Simulation of a Dry-Cleaner's Two-Piece Suit Processing System

This project presents a discrete-event simulation model of a multi-stage dry-cleaning system for two-piece suits, where garments are processed, split into components, and reassembled post-cleaning. The simulation captures not only the operational dynamics of such a system but also probabilistic failures (damage) and customer escalation workflows. Using the simlib library in C, the model evaluates system performance under both normal and high-load scenarios, guiding decisions on resource allocation and process improvement.

Section 1: Problem Statement

Two-piece suits are processed by a dry cleaner as follows. Suits arrive with exponential inter-arrival times having a mean of 10 minutes, and are all initially served by server 1, perhaps after a wait in a FIFO queue; (see Fig.1.1). Upon completion of service at server 1, one piece of the suit (the jacket) goes to server 2, and the other part (the pants) to server 3. During service at server 2, the jacket has a probability of 0.05 of being damaged, and while at server 3 the probability of a pair of pants being damaged is 0.10. Upon leaving server 2, the jackets go into a queue for server 4; upon leaving server 3, the pants go into a different queue for server 4. Server 4 matches and re-assembles suit parts, initiating this when he is idle and two parts from the same suit are available. If both parts of the reassembled suit are undamaged, the suit is returned to the customer. If either (or both) of the parts is (are) damaged, the suit goes to customer relations (server 5). Assume that all service times are exponential, with the following means (in minutes) and use the indicated stream assignments:

Server Number	Mean service time, in minutes	Stream
1	6	1
2	4	2
3	5	3
4	5 (undamaged)	4
4	8 (damaged)	5
5	12	6

In addition, use stream 7 for inter-arrival times, and streams 8 and 9 for determining whether the pieces are damaged at servers 2 and 3, respectively. The system is initially empty and idle, and runs for exactly 12 hours. Observe the average and maximum time in the system for each type of outcome (damaged or not), separately, the average and maximum length of each queue, and the utilization of each server. What would happen if the arrival rate were to double (i.e., the inter-arrival time mean were 5 minutes instead of 10 minutes)? In this case, if you could place another person anywhere in the system to help out with one of the 5 tasks, where should it be?

Section 2: System Analysis

The dry-cleaning system under consideration is a multi-phase queuing and service network, geared toward processing two-piece suits. The system models not only the sequential operations of garment cleaning, but also the dynamic routing and reassembly of suit components under some stochastic conditions.

Each arriving suit is composed of two independent garments, a jacket and a pair of pants, both of which must be processed separately after initial intake, and later reassembled. The flow of pieces through the system, is captured in Fig. 1.1, while the decision logic and control structures of key subroutines are illustrated in Flowcharts Fig. 2.1–2.5.

Suits enter the system according to a process with exponentially distributed inter-arrival times. All service times are exponentially distributed times as well. The mean inter-arrival time is 10 minutes. Each suit is uniquely identified upon entry and joins the FIFO queue for Server 1, as shown in Fig. 1.1, where it awaits initial processing.

Upon reaching Server 1, each suit is registered, tagged, and split into its components: the jacket proceeds to Server 2, and the pants proceed to Server 3. Each server may have its own queue, and both components are treated as independent service entities post-disassembly.

Server 2 services jackets with a mean time of 4 minutes. There is a 5% chance a jacket is damaged during this stage. Server 3 services pants with a mean time of 5 minutes. There is a 10% chance a pair of pants is damaged.

After service completion, jackets and pants are directed to separate holding queues for reassembly. Server 4 monitors both queues and begins reassembly only when it is idle and when both components of a suit with matching suit IDs are available. If neither part is damaged, the suit is returned to the customer. If either part is damaged, the suit is rerouted to Server 5 (Customer Relations). Processing times for Server 4 are mean 5 minutes for undamaged suits, and mean 8 minutes for damaged suits. Server 5 handles damaged suits with a mean service time of 12 minutes.

Random number streams are assigned as follows:

- Streams 1,2,3,4,5,6: Service times for Servers 1–5 (server 4 has 2 queues)
- Stream 7: Inter-arrival times
- Streams 8,9: Damage determination (jacket, pants)

Each suit is tracked using three attributes: (1) arrival time, (2) suit ID, and (3) damage flag. The simulation observes system time, queue statistics, and server utilization over a 12-hour run.

Section 3: Event Graph

The events for this model are quite straightforward:

Event Description	**Event Type**
Arrival of a suit into the system	1
End of simulation (after 12 hours)	2
Departure from Server 1 (suit split into parts)	3
Departure from Server 2 (jackets processed)	4
Departure from Server 3 (pants processed)	5
Departure from Server 4 (suit reassembled)	6
Departure from Server 5 (customer relations done)	7

The event graph is shown in **Fig. 3.1**. The arrival event is scheduled at the beginning of simulation. It assigns it a unique suit ID, records the current time, and attempts to place it into service or a queue for Server 1. A new arrival event is also scheduled from here. The end-of-simulation is a terminating event, also scheduled at the very beginning of the simulation to occur at 720 minutes (12 hours). When this event occurs, no further events are processed, and the simulation ends with the generation of the performance report. Departure from Server 1 represents the moment when the intake process is completed and the suit is split into its jacket and pants. These components are sent to Servers 2 and 3, respectively. Departure from Server 2 occurs when the jacket component finishes service. At this point, a damage assessment is performed, and the jacket is routed into the reassembly queue. Departure from Server 3 similar to Type 4, but for the pants component. It follows the same pattern of service completion, damage check, and queuing for reassembly. Departure from Server 4 signifies the completion of reassembly. The system checks if the reassembled suit is undamaged (and can exit the system) or damaged (and must be forwarded to Server 5). Departure from Server 5 marks the end of customer relations handling for damaged suits. After this event, the suit exits the system entirely. The data collected here contributes to the statistics for damaged items.

We will use the following simlib list structure:

List	**Attribute 1**	**Attribute 2**	**Attribute 3**
1 through 6, queues	Time of arrival into system	Suit ID	Damage Flag
7 through 11, servers	Time of arrival into system	Suit ID	Damage Flag
25, event list	Event Time	Event Type	-

Each record in the queue and server lists carries three attributes: the T**ime of Arrival into the system**, the **Suit ID**, and a **Damage Flag**. For servers, this information is used to track utilization and to route suits based on their damage state. It is also used to project suits or suit pieces forward into respective queues. For queues, it also allows us to calculate wait times and ensure correct sequencing and matching (especially in reassembly).

Section 4: Simulation Design

At the heart of the simulation is the **main loop**, which perpetually selects the earliest pending event from the event list using the timing() function provided by simlib, executes the corresponding function, and advances the simulation time. This is captured in **Listing 5.2**. Initialization routines precede the main loop, setting up random streams, input parameters, and simulation time bounds. Simlib provides **lists**, each acting as either a queue or a server resource pool. We assigned list IDs for all queues and server stations, for instance: LIST_S1_Q is the queue for Server 1, LIST_S4_JQ and LIST_S4_PQ are for jacket and pants reassembly queues, LIST_S1, LIST_S2, etc., hold items currently being serviced. By representing **servers as lists with zero or one item**, we are able to track utilization via filest(). Server idleness is simply the absence of an item in the list. Termination is dictated by the end-simulation event.

The arrive() function schedules the next arrival and initializes suit attributes. Each suit record is embedded with three attributes: Arrival Time (ATTR_TIME), used to compute the total system time at departure; Suit ID (ATTR_SUIT_ID), critical for ensuring correct reassembly, only jacket and pants with the same ID are allowed to be recombined; Damage Flag (ATTR_DAMAGE_FLAG): A boolean (0/1) indicating if either part of the suit has sustained damage. It then proceeds to check whether Server 1 is idle. If so, the suit immediately enters service; otherwise, it is enqueued. When service at Server 1 concludes, the suit is disassembled.

The depart1() function sends the jacket and pants to their respective servers, each with a copy of the suit ID. depart2() and depart3() handle jacket and pants processing respectively, including random damage assessment, using STREAM_DAMAGE_JACKET with a 5% threshold for jacket pieces and STREAM_DAMAGE_PANTS with a 10% for pant pieces. If Server 4 is idle and a matching pair is available, the reassembly() routine is triggered.

The non-event reassembly function handles the logic for checking if both queues, LIST_S4_JQ and LIST_S4_PQ, have matching suit components that can be reassembled into a complete unit. It first checks whether both queues have at least one item and whether the suit ID of the head (front) of each list is the same. If those conditions are met, it removes the front element from both queues and checks whether either piece has been marked as damaged. The damage status is retrieved through a shared transfer array using the ATTR_DAMAGE_FLAG attribute, and if either part is damaged, the overall status is updated to reflect that. Next, the function calculates how long it will take to service (reassemble and process) the suit based on its damage status. If the suit is damaged, it uses a different exponential distribution (SERVICE_DAMAGED) for generating the service time compared to when it's undamaged (SERVICE_UNDAMAGED). Once the service time is determined, the complete suit is placed into service (via list_file into LIST_S4), and a future event is scheduled to represent the departure of this reassembled suit after the calculated service time.

The report() function aggregates statistics including mean/max system time, mean/max queue lengths, and each server's utilization.

Section 6: Code Listings

Listing 6.1 – External Definitions and Global Variables

Listing 6.2 – `main()`, simulation Loop and Initialization

Listing 6.3 – `arrive()` function, initializes suit attributes and queues into Server 1

Listing 6.4 – `depart1()` function, splits suit and sends jacket/pants to respective servers

Listing 6.5 – `depart2()` and `depart3()` functions, damage assignment, queuing into reassembly queue

Listing 6.7 – `reassembly()` logic: Matches components and triggers Server 4

Listing 6.8 – `depart4()` Function: Handles undamaged vs. damaged exit paths

Listing 6.9 – `depart5()` Function: Final processing of damaged suits

Listing 6.10 – `report()` Function: Output summary

Listing 6.11 –`log_filest()` for formatting queue data

Section 7: Simulation Output and Discussion

The simulation was conducted under two operating conditions to assess the dry-cleaning system's behavior under a normal 10 minute mean inter-arrival time and a doubled mean inter-arrival time of 5 minutes. In both runs as specified already, the system operated for 12 hours (720 minutes), beginning in an empty and idle state. Figure 7.1 shows the output file (dryclean.out) for the 10-minute inter-arrival scenario and Figure 7.2 shows the output for the 5-minute inter-arrival simulation. In the outputs, the server utilizations are depicted as proportions, and not percentages.

In the initial scenario, 84 suits were accepted into the system, of which 83 were successfully processed. This near-complete throughput indicates a well-functioning system with minimal loss or rejection. The average system time was approximately 48.48 minutes for undamaged suits and 57.55 minutes for damaged suits. The maximum system times remained modest at 91.10 minutes and 88.50 minutes, respectively, with minimum times dipping to as low as 3.85 minutes for undamaged cases. The intake queue (S1_Q) averaged 2.13 suits, peaking at 9. The jacket and pants queues before reassembly (S4_JQ and S4_PQ) averaged 1.53 and 1.12 respectively, indicating consistent but non-congested flow. Server 5's queue remained empty for the most part, reflecting the relatively low number of damaged suits (only 9 out of 84). Server utilization rates were comfortably under full capacity. Server 1, the intake point, showed the highest load at 0.737, followed by Server 4 at 0.600, which handles reassembly. Servers 2 and 3, which clean jackets and pants, reported moderate utilization of 0.414 and 0.496. Server 5, used only when garments are damaged, remained lightly burdened at 0.093. These values suggest the system is both resilient and well-balanced under standard conditions.

In the scenario of the doubled mean inter-arrival time, 157 suits were accepted, and 109 were successfully processed, revealing a notable shortfall. Approximately 30.6% of accepted suits did not complete processing within the 12-hour window, possibly indicating that a system capacity was exceeded.

System times spiked dramatically. Undamaged suits now spent an average of 170.57 minutes, while damaged suits lingered even longer at 201.12 minutes. Maximum times spent approached 277 minutes, with minimums still hovering in the 22–29 minute range, a subtle sign that a few early entries were lucky enough to dodge the rush.

A bottleneck is evident in the Server 1 queue, which ballooned to an average of 27.83 suits in queue at a time, peaking at 48. This dramatic increase cascaded down the system, as Server 2 and 3 queues more than doubled their average lengths, and the reassembly queues for jackets and pants also expanded significantly. Even Server 5, previously dormant, now exhibited an average queue presence, albeit small. Utilization data paints a limpid picture of system strain. Server 1 reached a critical saturation point at 0.993, operating nearly every moment of the simulation. Servers 3 and 4 rose to 0.722 and 0.787, respectively, while Server 2 climbed to 0.576. Server 5 showed a notable increase to 0.173, corresponding to the higher volume of damaged suits in this run.

The most significant constraint revealed under high load is the intake bottleneck at Server 1, whose utilization reached 0.993 and whose queue length ballooned to 48 suits. Since every suit must pass through Server 1 exactly once, and no part of the system can proceed without this initial processing, Server 1 becomes the controlling gate of system throughput. From simply looking at these numbers, we see that a person should apparently be added to server 1 to potentially achieve the greatest reduction in overall average queue lengths and total service times. As is often the case, however, this conclusion must be regarded as tentative, given that it rests upon the outcome of only a single simulation run for each variant of the model.

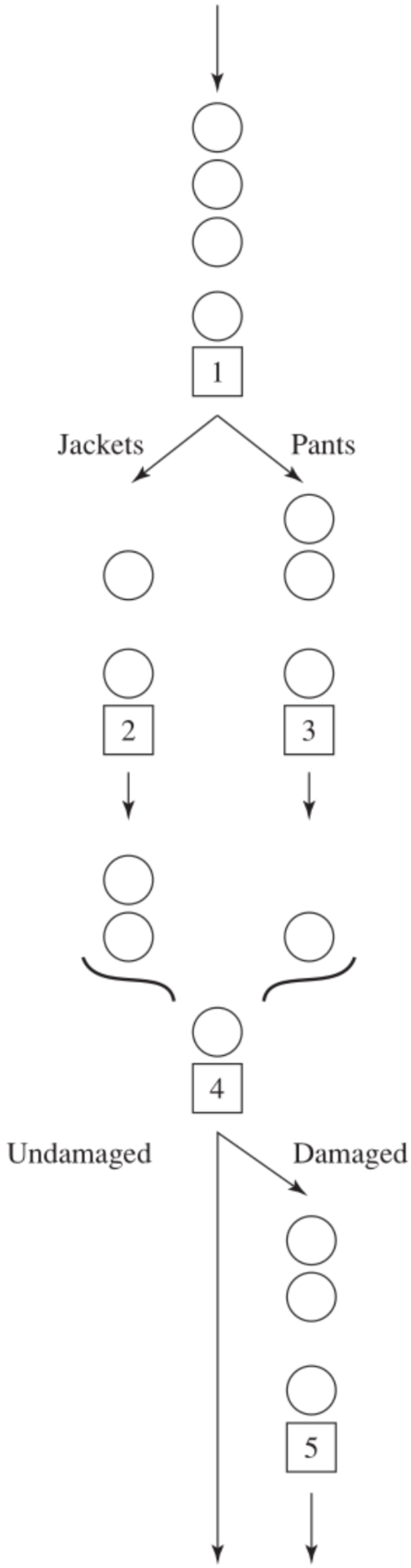


FIGURE 1.1
A dry-cleaning operation

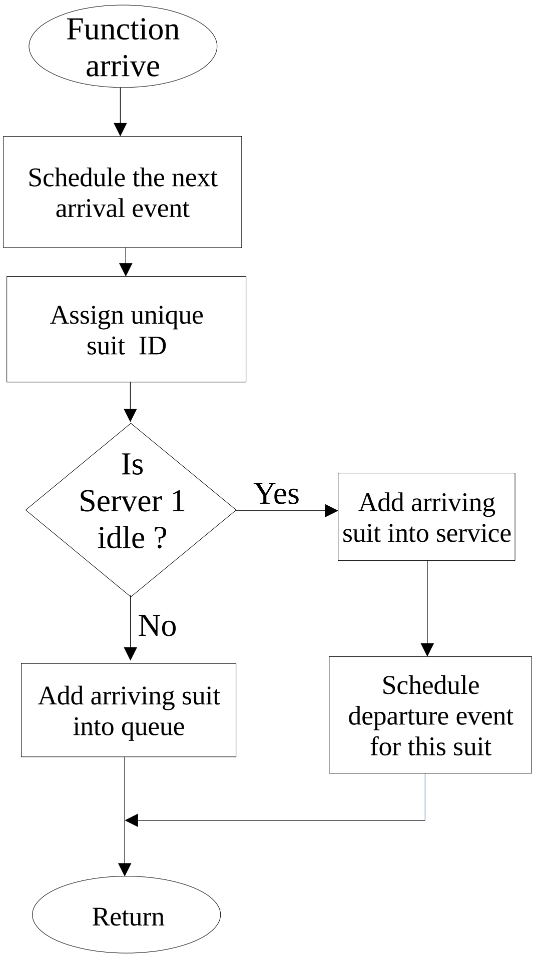

Function arrive
Schedule the next arrival event
Assign unique suit ID
Is Server 1 idle ?
Yes
Add arriving suit into service
No
Add arriving suit into queue
Schedule departure event for this suit
Return

FIGURE 2.2
Flowchart for arrive function, dry-clean model

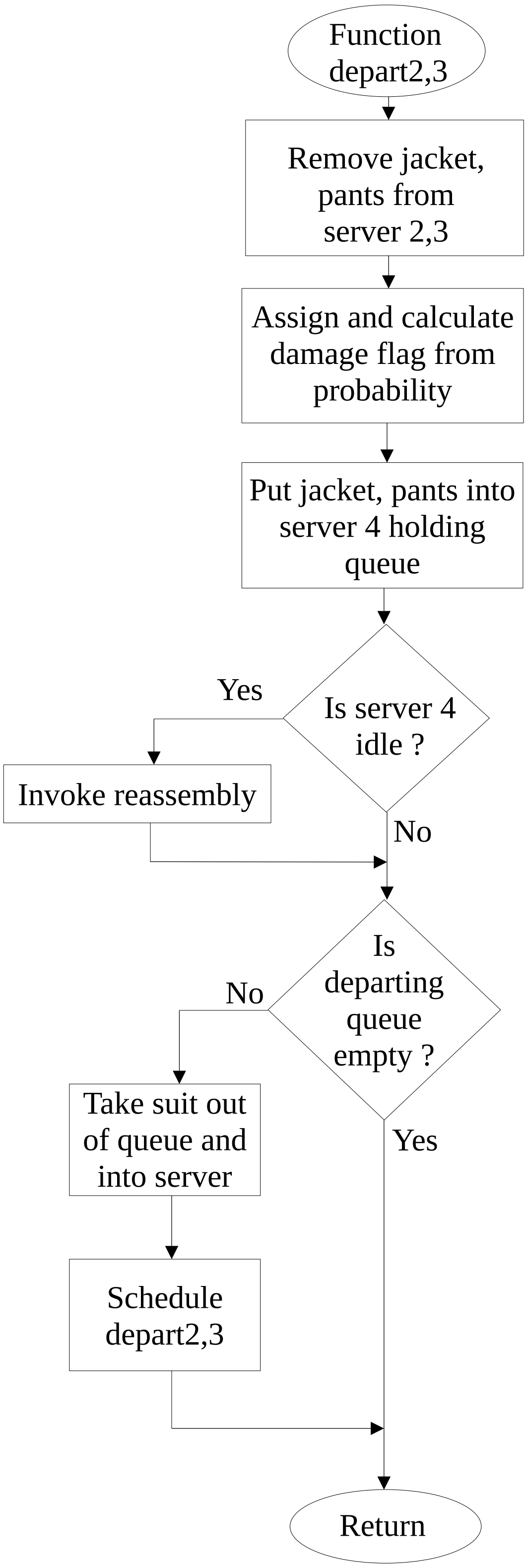

Function depart2,3
Remove jacket, pants from server 2,3
Assign and calculate damage flag from probability
Put jacket, pants into server 4 holding queue
Yes
Is server 4 idle ?
Invoke reassembly
No
No
Is departing queue empty ?
Take suit out of queue and into server
Yes
Schedule depart2,3
Return

FIGURE 2.4
Flowchart for depart2 and depart3 functions, dry-clean model

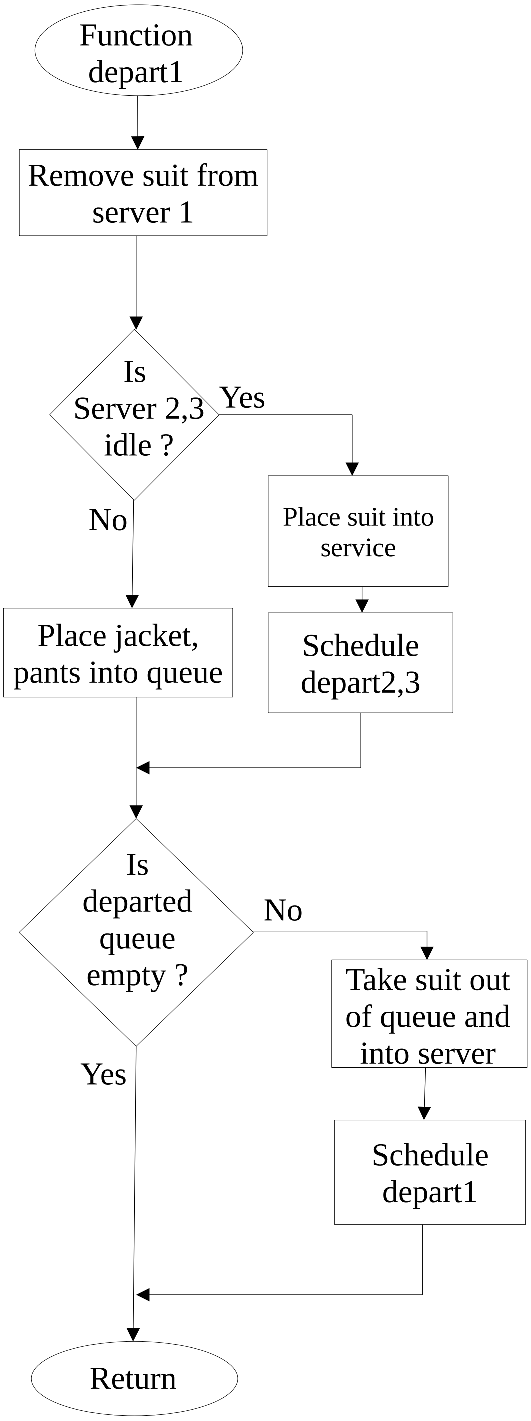

Function depart1
Remove suit from server 1
Is Server 2,3 idle ?
Yes
No
Place suit into service
Schedule depart2,3
Place jacket, pants into queue
Is departed queue empty ?
No
Yes
Take suit out of queue and into server
Schedule depart1
Return

FIGURE 2.3
Flowchart for depart1 function, dry-clean model

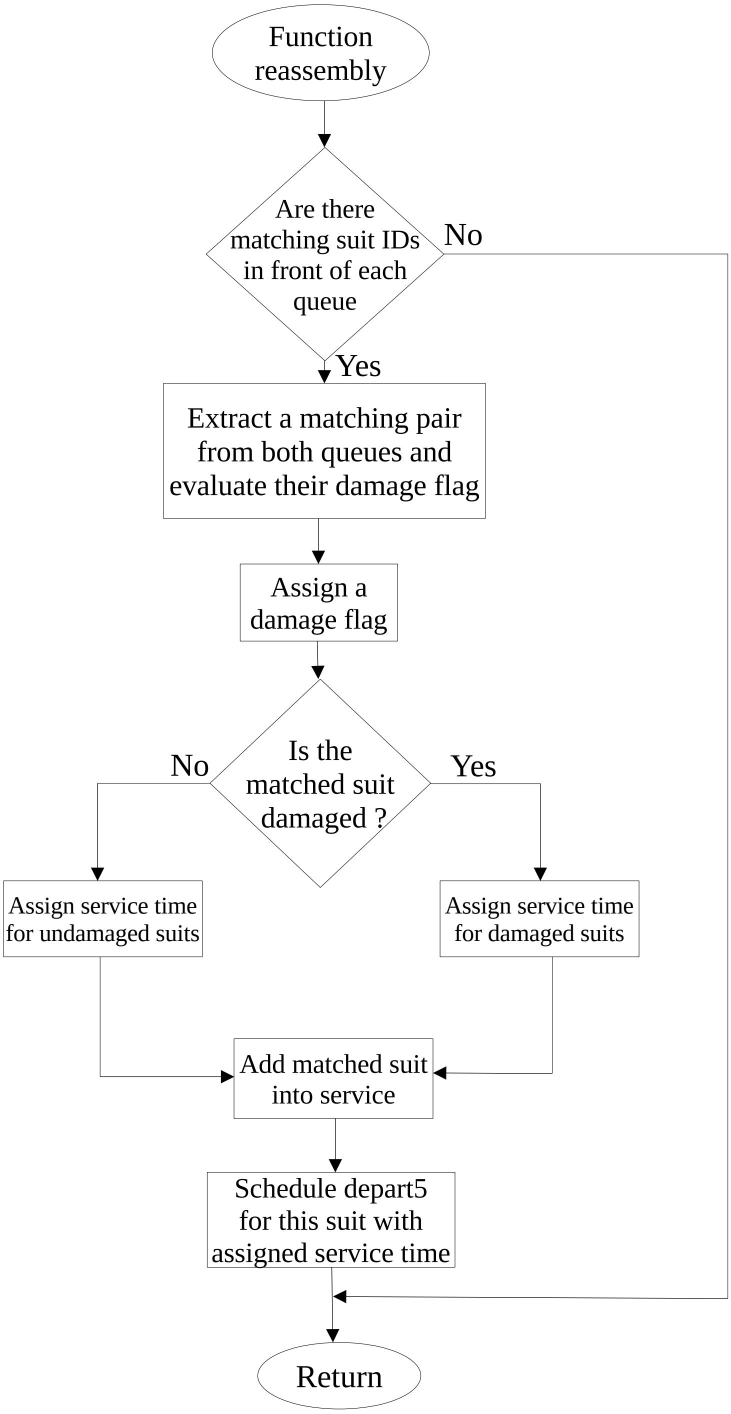

Function reassembly
Are there matching suit IDs in front of each queue
No
Yes
Extract a matching pair from both queues and evaluate their damage flag
Assign a damage flag
Is the matched suit damaged ?
No
Yes
Assign service time for undamaged suits
Assign service time for damaged suits
Add matched suit into service
Schedule depart5 for this suit with assigned service time
Return

FIGURE 2.7
Flowchart for reassembly function, dry-clean model

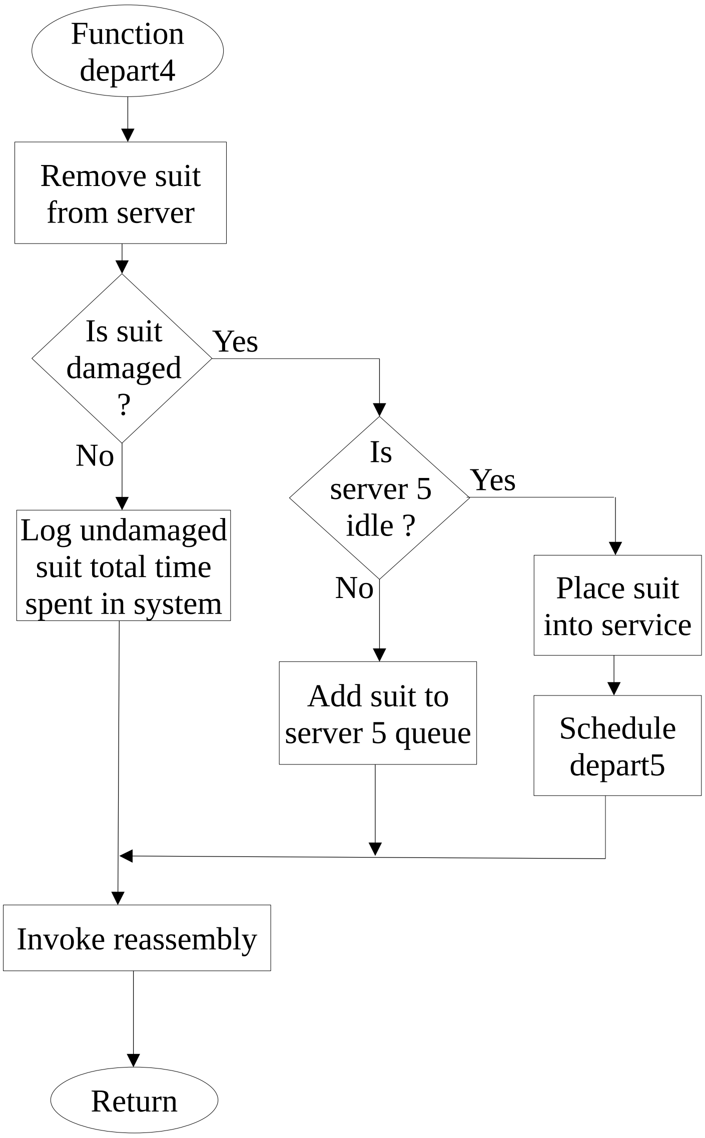


FIGURE 2.5
Flowchart for function depart4, dry-clean model

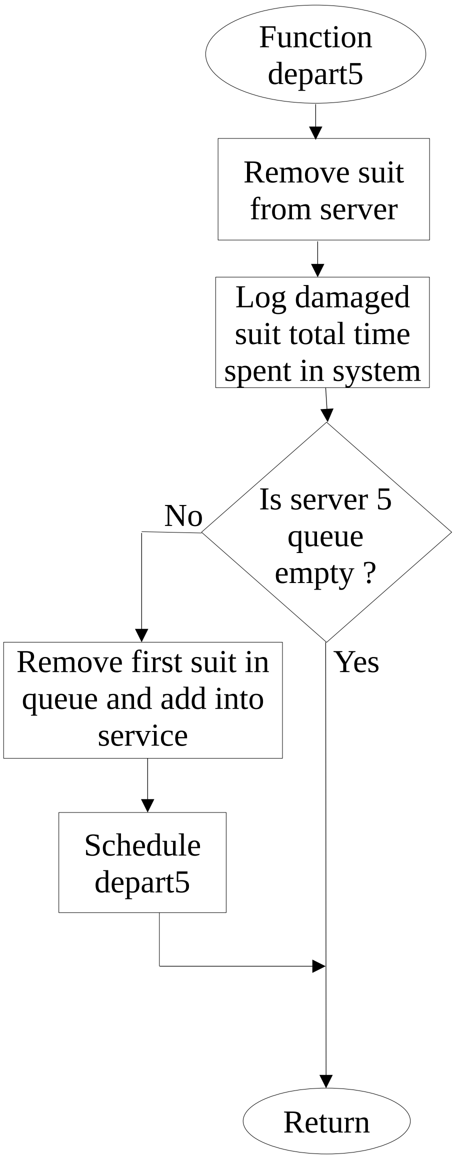

Function depart5
Remove suit from server
Log damaged suit total time spent in system
Is server 5 queue empty ?
No
Yes
Remove first suit in queue and add into service
Schedule depart5
Return

FIGURE 2.6
Flowchart for function depart5, dry-clean model

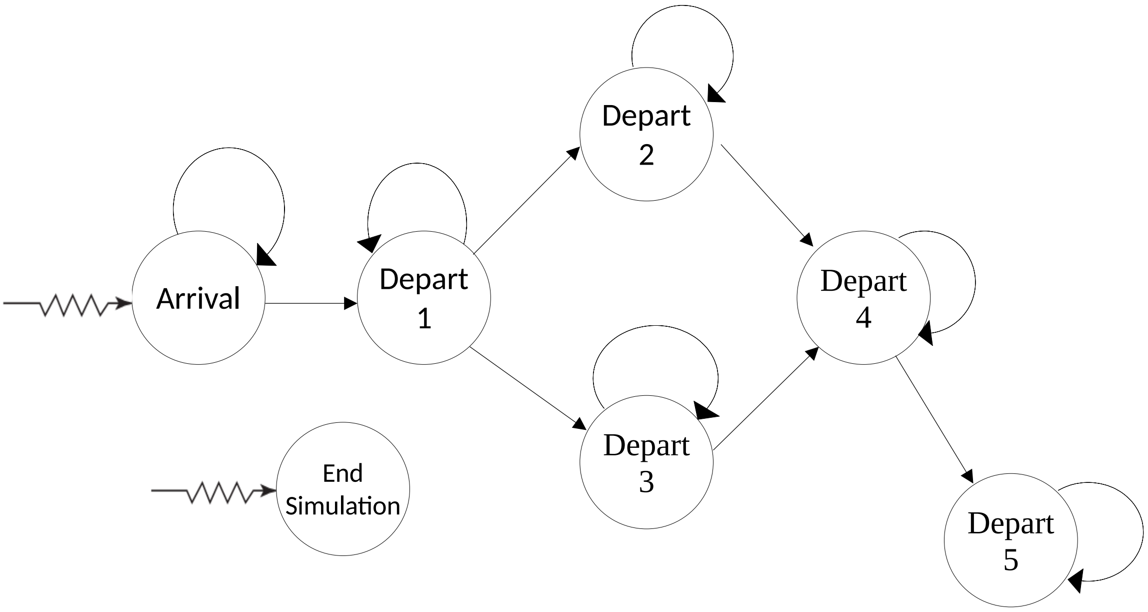
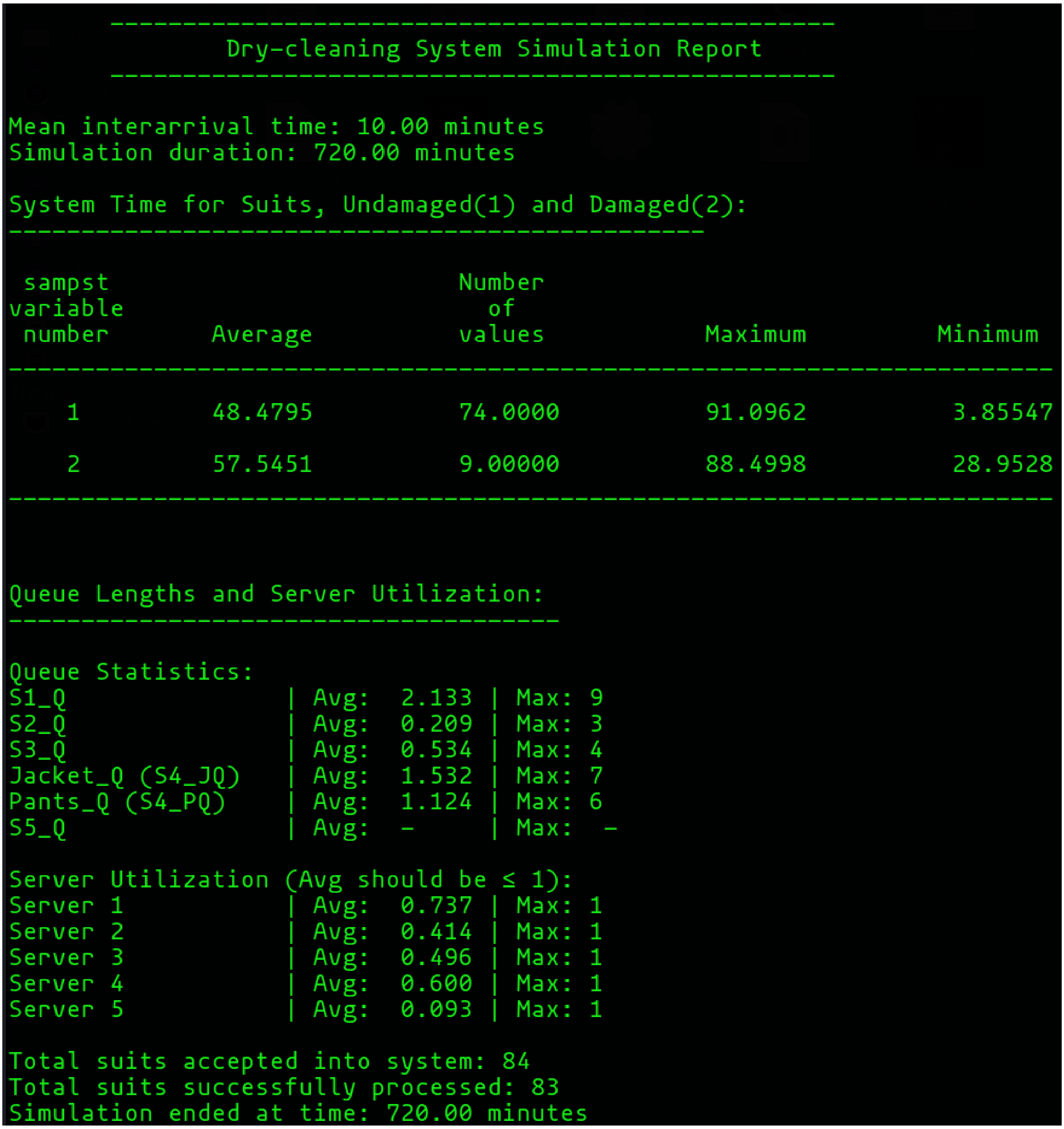
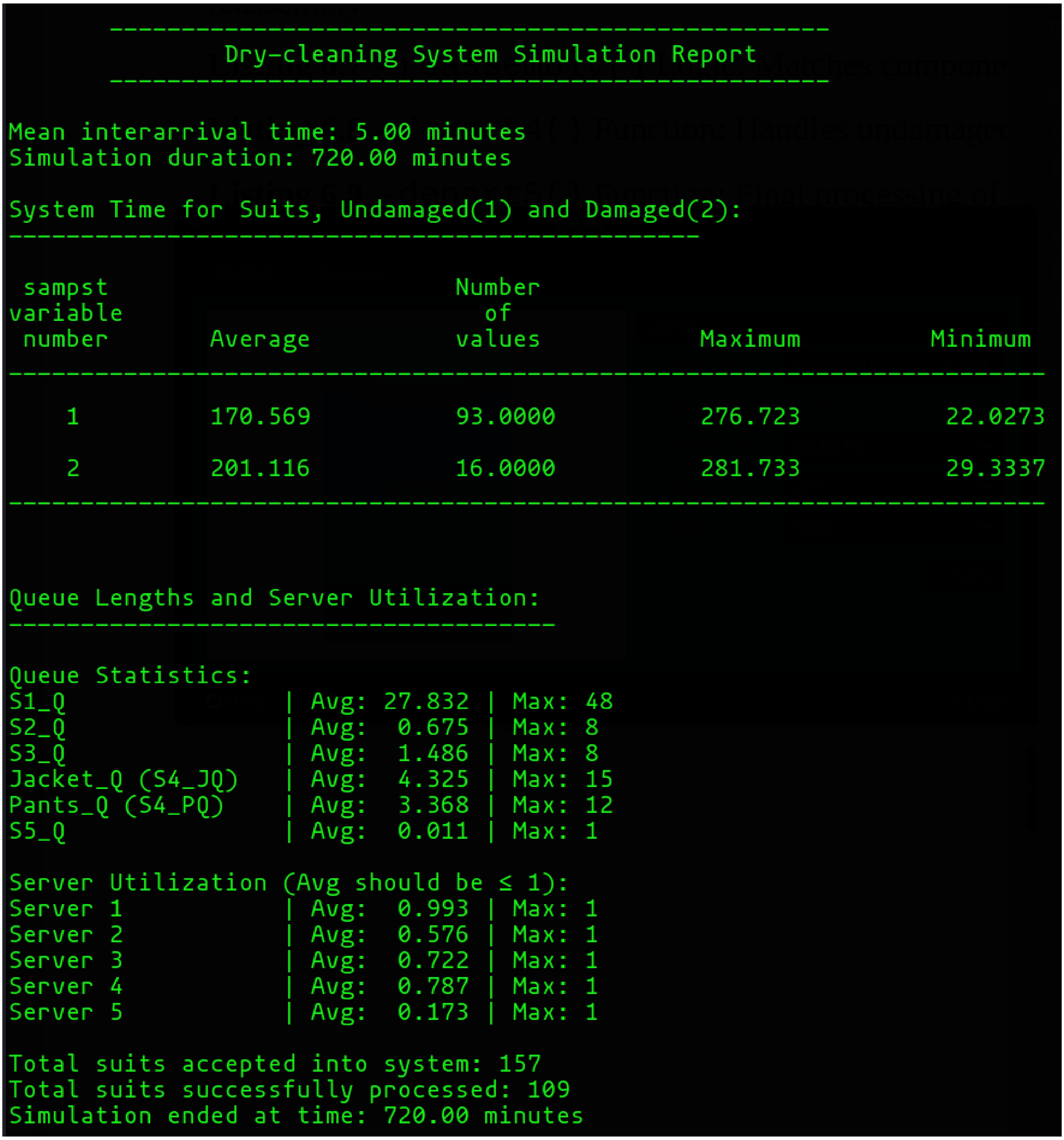


FIGURE 3.1
Event graph, dryclean model



```
            ------------------------------------------------
                   Dry-cleaning System Simulation Report
            ------------------------------------------------

Mean interarrival time: 10.00 minutes
Simulation duration: 720.00 minutes

System Time for Suits, Undamaged(1) and Damaged(2):
----------------------------------------------

 sampst                      Number
variable                       of
 number       Average        values          Maximum         Minimum

----------------------------------------------------------------------

    1         48.4795        74.0000         91.0962         3.85547

    2         57.5451        9.00000         88.4998         28.9528

----------------------------------------------------------------------



Queue Lengths and Server Utilization:
-------------------------------------

Queue Statistics:
S1_Q                  | Avg:  2.133 | Max: 9
S2_Q                  | Avg:  0.209 | Max: 3
S3_Q                  | Avg:  0.534 | Max: 4
Jacket_Q (S4_JQ)      | Avg:  1.532 | Max: 7
Pants_Q (S4_PQ)       | Avg:  1.124 | Max: 6
S5_Q                  | Avg:  -     | Max:  -

Server Utilization (Avg should be ≤ 1):
Server 1              | Avg:  0.737 | Max: 1
Server 2              | Avg:  0.414 | Max: 1
Server 3              | Avg:  0.496 | Max: 1
Server 4              | Avg:  0.600 | Max: 1
Server 5              | Avg:  0.093 | Max: 1

Total suits accepted into system: 84
Total suits successfully processed: 83
Simulation ended at time: 720.00 minutes
```

FIGURE 7.2
Output report for mean 10-minute inter-arrival, dry-clean model.



```
        --------------------------------------------------
                Dry-cleaning System Simulation Report
        --------------------------------------------------

Mean interarrival time: 5.00 minutes
Simulation duration: 720.00 minutes

System Time for Suits, Undamaged(1) and Damaged(2):
------------------------------------------------

 sampst                        Number
variable                        of
 number       Average          values             Maximum          Minimum
------------------------------------------------------------------------

    1         170.569          93.0000            276.723          22.0273

    2         201.116          16.0000            281.733          29.3337
------------------------------------------------------------------------


Queue Lengths and Server Utilization:
--------------------------------------

Queue Statistics:
S1_Q                | Avg: 27.832 | Max: 48
S2_Q                | Avg:  0.675 | Max: 8
S3_Q                | Avg:  1.486 | Max: 8
Jacket_Q (S4_JQ)    | Avg:  4.325 | Max: 15
Pants_Q (S4_PQ)     | Avg:  3.368 | Max: 12
S5_Q                | Avg:  0.011 | Max: 1

Server Utilization (Avg should be ≤ 1):
Server 1            | Avg:  0.993 | Max: 1
Server 2            | Avg:  0.576 | Max: 1
Server 3            | Avg:  0.722 | Max: 1
Server 4            | Avg:  0.787 | Max: 1
Server 5            | Avg:  0.173 | Max: 1

Total suits accepted into system: 157
Total suits successfully processed: 109
Simulation ended at time: 720.00 minutes
```

FIGURE 7.2
Output report for mean 10-minute inter-arrival, dry-clean model