SYSC 4005 Simulation Project

Deliverable 2

Ryan Ribeiro (100997936)

Luke Newton (100999309)

Table of Contents

1.0 Problem Formulation

The manufacturing facility produces three types of product, each requiring one or more different components. Products are assembled at workstations with each unique product type having its own workstation. Each workstation has one or more buffers for each unique component and the workstation must wait for one of each of the required component(s) before assembling the product. There are two inspectors who must inspect the components before sending the components in the appropriate buffer for a given workstation. The policy for selecting which buffer to send the components from inspector one is to pick the workstation with the shortest buffer. If there is a tie between two buffers, the priority is workstation one, then workstation two, then workstation three. If all the buffers of a given component type become full, then inspector for that component is blocked from sending any more components until a buffer has space available.

There are three different types of products, P_1, P_2, and P_3, each assembled at its own workstation W_1, W_2, and W_3 respectively. There are three different types of components, C_1, C_2, and C_3, required to assemble the products. P_1 requires one C_1, P_2 requires one C_1 and one C_2, and P_3 requires one C_1 and one C_3. There are two inspectors, I_1 and I_2, who inspect the components. I_1 inspects component C_1 and I_2 inspects components C_2 and C_3. The inspected components are placed into the workstation buffers, denoted as $B_{i,j}$, where i is the workstation number and j is the component type of the buffer. This means C_1 can be sent to $B_{1,1}$, $B_{2,1}$, or $B_{3,1}$, C_2 is sent to $B_{2,2}$, and C_3 is sent to $B_{3,3}$.

2.0 Setting of Objectives and Overall Project Plan

2.1 Objectives

The question that we want to answer is:

How should inspector 1 distribute their C_1 components?

Criteria for evaluating alternatives is the throughput of the system and idle time of each workstation and inspector. Our primary objective is to maximize the throughput of the system; other objectives are to minimize the idle time of inspectors and workstations.

The current method inspector 1 uses to place C_1 components is to place them in the buffer with the fewest components. If there is a tie, W_1 has highest priority, then W_2, then W_3. We will consider two alternatives to the current method.

First, we will alter the criteria inspector 1 uses to pick a buffer to place C1 components. Instead of picking the shortest queue the inspector's C1 component into, we will use a fair/round-robin policy (i.e. 1st C_1 to W_1 buffer, 2nd C_1 to W_2 buffer, 3rd C_1 to W_3 buffer, 4th C_1 to W_1 buffer, etc.). This will mean that there is no need for the secondary workstation priorities, and we should see a more even number of each product being created. In the event that a buffer is full when it is the buffer's "turn" to receive a C_1, the buffer will be skipped and C_1 will be placed into the next available buffer. If all buffers for component C_1 are full, then inspector 1 is blocked from sending any more components to the buffers. Once a buffer for component C_1 becomes available, inspector 1 sends the component to that buffer and the selection policy continues from that buffer.

Second, we will change the priority of the workstations in the event of a tie when using the current criteria. The current priority when multiple buffers are equally empty prioritizes W_1, but this may lead to inspector 2 being idle for long periods of time, so we will invert the priority order to W_3, then W_2, then W_1.

Simulation is an appropriate method to answer this question because there are many different criteria and priorities to choose, and it would be too time consuming and resource intensive to actually implement a production line for each C_1 distribution method we want to compare.

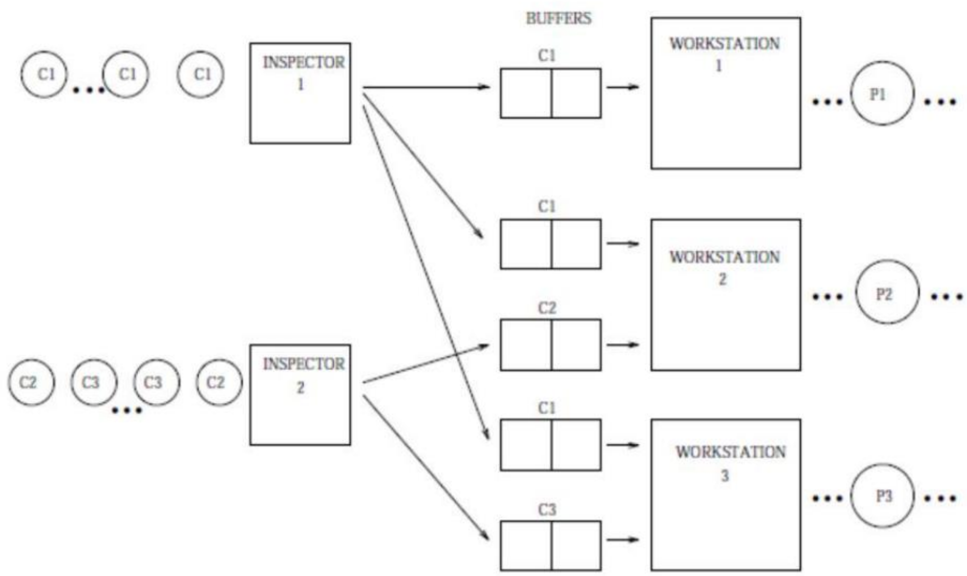
2.2 Project Plan

Our group will meet every Monday from 11:00am to 12:00pm and Tuesday 10:30 to 11:30am to work on the project. These times should work well because they are the scheduled times for the labs each group member is registered in, so we will always be free and will have access to the TA if needed. Depending on progress, additional meetings may be required to ensure completion on time.

The schedule for completing each task is not predictable as it depends on the progress we make in class, but during each weekly meeting the goal is to apply the new knowledge from the previous week's classes so that the project stays up to date.

3.0 Model Conceptualization

Below is the provided schematic of the manufacturing facility from which the entities, relationships among these entities, abstractions, and assumptions that are being made about the system will be listed.



3.1 Entities

Component: The lowest level piece of the system which is used by every other part of the manufacturing system. There is an infinite inventory of components available and taking components from the inventory happens instantaneously. There are three types of components.

Inspector: An inspector is an individual that inspects specific components for some time and then sends the component off to a buffer before starting to inspect the next component. There are two different inspectors.

Buffer: A queue which holds only a specific type of component for a given work station. At most, a queue will hold two items. There are five buffers among all workstations.

Workstation: The part of the system which takes in components as inputs to general different products as outputs. There are three workstations total and each workstation only assembles on type of product. Workstations can only begin to assemble products once they have the required components in their buffers.

Product: The final output of the system obtained from a workstation. There are three different products created and output by the system.

3.2 Entity Interactions

Components and Inspectors: Inspectors take a single component (C1 to inspector 1 and C2/C3 to inspector 2) and inspect the component for some amount of time. Once completed, the inspector will send the component to a specific buffer and instantaneously retrieve take another component for inspection.

Components and Buffers: Components are placed into buffers of the same component type and are only removed from the buffer when the workstation begins assembly.

Components and Workstations: Workstations require a specific combination of components before constructing a product out of the components.

Components and Products: Components are used in the assembly of products.

Inspectors and Buffers: Inspectors place the inspected components into a buffer of the same component type. The buffer chosen depends on the specific routing policy being used. Additionally, buffers can block inspectors from sending additional components if the buffers for that component type are full.

Workstations and Buffers: Each workstation has one buffer per component type that is used in the assembly the product associated with the specific workstation. The workstation will remove one of each of the required components from the workstation's buffer(s) in order to assemble a product.

Workstation and Product: Workstations will assemble a specific product over some amount of time once the required components exist in the workstation's buffers.

3.3 Essential Features

The essential features of this simulation are the inspectors, the buffers, and the workstations. The inspectors are essential because the amount of time taken to inspect the components directly affects how long it takes to go from component to finished product. The buffers are essential because it is important to observe whether the buffers are being filled up too often, or not often enough, to determine if there is a bottle neck in the speed of inspecting a given component. Finally, the workstations are essential for a similar reason as the inspectors in determining the total time taken to go from components to final products.

3.4 Assumptions

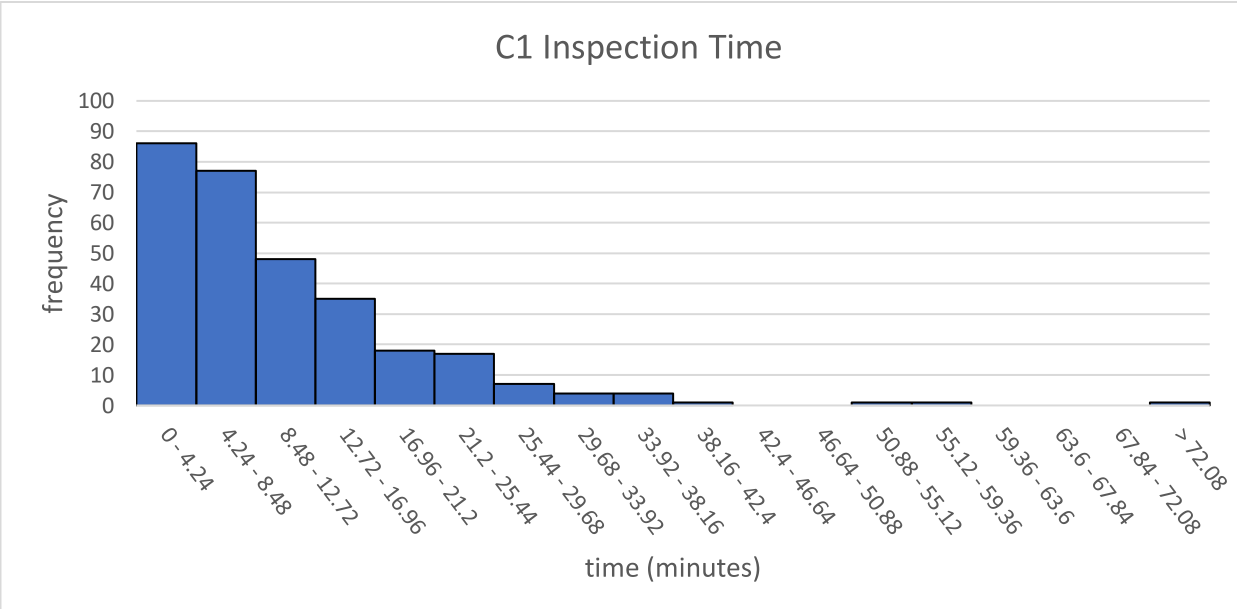
When an inspector gets blocked from placing a component into a buffer on account of the buffer being full, it is assumed that the inspector will keep that component and will able to instantly send that component to a buffer once a buffer becomes available. The reason for this assumption is that the component will eventually be needed, and it would be a waste of time to dispose of the component and then inspect another one later.

4.0 Data Collection and Input Modelling

Data has been collected for 6 different aspects of the environment: Inspection time for C1, Inspection time for C2, Inspection time for C3, W1 assembly time, W2 assembly time, and W3 assembly time.

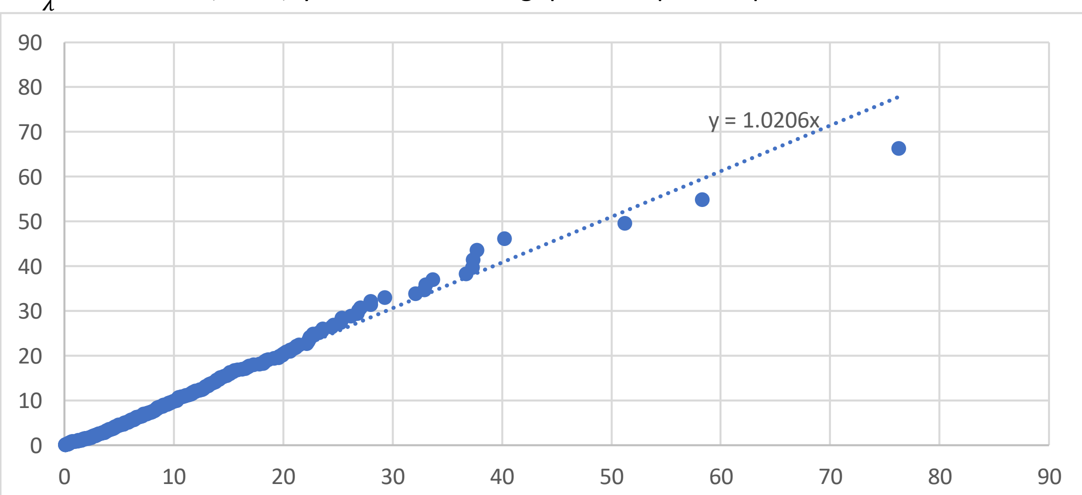
4.1 Inspection Time for C1

Given that the data represents service times and is a continuous variable, it likely follows an exponential distribution. As there are 300 data points, $\lceil\sqrt{300}\,\rceil = 18$ bins will be used to produce the histogram below:



Based on the shape of this plot, the sample data does seem to fit an exponential distribution, but quantile-quantile plot and chi-squared test must be used before we can say with certainty that this is the distribution that fits.

Using $\lambda = \frac{1}{\bar{X}}$ as the parameter estimation and the quantile function $F^{-1}(x) = \frac{-ln(1-x)}{\lambda} = -\bar{X}ln(1-x)$ yields the following quantile-quantile plot:



The linearity of this plot further confirms that an exponential distribution for C1 inspection time is a good fit, and the parameter estimation is accurate since the slope of the plot is almost exactly one.

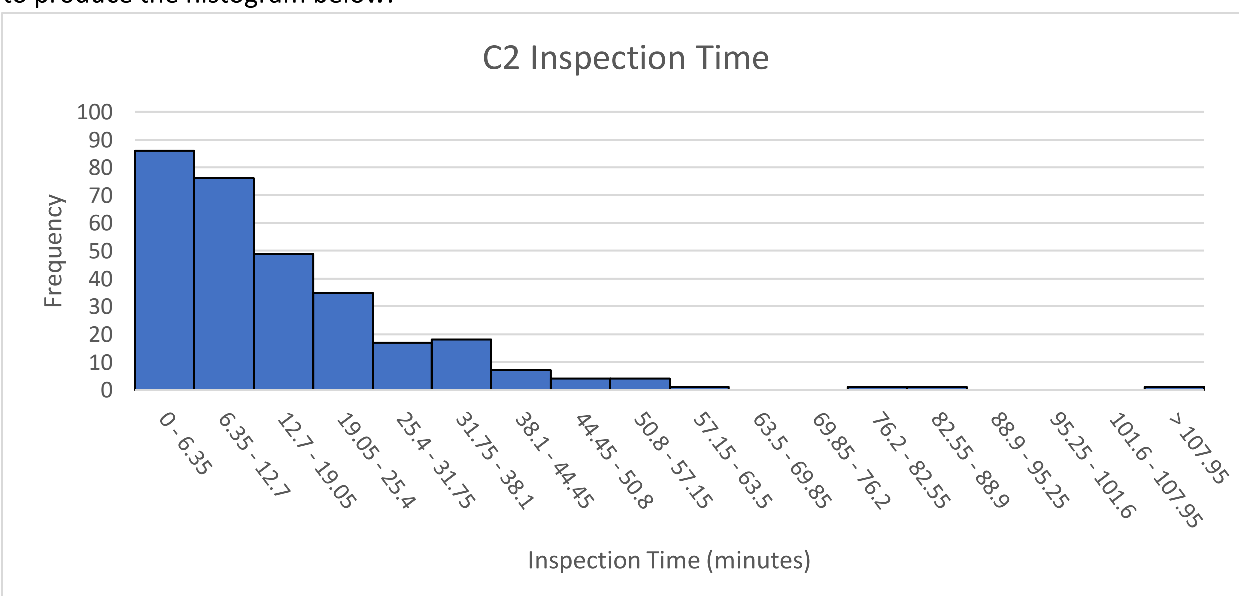
To finalize this choice of distribution, a chi-squared test must be performed. This is an appropriate test to use in these circumstances since there are a large number of data points and there are estimated parameters.

H_0 = the inspection time for C1 follows an exponential distribution with $\lambda = \frac{1}{\bar{X}}$ =0.096545

With 18 bins and one variable estimated, the assumed chi-square distribution has $18 - 1 - 1 = 16$ degrees of freedom. Using a significance level of 0.05, the threshold value obtained from tables is $z_o{}^2{}_{0.05,\,16} = 26.3$. By calculating expected frequencies from the hypothesized exponential distribution (see Appendix 1.1), $z_o{}^2 = 15.0409$. Since this is below the threshold value, the hypothesis can be confidently accepted, and we can say with certainty that the inspection time for C1 components follows an exponential distribution with $\lambda =$ 0.096545.

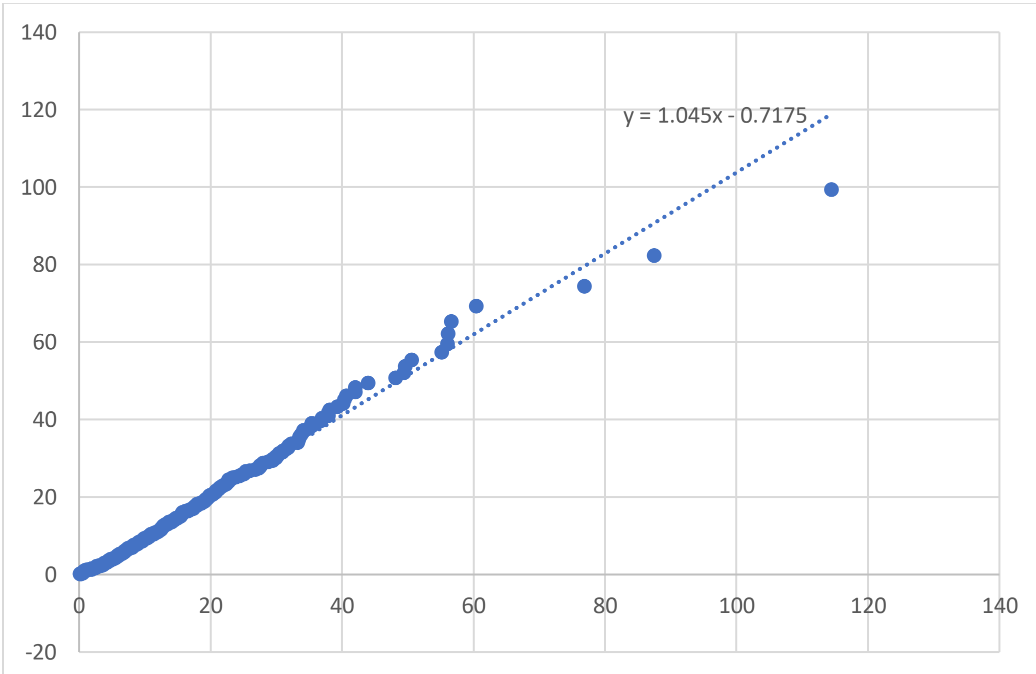
4.2 Inspection Time for C2

Given that the data represents service times and is a continuous variable, it likely follows an exponential distribution. As there are 300 data points, $\lceil\sqrt{300}\,\rceil = 18$ bins will be used to produce the histogram below:



Based on the shape of this plot, the sample data does seem to fit an exponential distribution, but quantile-quantile plot and chi-squared test must be used before we can say with certainty that this is the distribution that fits.

Using $\lambda = \frac{1}{\bar{X}}$ as the parameter estimation and the quantile function $F^{-1}(x) = \frac{-ln(1-x)}{\lambda} = -\bar{X}ln(1-x)$ yields the following quantile-quantile plot:



The linearity of this plot further confirms that an exponential distribution for C2 inspection time is a good fit, and the parameter estimation is accurate since the slope of the plot is almost exactly one. A few of the largest values do not adhere to this linear trend as strongly, but this is fairly typical of the extrema in q-q plot.

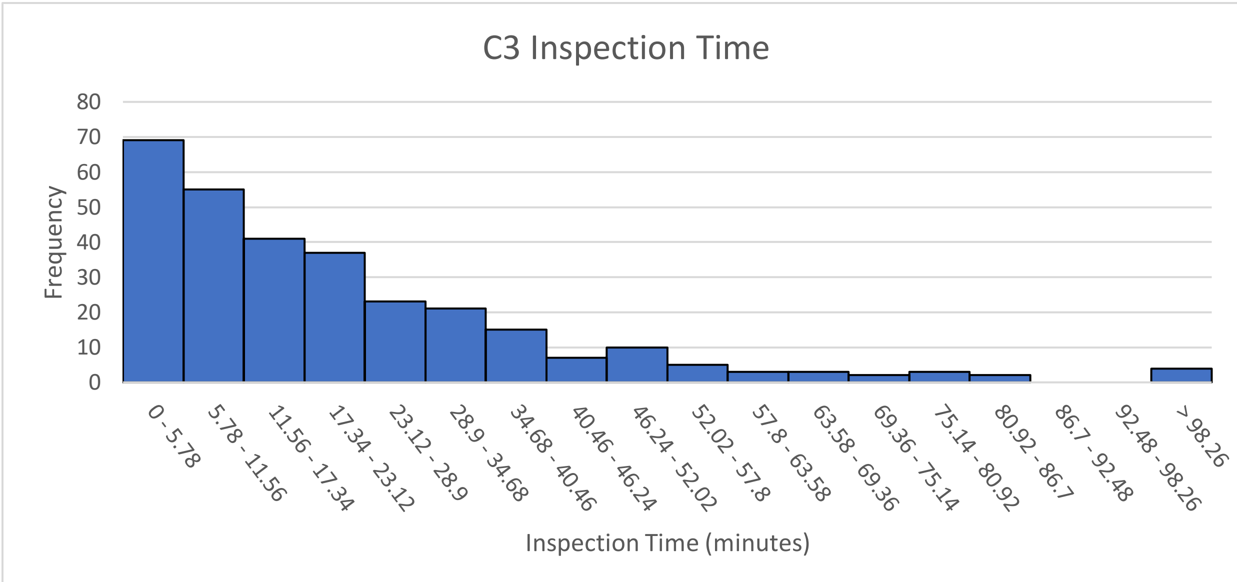
To finalize this choice of distribution, a chi-squared test must be performed. This is an appropriate test to use in these circumstances since there are a large number of data points and there are estimated parameters.

H_0 = the inspection time for C2 follows an exponential distribution with $\lambda = \frac{1}{\bar{X}}$ =0.0644

With 18 bins and one variable estimated, the assumed chi-square distribution has $18 - 1 - 1 = 16$ degrees of freedom. Using a significance level of 0.05, the threshold value obtained from tables is $z_o{}^2{}_{0.05,\,16}$ = 26.3. By calculating expected frequencies from the hypothesized exponential distribution (see Appendix 1.2), $z_o{}^2$ = 15.663. Since this is below the threshold value, the hypothesis can be confidently accepted, and we can say with certainty that the inspection time for C2 components follows an exponential distribution with $\lambda =$ 0.0644.

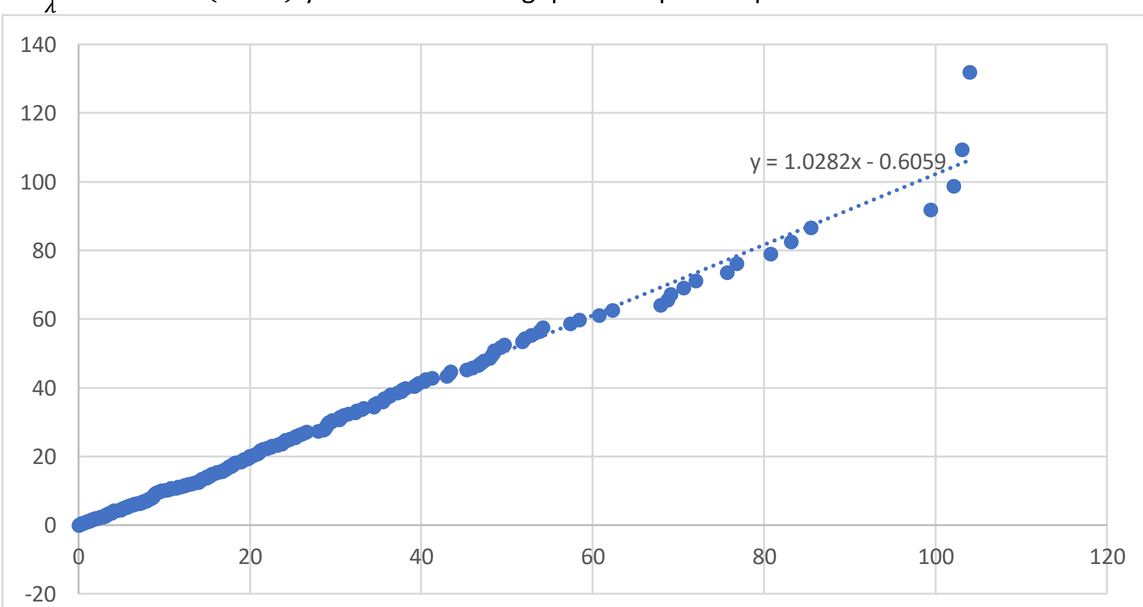
4.3 Inspection Time for C3

Given that the data represents service times and is a continuous variable, it likely follows an exponential distribution. As there are 300 data points, $\lceil\sqrt{300}\,\rceil = 18$ bins will be used to produce the histogram below:



Based on the shape of this plot, the sample data does seem to fit an exponential distribution, but quantile-quantile plot and chi-squared test must be used before we can say with certainty that this is the distribution that fits.

Using $\lambda = \frac{1}{\bar{X}}$ as the parameter estimation and the quantile function $F^{-1}(x) = \frac{-ln(1-x)}{\lambda} = -\bar{X}ln(1-x)$ yields the following quantile-quantile plot:



The linearity of this plot further confirms that an exponential distribution for C3 inspection time is a good fit, and the parameter estimation is accurate since the slope of the plot is almost exactly one. A few of the largest values do not adhere to this linear trend as strongly, but this is fairly typical of the extrema in q-q plot.

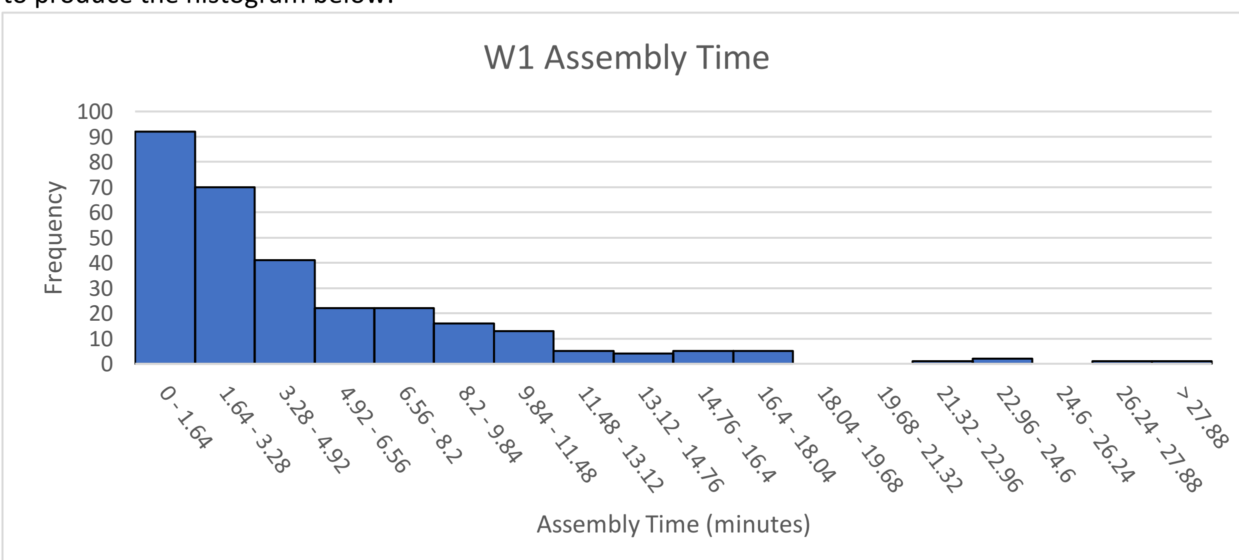
To finalize this choice of distribution, a chi-squared test must be performed. This is an appropriate test to use in these circumstances since there are a large number of data points and there are estimated parameters.

H_0 = the inspection time for C3 follows an exponential distribution with $\lambda = \frac{1}{\bar{X}}$ =0.048467

With 18 bins and one variable estimated, the assumed chi-square distribution has $18 - 1 - 1 = 16$ degrees of freedom. Using a significance level of 0.05, the threshold value obtained from tables is $z_o{}^2_{0.05,\,16} = 26.3$. By calculating expected frequencies from the hypothesized exponential distribution (see Appendix 1.3), $z_o{}^2 = 17.48892$. Since this is below the threshold value, the hypothesis can be confidently accepted, and we can say with certainty that the inspection time for C3 components follows an exponential distribution with $\lambda =$ 0.048467.

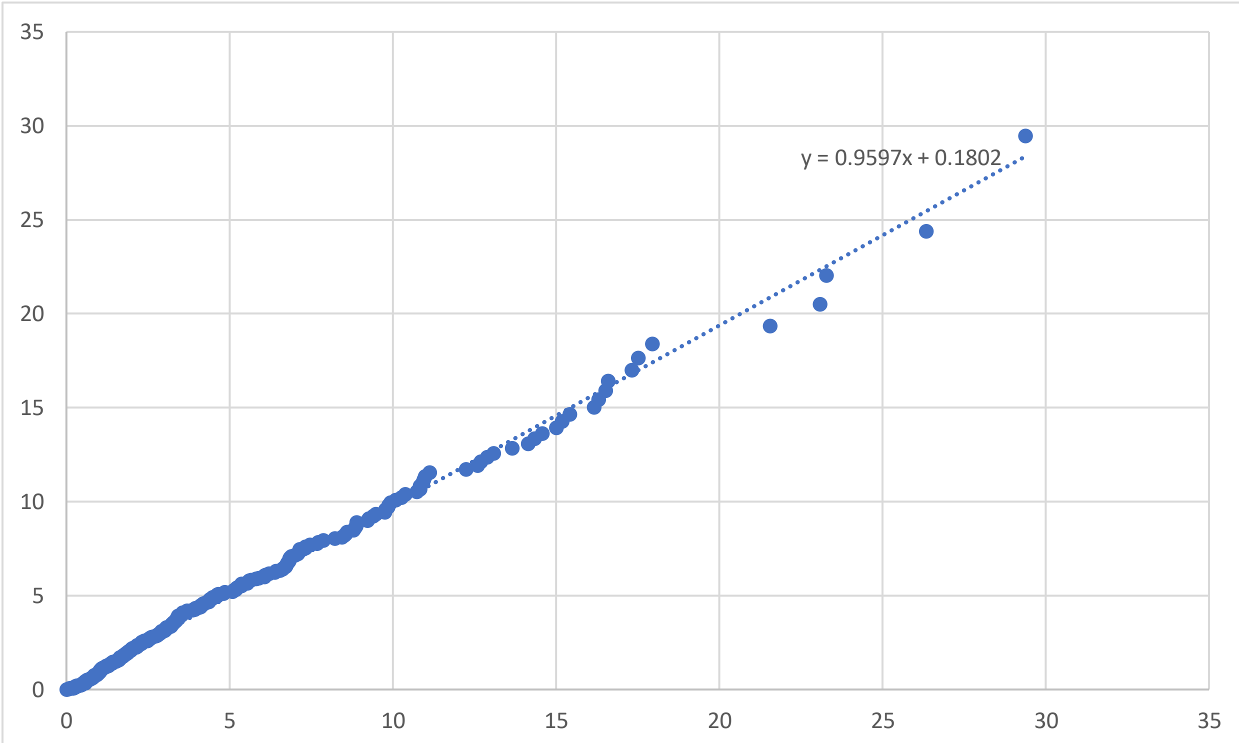
4.4 W1 Assembly Time

Given that the data represents service times and is a continuous variable, it likely follows an exponential distribution. As there are 300 data points, $\lceil\sqrt{300}\,\rceil = 18$ bins will be used to produce the histogram below:



Based on the shape of this plot, the sample data does seem to fit an exponential distribution, but quantile-quantile plot and chi-squared test must be used before we can say with certainty that this is the distribution that fits.

Using $\lambda = \frac{1}{\bar{X}}$ as the parameter estimation and the quantile function $F^{-1}(x) = \frac{-ln(1-x)}{\lambda} = -\bar{X}ln(1-x)$ yields the following quantile-quantile plot:



The linearity of this plot further confirms that an exponential distribution for W1 assembly time is a good fit, and the parameter estimation is accurate since the slope of the plot is almost exactly one.

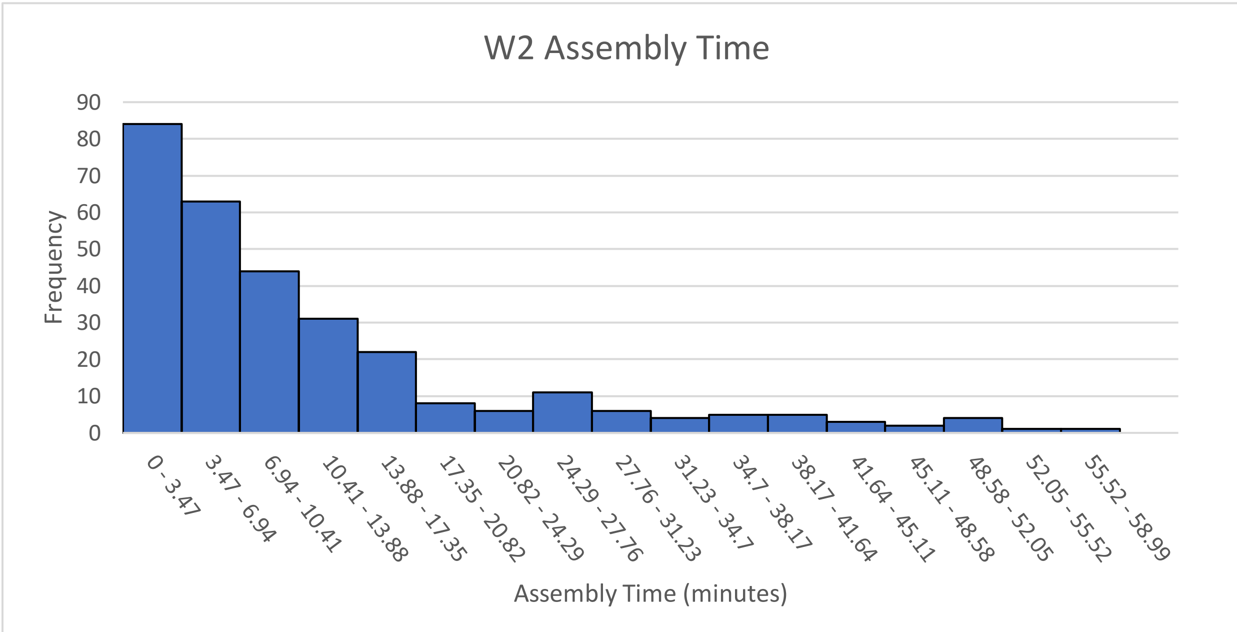
To finalize this choice of distribution, a chi-squared test must be performed. This is an appropriate test to use in these circumstances since there are a large number of data points and there are estimated parameters.

H_0 = the assembly time for W1 follows an exponential distribution with $\lambda = \frac{1}{\bar{X}}$ =0.217183

With 18 bins and one variable estimated, the assumed chi-square distribution has $18 - 1 - 1 = 16$ degrees of freedom. Using a significance level of 0.05, the threshold value obtained from tables is $z_o{}^2{}_{0.05,\,16}$ = 26.3. By calculating expected frequencies from the hypothesized exponential distribution (see Appendix 1.4), $z_o{}^2$ = 13.52932. Since this is below the threshold value, the hypothesis can be confidently accepted, and we can say with certainty that the assembly time for W1 components follows an exponential distribution with $\lambda = 0.217183$.

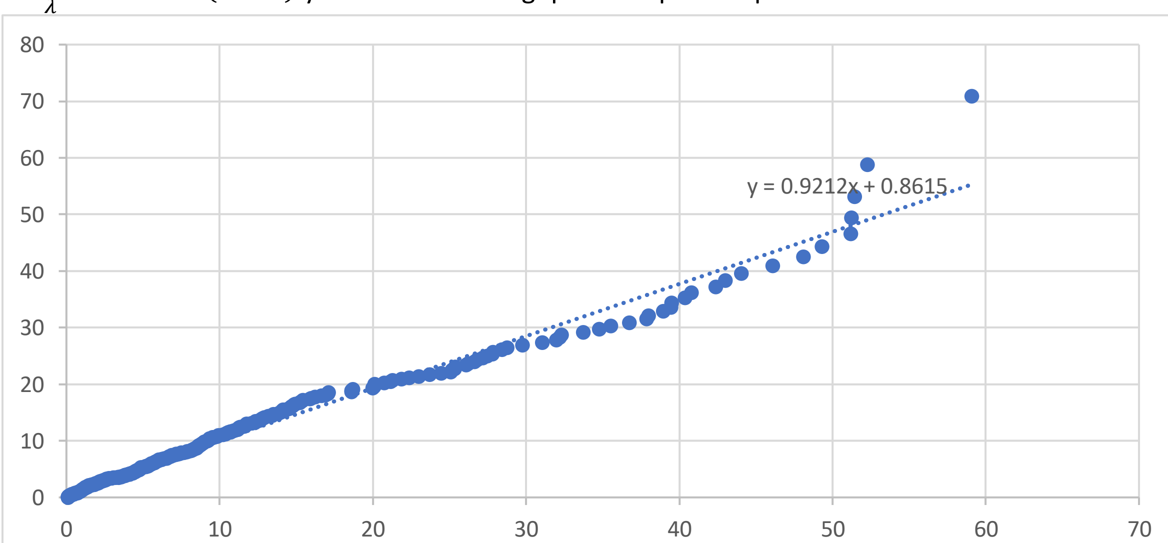
4.5 W2 Assembly Time

Given that the data represents service times and is a continuous variable, it likely follows an exponential distribution. As there are 300 data points, $\lceil \sqrt{300} \rceil = 18$ bins would be the standard; however, to reduce the number of empty bins, 17 bins will be used in the graph below:



Based on the shape of this plot, the sample data does seem to fit an exponential distribution, but quantile-quantile plot and chi-squared test must be used before we can say with certainty that this is the distribution that fits.

Using $\lambda = \frac{1}{\bar{X}}$ as the parameter estimation and the quantile function $F^{-1}(x) = \frac{-ln(1-x)}{\lambda} = -\bar{X}ln(1-x)$ yields the following quantile-quantile plot:



The linearity of this plot further confirms that an exponential distribution for W2 assembly time is a good fit, and the parameter estimation is accurate since the slope of the plot is almost exactly one. A few of the largest values do not adhere to this linear trend as strongly, but this is fairly typical of the extrema in q-q plot.

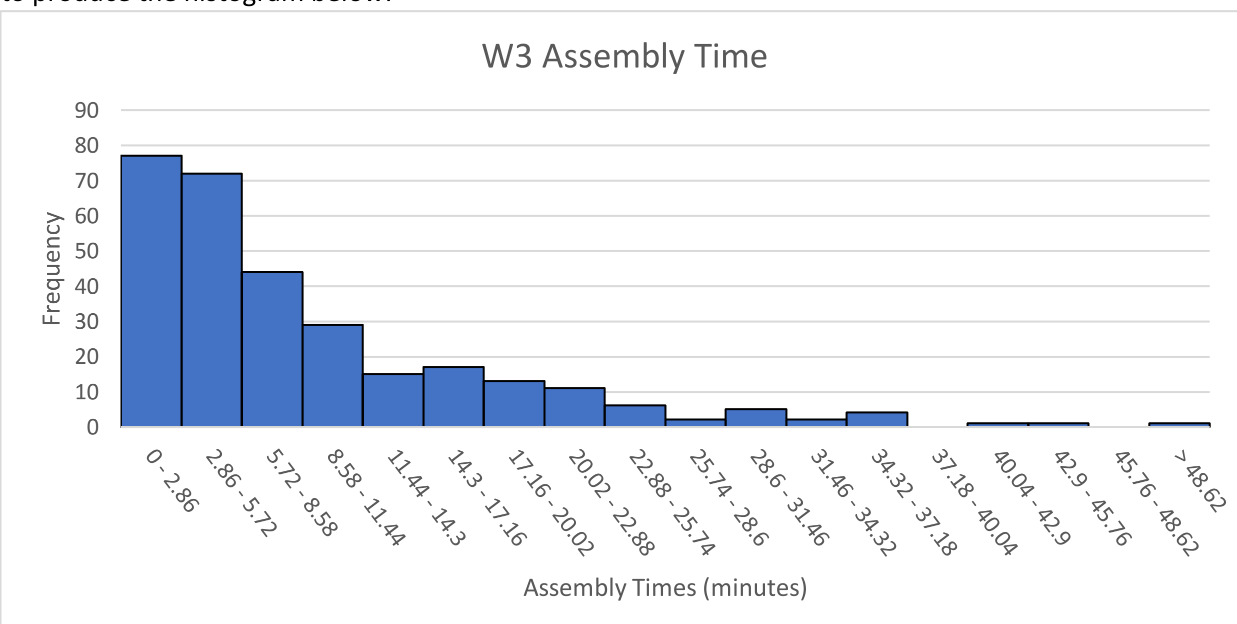
To finalize this choice of distribution, a chi-squared test must be performed. This is an appropriate test to use in these circumstances since there are a large number of data points and there are estimated parameters.

H_0 = the assembly time for W2 follows an exponential distribution with $\lambda = \frac{1}{\bar{X}}$ =0.0902

With 17 bins and one variable estimated, the assumed chi-square distribution has $17 - 1 - 1 = 15$ degrees of freedom. Using a significance level of 0.05, the threshold value obtained from tables is $z_o^2{}_{0.05,\,15}$ = 25. By calculating expected frequencies from the hypothesized exponential distribution (see Appendix 1.5), z_o^2 = 13.64. Since this is below the threshold value, the hypothesis can be confidently accepted, and we can say with certainty that the assembly time for W2 components follows an exponential distribution with $\lambda =$ 0.0902.

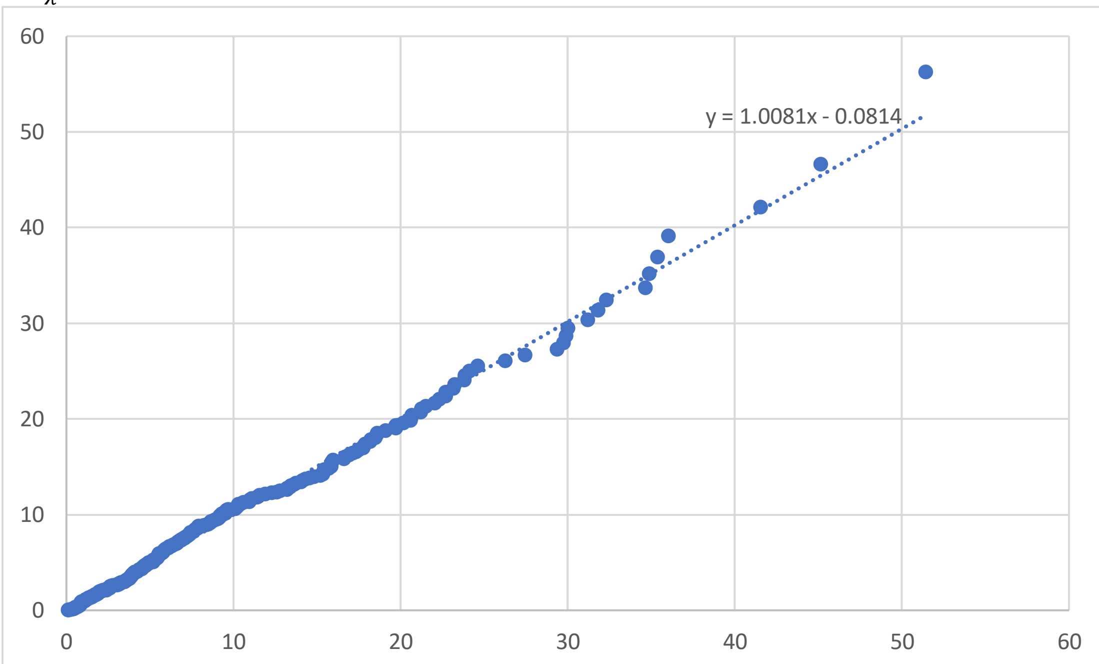
4.6 W2 Assembly Time

Given that the data represents service times and is a continuous variable, it likely follows an exponential distribution. As there are 300 data points, $\lceil\sqrt{300}\,\rceil = 18$ bins will be used to produce the histogram below:



Based on the shape of this plot, the sample data does seem to fit an exponential distribution, but quantile-quantile plot and chi-squared test must be used before we can say with certainty that this is the distribution that fits.

Using $\lambda = \frac{1}{\bar{X}}$ as the parameter estimation and the quantile function $F^{-1}(x) = \frac{-ln(1-x)}{\lambda} = -\bar{X}ln(1-x)$ yields the following quantile-quantile plot:



The linearity of this plot further confirms that an exponential distribution for W3 assembly time is a good fit, and the parameter estimation is accurate since the slope of the plot is almost exactly one.

To finalize this choice of distribution, a chi-squared test must be performed. This is an appropriate test to use in these circumstances since there are a large number of data points and there are estimated parameters.

H_0 = the assembly time for W3 follows an exponential distribution with $\lambda = \frac{1}{\bar{X}}$ =0.113693

With 18 bins and one variable estimated, the assumed chi-square distribution has $18 - 1 - 1 = 16$ degrees of freedom. Using a significance level of 0.05, the threshold value obtained from tables is $z_o^2{}_{0.05,\,16}$ = 26.3. By calculating expected frequencies from the hypothesized exponential distribution (see Appendix 1.6), z_o^2 = 13.68477. Since this is below the threshold value, the hypothesis can be confidently accepted, and we can say with certainty that the assembly time for W3 components follows an exponential distribution with $\lambda = 0.113693$.

5.0 Model Translation

5.1 Choice of Simulation Language

MATLAB is the chosen language for this simulation, as it provides many benefits over other options. MATLAB is readily available on university computers, and can be downloaded for free through the university license, so the simulation will not require purchasing any specialized software tools. All team members have experience using MATLAB as well, meaning that most time can be spent on implementing the model, rather than learning the language. Most importantly, MATLAB has built in support for plotting, creating distributions, sampling from distributions, and generating random numbers; all of which will be useful functionalities for the simulation that we would otherwise need to implement ourselves.

5.2 High Level Description

The simulation starts by first initializing the model based on the results from milestone one, along with the setup of the future event list (FEL) and the adding of initial events into the FEL. Next, we take the first event from the FEL, sorted in chronological order, and we process the event; this processing differs between event types. After the event is processed, we check if the FEL is empty. If it is empty, then we are finished, and we can output the statistics recorded from the simulation. If we are not finished, then we keep processing the top event from our FEL until all events are processed.

For a component C1 event, we first check if all the C1 queues are blocked. If they are blocked, then we block inspector 1 from producing more C1s. If there is an open queue to place a C1, then we do so for a given workstation. At that workstation, we then check if all the necessary components are available to assemble the given product for that workstation. If we are missing components, then we just advance the simulation and generate an event indicating that a component C1 is ready. If the workstation does have the remaining components, then we check if there is already a product being produced in the workstation. If there is a product being produced, we generate an event for component C1 being ready as before. If there is no product at that workstation, then we unblock the workstation, remove the components from the queues, we generate an event to indicate that the product has been built, and then we generate an event for component C1 being ready.

For component C2 and C3 events, we handle them both in a similar fashion to C1 events. We first check if workstation two and three respectively are full. If either are, then we block inspector two and advance to the next event. If they aren't full, then we check if the respective workstations have the necessary components to assemble the product and if they are not currently assembling a product. If they do have the components and are not assembling a product, then we unblock the workstation, remove the components, and generate a product-built event. If the workstation didn't have the components or it was assembling a product already, then we check if the workstation's component queue is full for their respective component. If the queue is full, we generate an event for the opposite component (i.e. workstation 2 generates a C3 event, and workstation 3 generates a C2 event) and we exit. If the queue isn't full, then we check if the other workstation's queue is full and generate an event for that workstation's component and generate an event if we can. If both workstations have full queues, then we randomly select what component to generate an event for.

For the P1Built event, we start by increasing the total number of product P1s built. Then, we check if the queue for workstation 1 is empty. If the queue is empty, we block workstation one to put it in an idle state. If it is not empty, then we want to remove a component C1. We then check if inspector one is currently blocked or not. If they aren't blocked, we just generate the next P1Built event. If they are blocked, we unblock them and generate the next P1Built event all the same.

For P2Built and P3Built events, we handle them in a similar fashion. We treat them in the same way as P1Built events up until the point where we check if the inspector one is blocked. After we have checked if inspector one is blocked, we unblock him if he is (or leave him unblocked if he isn't blocked), and we proceed to check if inspector two is blocked. If inspector two is not blocked, we just generate the next P2Built or P3Built event respectively. If the inspector is blocked, we unblock them and generate the same events as previously stated.

5.3 Design Choices

5.3.1 Calculating Idle time for Inspectors and Workstations

We calculate the idle times for the inspectors and the workstations in a similar fashion. In essence, we are using global variable to record the start and end times for when an inspector or workstation is idle. Then, we take the difference between the end and start time and add it to the overall idle time for a specific inspector or workstation. For inspectors, we take the start time at when they get blocked from inspecting more components, and the end time from when they are unblocked. Similarly, for workstations we take the start time at when the one or both queues for that workstation are empty and a product is not in production, and we record the end time at the time the workstation begins to assemble an item. We take the difference and add it to the overall workstation idle time.

5.3.2 Unblocking Inspectors

We handle the unblocking of inspectors in our product-built events. We have decided to do this because an inspector is only blocked when they have nowhere to place a component. As a result, an inspector can only be unblocked if the components are consumed and used in the assembly of a product. Therefore, we determine which inspector should be unblocked depending on the type of product being assembled, and we handle this when the product is built.

5.3.3 Using a Single Random Number Stream for all Sampling

For the initial implementation of the simulation, we use a single random number stream for sampling points from all random variables. This was primarily done due to our implementation priorities; we have initially focused on the logic of each event and the interaction between them over smaller items like random sampling. In later stages this will be replaced with a separate random number stream for each distribution, so that results can be reproducible by picking the same seeds from each stream.

5.3.4 Verbose Mode

We made the decision to include a verbose mode for the simulation. This is controlled by a boolean value at the beginning on the Sim.m script and allows detailed information about the execution to be displayed in the command window. We chose to do this because it allows us to see the order that events occur and are processed so we can ensure the simulation runs properly without doing much formal testing (which will be in the next deliverable).

5.3.5 Defining Objects for Event and FutureEventList

While most of the simulation is written in a procedural style, we chose to also create specific objects for events and for the future event list. Making an object for the future event list was helpful for the design of the program, because it allows us to create a customized list with capabilities beyond what a normal MATLAB vector could do and allows us to encapsulate all the functionalities related to the future event list in one place. Since we created this custom list object, it was also necessary to create an object for an event so we can group all the information for a single event together and sort them all.

5.4 Alternative Design Description

The proposed alternative design (description here) appears in the function to process C1Ready events. When the activity reaches the point where the inspector must decide which queue to place the new component one into, two global booleans called alternativeStrategy and alternativePriority are checked. These boolean variables are set at the top of Sim.m, along with other program control variables.

If alternativeStrategy is true, then we use the alternative round-robin strategy for place component 1's, otherwise the components are placed in the shortest queue. If alternativePriority is true, then when two queues have the same length, the order of priority for choosing a queue is workstation 3, then 2, then 1; otherwise the default priority of workstation 1, then 2, then 3 is used. Note that if both alternativeStrategy and alternativePriority are true, only the alternative strategy is used, because a round-robin approach does not require a tie-breaking decision.

Appendix

Appendix 1.1: Spreadsheet Data for C1 Inspection Time

number of points	300
number of bins	18
max value	76.284
min value	0.087
data range	76.197
bin width	4.233167
sample mean	10.08446
sample variance	92.5225

bin	bin lower range	bin upper range	range	observed frequency O_i	expected frequency E_i	X_0^2 $\frac{(O_i - E_i)^2}{E_i}$
1	0	4.24	0 - 4.24	86	101	2.227723
2	4.24	8.48	4.24 - 8.48	77	67	1.492537
3	8.48	12.72	8.48 - 12.72	48	45	0.2
4	12.72	16.96	12.72 - 16.96	35	30	0.833333
5	16.96	21.2	16.96 - 21.2	18	20	0.2
6	21.2	25.44	21.2 - 25.44	17	14	0.642857
7	25.44	29.68	25.44 - 29.68	7	9	0.444444
8	29.68	33.92	29.68 - 33.92	4	6	0.666667
9	33.92	38.16	33.92 - 38.16	4	4	0
10	38.16	42.4	38.16 - 42.4	1	3	1.333333
11	42.4	46.64	42.4 - 46.64	0	2	2
12	46.64	50.88	46.64 - 50.88	0	2	2
13	50.88	55.12	50.88 - 55.12	1	1	0
14	55.12	59.36	55.12 - 59.36	1	1	0
15	59.36	63.6	59.36 - 63.6	0	1	1
16	63.6	67.84	63.6 - 67.84	0	1	1
17	67.84	72.08	67.84 - 72.08	0	1	1
18	72.08	76.284	> 72.08	1	1	0
				300		15.0409

Appendix 1.2: Spreadsheet Data for C2 Inspection Time

number of points	300
number of bins	18
max value	114.426
min value	0.13
data range	114.296
bin width	6.349778
sample mean	15.5369
sample variance	215.6443

bin	bin lower range	bin upper range	range	observed frequency O_i	expected frequency E_i	X_0^2 $\frac{(O_i - E_i)^2}{E_i}$
1	0	6.35	0 - 6.35	86	101	2.227723
2	6.35	12.7	6.35 - 12.7	76	67	1.208955
3	12.7	19.05	12.7 - 19.05	49	45	0.355556
4	19.05	25.4	19.05 - 25.4	35	30	0.833333
5	25.4	31.75	25.4 - 31.75	17	20	0.45
6	31.75	38.1	31.75 - 38.1	18	14	1.142857
7	38.1	44.45	38.1 - 44.45	7	9	0.444444
8	44.45	50.8	44.45 - 50.8	4	6	0.666667
9	50.8	57.15	50.8 - 57.15	4	4	0
10	57.15	63.5	57.15 - 63.5	1	3	1.333333
11	63.5	69.85	63.5 - 69.85	0	2	2
12	69.85	76.2	69.85 - 76.2	0	2	2
13	76.2	82.55	76.2 - 82.55	1	1	0
14	82.55	88.9	82.55 - 88.9	1	1	0
15	88.9	95.25	88.9 - 95.25	0	1	1
16	95.25	101.6	95.25 - 101.6	0	1	1
17	101.6	107.95	101.6 - 107.95	0	1	1
18	107.95	114.426	> 107.95	1	1	0
				300		15.66287

Appendix 1.3: Spreadsheet Data for C3 Inspection Time

number of points	300
number of bins	18
max value	104.019
min value	0.031
data range	103.988
bin width	5.777111
sample mean	20.63276
sample variance	394.3862

bin	bin lower range	bin upper range	range	observed frequency O_i	expected frequency E_i	X_0^2 $\frac{(O_i - E_i)^2}{E_i}$
1	0	5.78	0 - 5.78	69	74	0.337838
2	5.78	11.56	5.78 - 11.56	55	56	0.017857
3	11.56	17.34	11.56 - 17.34	41	42	0.02381
4	17.34	23.12	17.34 - 23.12	37	32	0.78125
5	23.12	28.9	23.12 - 28.9	23	24	0.041667
6	28.9	34.68	28.9 - 34.68	21	19	0.210526
7	34.68	40.46	34.68 - 40.46	15	14	0.071429
8	40.46	46.24	40.46 - 46.24	7	11	1.454545
9	46.24	52.02	46.24 - 52.02	10	8	0.5
10	52.02	57.8	52.02 - 57.8	5	6	0.166667
11	57.8	63.58	57.8 - 63.58	3	5	0.8
12	63.58	69.36	63.58 - 69.36	3	4	0.25
13	69.36	75.14	69.36 - 75.14	2	3	0.333333
14	75.14	80.92	75.14 - 80.92	3	2	0.5
15	80.92	86.7	80.92 - 86.7	2	2	0
16	86.7	92.48	86.7 - 92.48	0	2	2
17	92.48	98.26	92.48 - 98.26	0	1	1
18	98.26	104.019	> 98.26	4	1	9
				300		17.48892

Appendix 1.4: Spreadsheet Data for W1 Assembly Time

number of points	300
number of bins	18
max value	29.375
min value	0.007
data range	29.368
bin width	1.631556
sample mean	4.604417
sample variance	22.62537

bin	bin lower range	bin upper range	range	observed frequency O_i	expected frequency E_i	X_0^2 $\frac{(O_i - E_i)^2}{E_i}$
1	0	1.64	0 - 1.64	92	90	0.044444
2	1.64	3.28	1.64 - 3.28	70	63	0.777778
3	3.28	4.92	3.28 - 4.92	41	45	0.355556
4	4.92	6.56	4.92 - 6.56	22	31	2.612903
5	6.56	8.2	6.56 - 8.2	22	22	0
6	8.2	9.84	8.2 - 9.84	16	16	0
7	9.84	11.48	9.84 - 11.48	13	11	0.363636
8	11.48	13.12	11.48 - 13.12	5	8	1.125
9	13.12	14.76	13.12 - 14.76	4	6	0.666667
10	14.76	16.4	14.76 - 16.4	5	4	0.25
11	16.4	18.04	16.4 - 18.04	5	3	1.333333
12	18.04	19.68	18.04 - 19.68	0	2	2
13	19.68	21.32	19.68 - 21.32	0	2	2
14	21.32	22.96	21.32 - 22.96	1	1	0
15	22.96	24.6	22.96 - 24.6	2	1	1
16	24.6	26.24	24.6 - 26.24	0	1	1
17	26.24	27.88	26.24 - 27.88	1	1	0
18	27.88	29.375	> 27.88	1	1	0
				300		13.52932

Appendix 1.5: Spreadsheet Data for W2 Assembly Time

number of points	300
number of bins	17
max value	59.078
min value	0.091
data range	58.987
bin width	3.469824
sample mean	11.09261
sample variance	140.3376

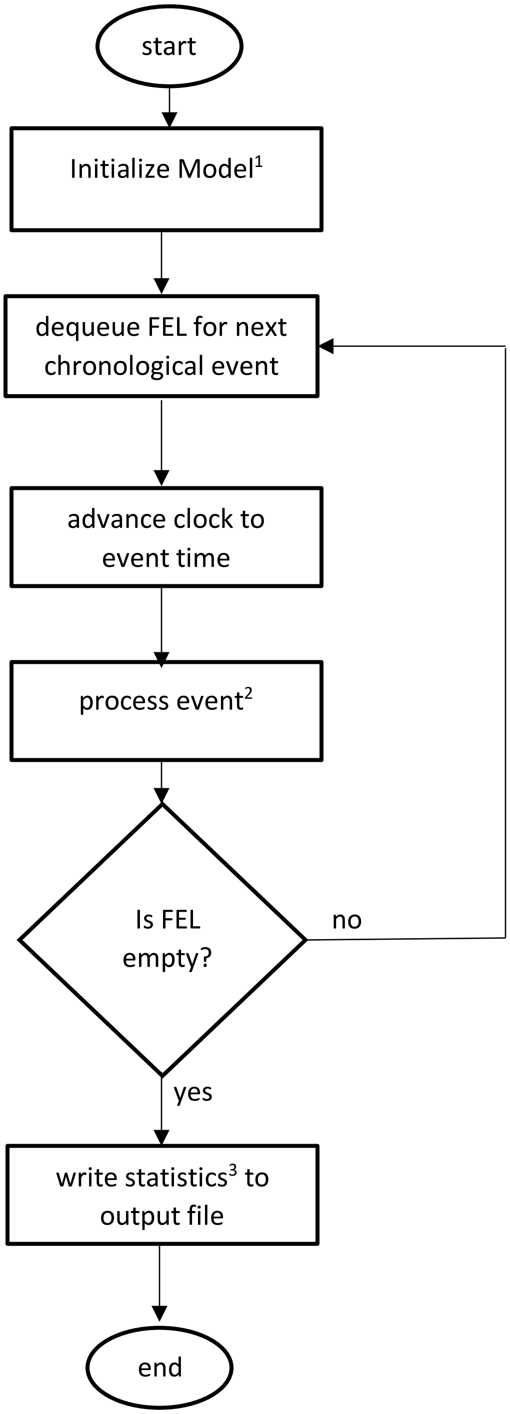
bin	bin lower range	bin upper range	range	observed frequency O_i	expected frequency E_i	X_0^2 $\frac{(O_i - E_i)^2}{E_i}$
1	0	3.47	0 - 3.47	84	81	0.111111
2	3.47	6.94	3.47 - 6.94	63	59	0.271186
3	6.94	10.41	6.94 - 10.41	44	44	0
4	10.41	13.88	10.41 - 13.88	31	32	0.03125
5	13.88	17.35	13.88 - 17.35	22	24	0.166667
6	17.35	20.82	17.35 - 20.82	8	17	4.764706
7	20.82	24.29	20.82 - 24.29	6	13	3.769231
8	24.29	27.76	24.29 - 27.76	11	10	0.1
9	27.76	31.23	27.76 - 31.23	6	7	0.142857
10	31.23	34.7	31.23 - 34.7	4	5	0.2
11	34.7	38.17	34.7 - 38.17	5	4	0.25
12	38.17	41.64	38.17 - 41.64	5	3	1.333333
13	41.64	45.11	41.64 - 45.11	3	2	0.5
14	45.11	48.58	45.11 - 48.58	2	2	0
15	48.58	52.05	48.58 - 52.05	4	2	2
16	52.05	55.52	52.05 - 55.52	1	1	0
17	55.52	58.99	55.52 - 58.99	1	1	0
				300		13.64034

Appendix 1.6: Spreadsheet Data for W3 Assembly Time

number of points	300
number of bins	18
max value	51.418
min value	0.102
data range	51.316
bin width	2.850889
sample mean	8.79558
sample variance	74.82545

bin	bin lower range	bin upper range	range	observed frequency O_i	expected frequency E_i	X_0^2 $\frac{(O_i - E_i)^2}{E_i}$
1	0	2.86	0 - 2.86	77	84	0.583333
2	2.86	5.72	2.86 - 5.72	72	61	1.983607
3	5.72	8.58	5.72 - 8.58	44	44	0
4	8.58	11.44	8.58 - 11.44	29	32	0.28125
5	11.44	14.3	11.44 - 14.3	15	23	2.782609
6	14.3	17.16	14.3 - 17.16	17	17	0
7	17.16	20.02	17.16 - 20.02	13	12	0.083333
8	20.02	22.88	20.02 - 22.88	11	9	0.444444
9	22.88	25.74	22.88 - 25.74	6	7	0.142857
10	25.74	28.6	25.74 - 28.6	2	5	1.8
11	28.6	31.46	28.6 - 31.46	5	4	0.25
12	31.46	34.32	31.46 - 34.32	2	3	0.333333
13	34.32	37.18	34.32 - 37.18	4	2	2
14	37.18	40.04	37.18 - 40.04	0	2	2
15	40.04	42.9	40.04 - 42.9	1	1	0
16	42.9	45.76	42.9 - 45.76	1	1	0
17	45.76	48.62	45.76 - 48.62	0	1	1
18	48.62	51.418	> 48.62	1	1	0
				300		13.68477

Appendix 2.1: Flowchart for General Program Flow

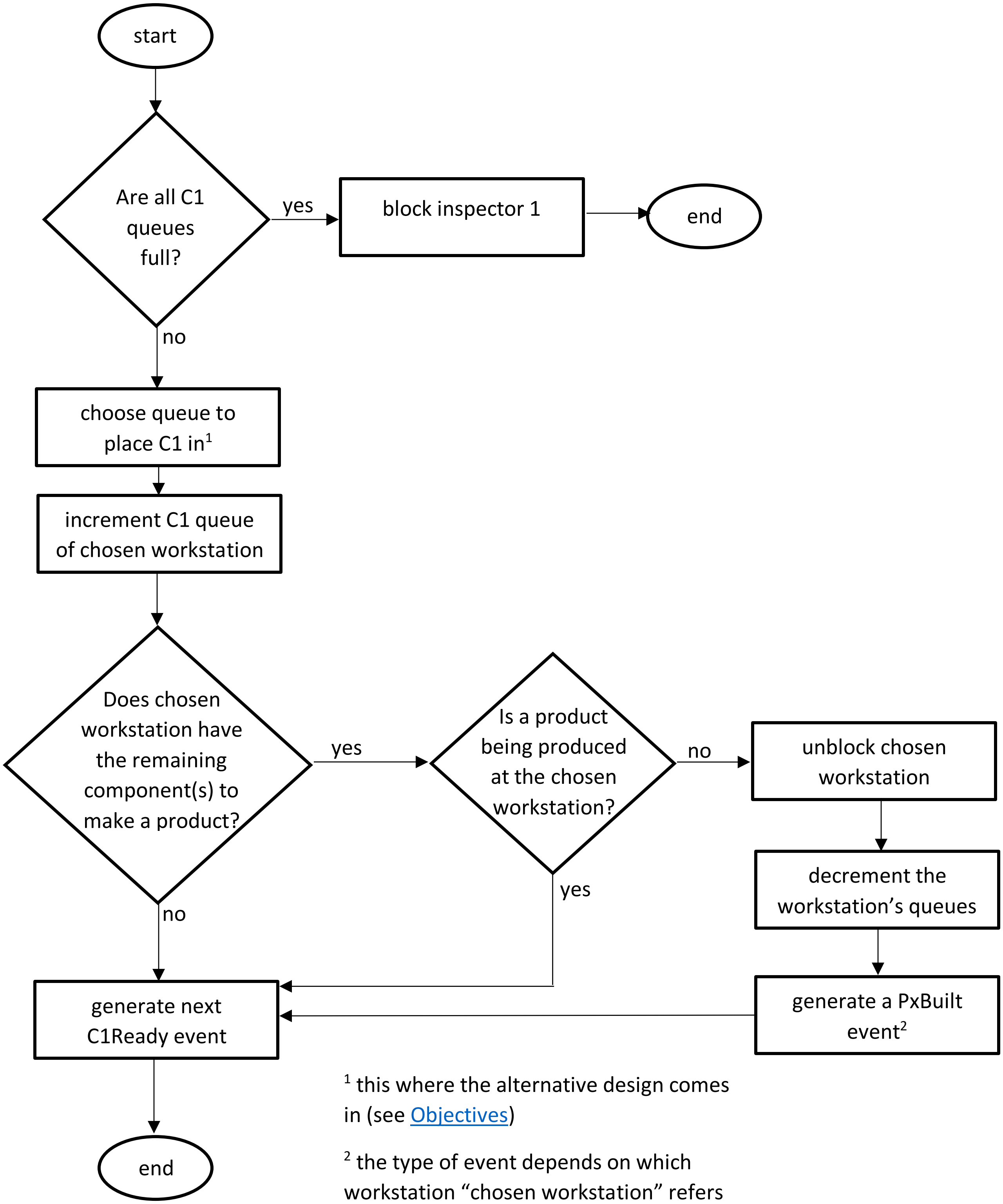


[1] initializing model includes creating distributions, creating the future event list (FEL) with initial event(s), and setting default value for statistics

[2] dependent on type of event, elaborated on in next 6 flowcharts

[3] statistics of interest are simulation time, number of each product produced, and the amount of time each inspector/ workstation spends idle

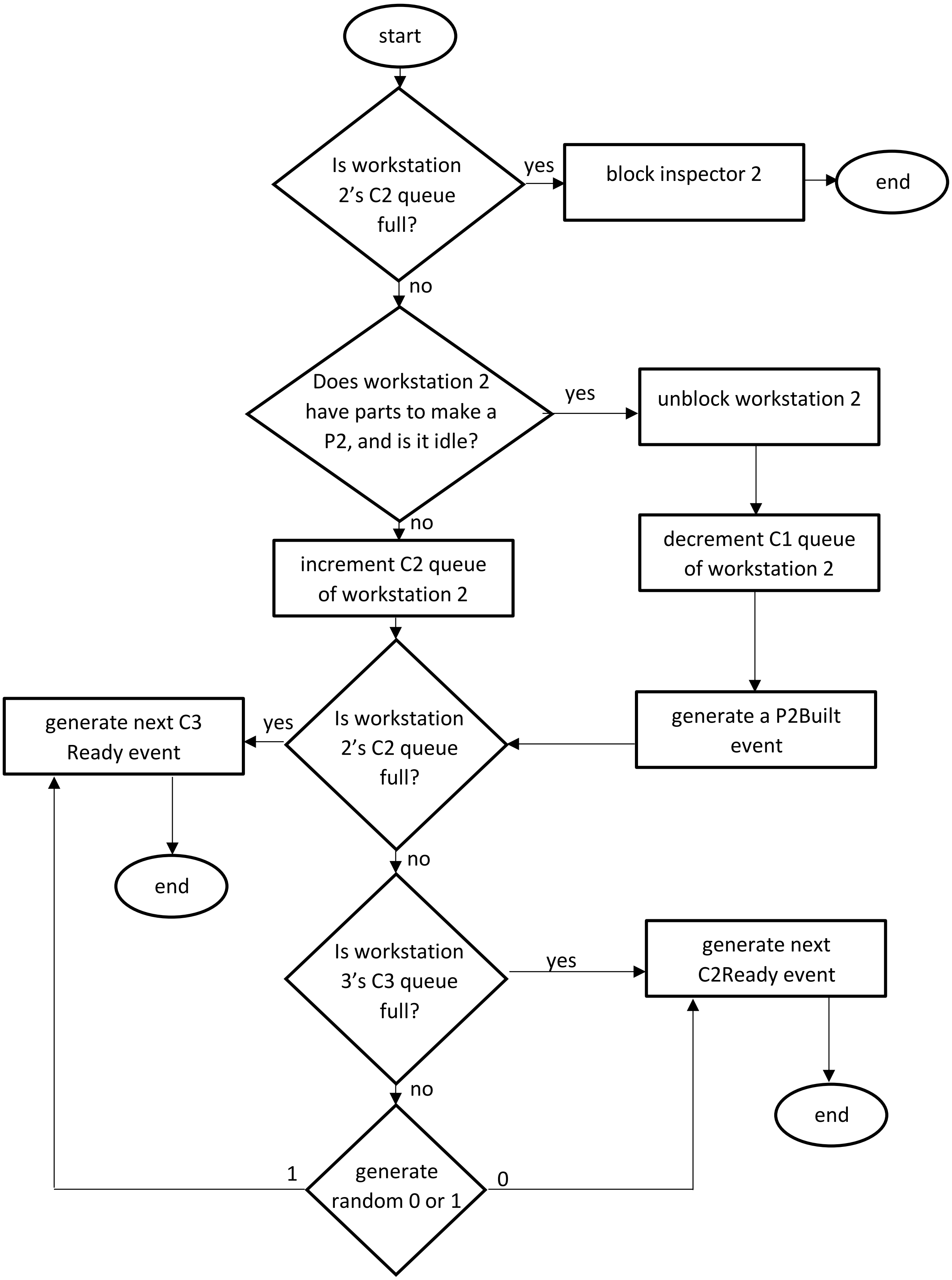
Appendix 2.2: Flowchart for C1Ready Event



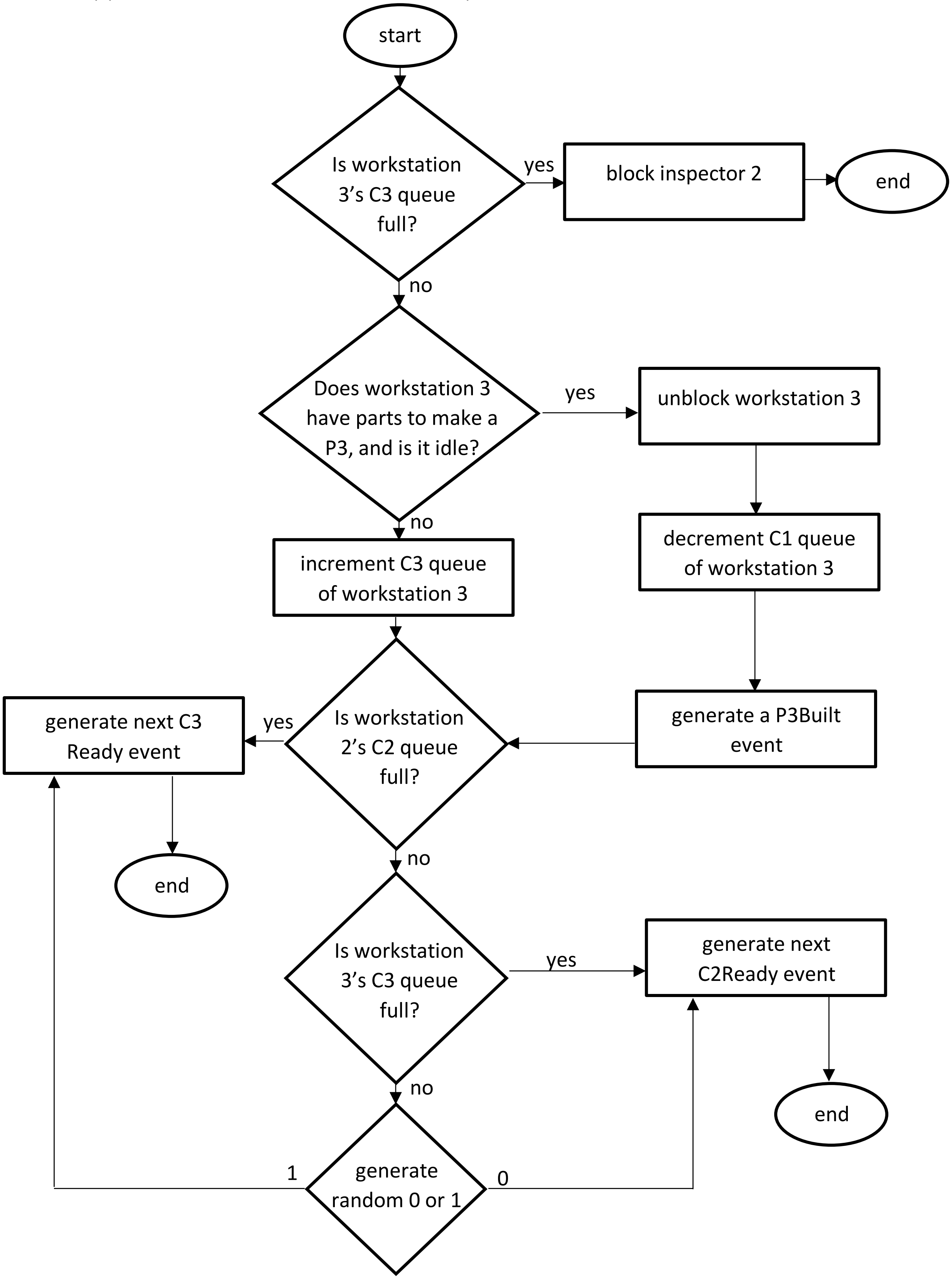
[1] this where the alternative design comes in (see Objectives)

[2] the type of event depends on which workstation "chosen workstation" refers to (P1Built for workstation 1, P2Built for workstation 2, P3Built for workstation 3)

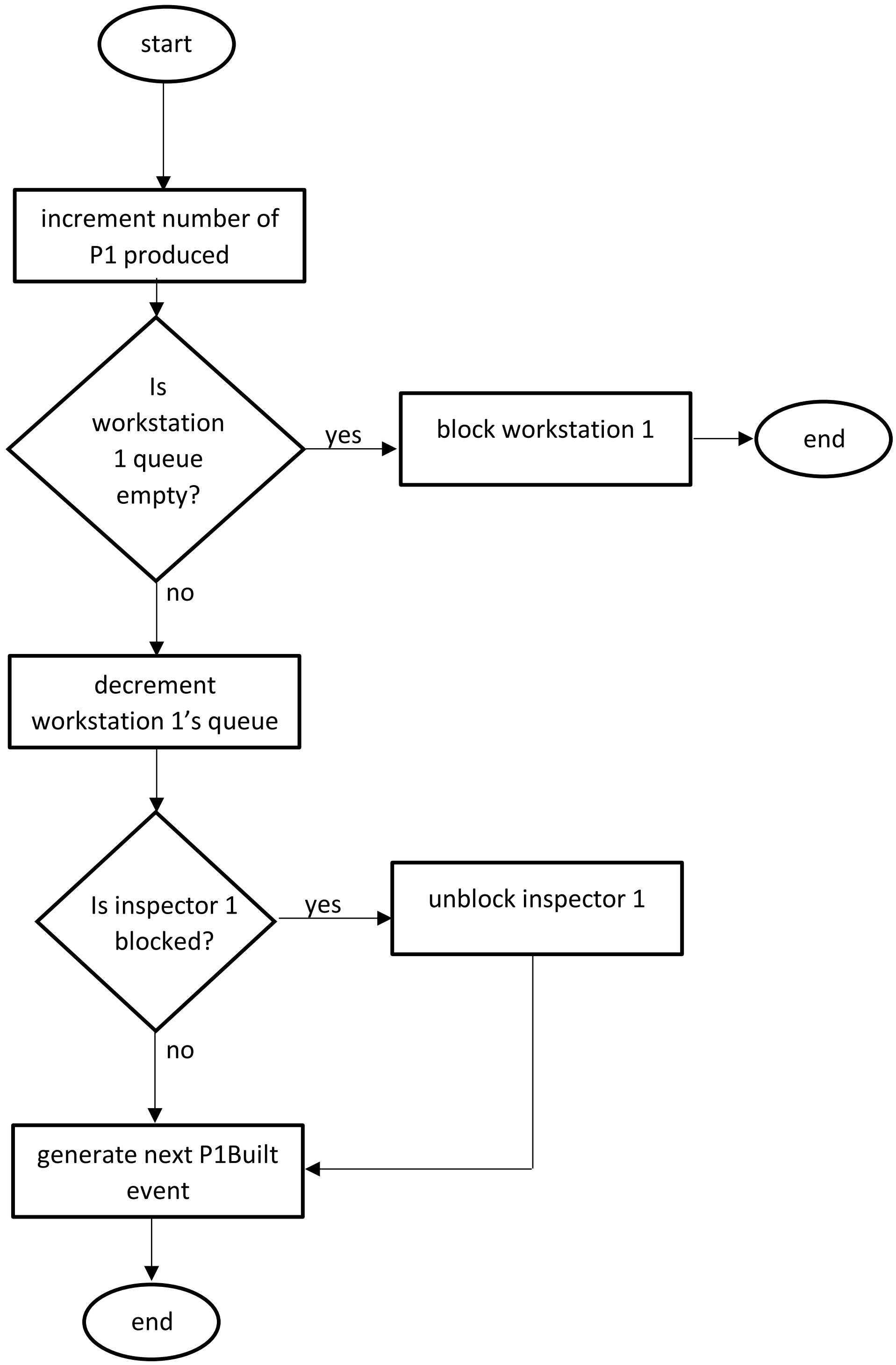
Appendix 2.3: Flowchart for C2Ready Event



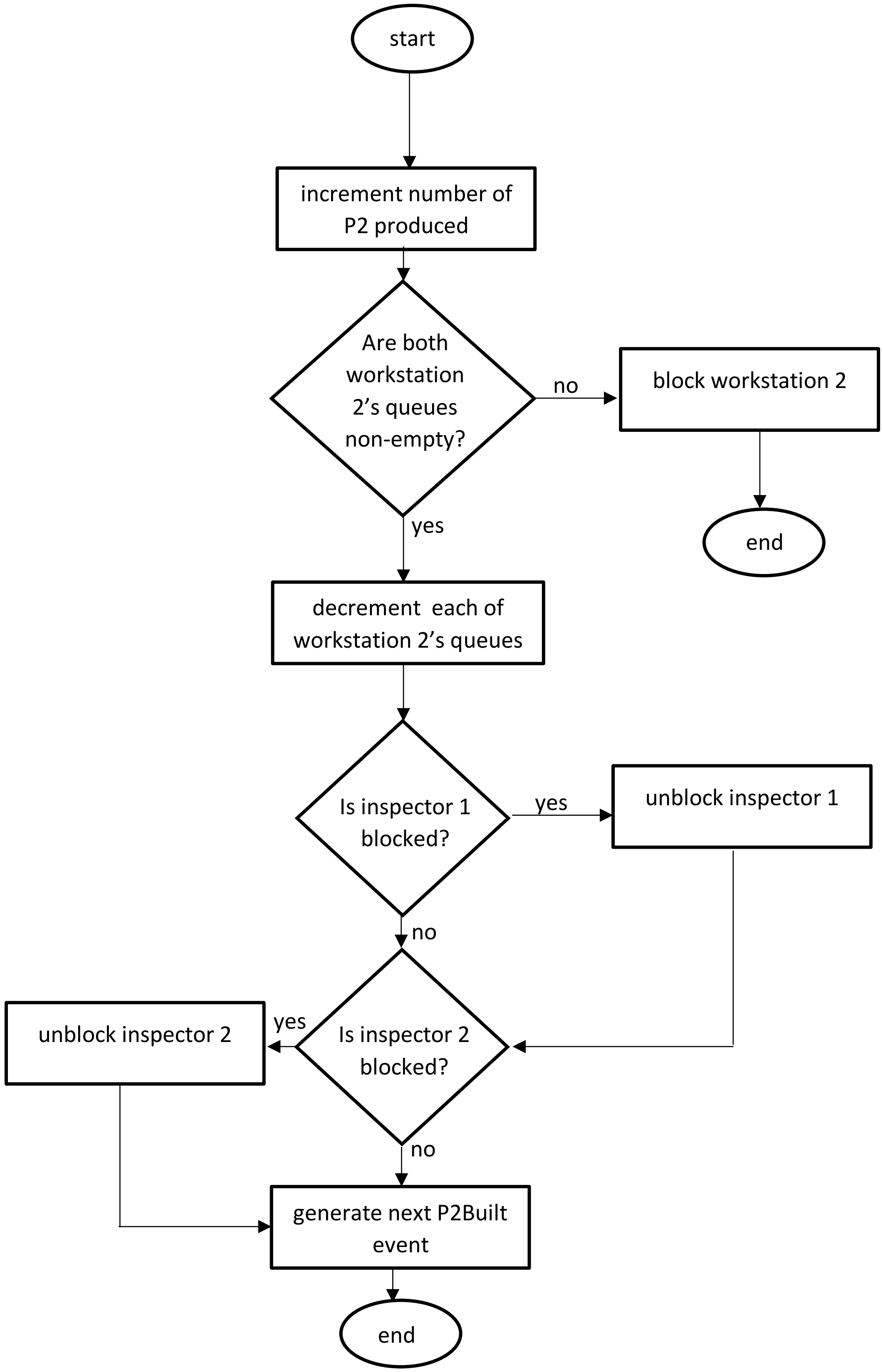
Appendix 2.4: Flowchart for C3Ready Event



Appendix 2.5: Flowchart for P1Built Event



Appendix 2.5: Flowchart for P2Built Event



Appendix 2.6: Flowchart for P3Built Event