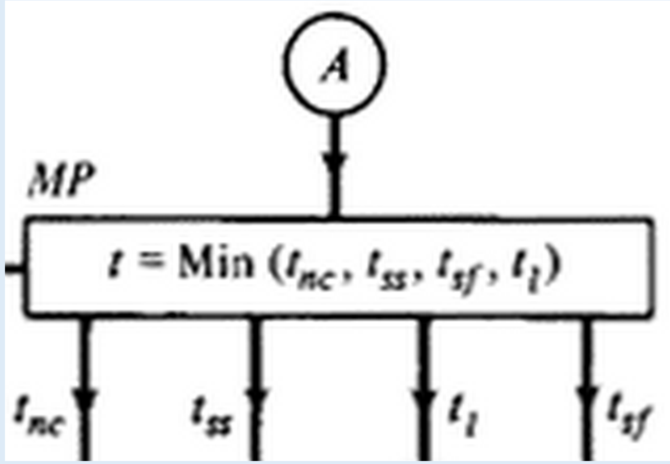
- If there is another call in the queue then set the next service start time to current time.
- Move the call, whose service has just been completed, out of the active call list.
- Set $N = 0$, i.e. the server is set to be idle.
- t_{sf} is set to a large enough value so that the next event will not be another service completion.

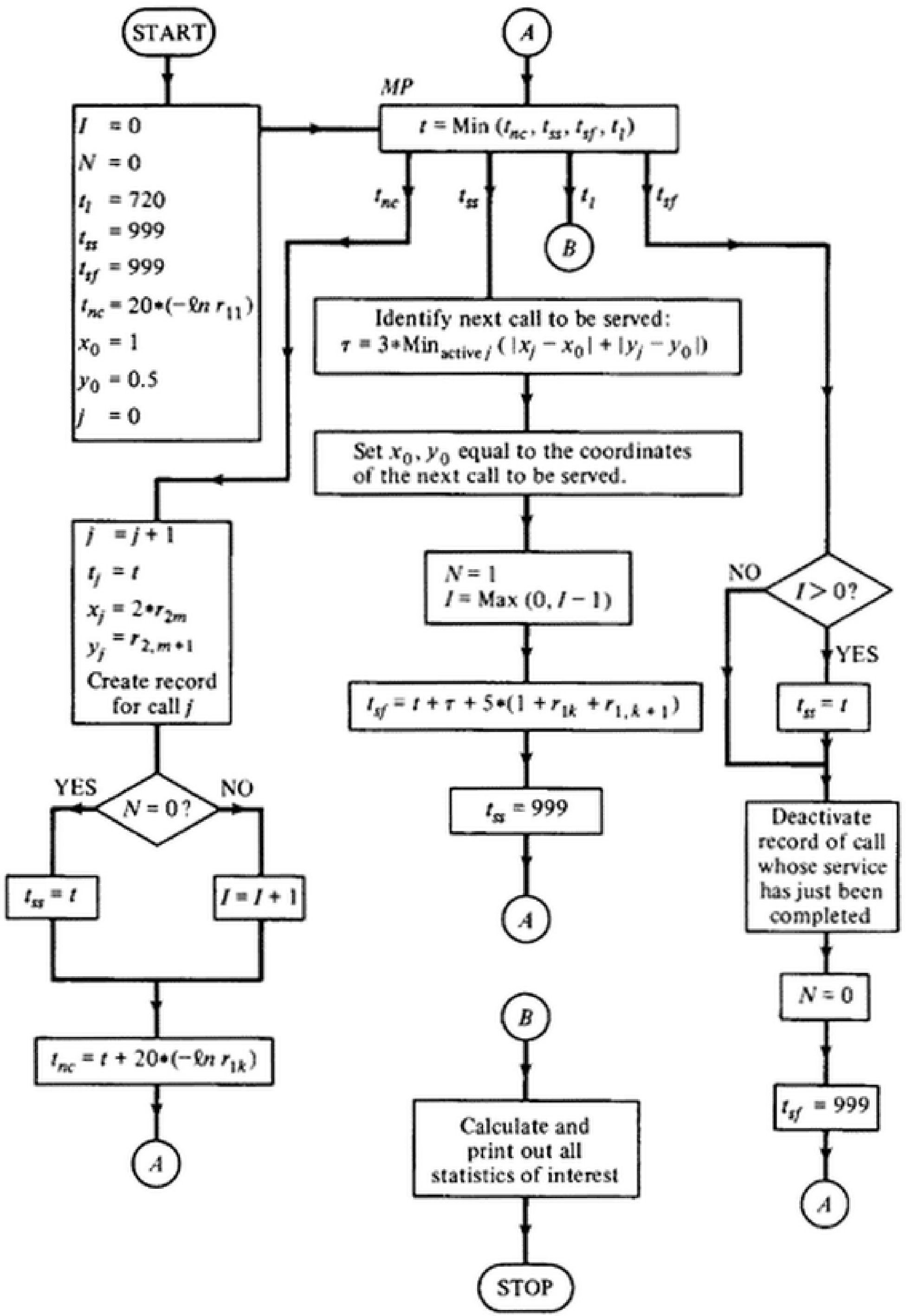


- Calculate all relevant statistics, such as:
  1) Average wait time between a call and the vehicle arrival on call location.
  2) Maximum observed queue length.
  3) Average length of a busy period.
  4) Variance of total service times, etc.
- Print all relevant outputs to screen or to a file.
- End the simulation program.



- The only thing that remains to be done to put everything together is to decide to which of the four types does the next event belong.
- This is done by simply comparing the times when the next event of each type will happen and finding the minimum of those four time values. The event type corresponding to this minimum time value will be the type of the next event that will happen.
- This is how the flow of the overall program is governed.
- Next slide shows the overall algorithm.




START
I = 0
N = 0
t_l = 720
t_ss = 999
t_sf = 999
t_nc = 20*(−ℓn r_11)
x_0 = 1
y_0 = 0.5
j = 0
A
MP
t = Min (t_nc, t_ss, t_sf, t_l)
t_nc
t_ss
t_l
t_sf
B
Identify next call to be served:
τ = 3*Min_active j (|x_j − x_0| + |y_j − y_0|)
Set x_0, y_0 equal to the coordinates of the next call to be served.
N = 1
I = Max (0, I − 1)
t_sf = t + τ + 5*(1 + r_1k + r_1,k+1)
t_ss = 999
A
j = j + 1
t_j = t
x_j = 2*r_2m
y_j = r_2,m+1
Create record for call j
YES
N = 0?
NO
t_ss = t
I = I + 1
t_nc = t + 20*(−ℓn r_1k)
A
NO
I > 0?
YES
t_ss = t
Deactivate record of call whose service has just been completed
N = 0
t_sf = 999
A
B
Calculate and print out all statistics of interest
STOP

Objective :

Key Concepts :