Cannibalization in a Repair/Replacement Inventory System

Abstract

In this paper we consider policies for cannibalization in a repair/replacement inventory system where our focus will be the waiting time of a customer in the repair queue. The cannibalization policy will be affected most by the time required to effect the cannibalization and specifically the time required in dis-assembling the functioning components from the newly arrived customer. We will first present a heuristic policy which is calculation intensive and second heuristic which will require far less calculations at each decision point in time. These heuristics, and a comparison between them, will be examined with the aid of simulation.

Key Words:
Cannibalization, Waiting Time, Heuristic, Simulation.

1. Introduction

Consider a repair/replacement system where a customer arrives at a repair queue according to a Poisson process with parameter \( \lambda \). The customer brings an item for repair/replacement which consists of a number of distinct components. Some or all of the components have failed.

On entering the queue, the failed components are immediately removed and sent for individual repair while the customer waits in the queue. We assume that each component is itself repaired in an "ample server" queue with a repair time distribution \( G_i(x) \) (where the subscript "i" refers to component type \( i \)).

When the individual components are eventually repaired they are placed on an appropriate available components shelf. When our customer reaches the front of the queue all the component types that were originally removed for repair are then replaced from the appropriate available components shelves if they are available. If one or more component types are not yet available, then our customer waits until they do become available. Only on complete satisfaction does our customer leave the queue.

Below is a schematic (Figure 1) for the case where each item has exactly two components. (This schematic is adopted from Dreyfuss and Stulman [5]).
Berg and Posner [2] discussed finding the waiting time distribution $W_t(x)$ for the amount of time a new customer arriving at time $t$ will be in the system when the item has a single component. Dreyfuss and Posner [4] extended this to a system where each customer requires a random number of that same single component. Dreyfuss and Stulman [5] extended the original work of Berg and Posner [2] to a situation where each customer arrives in the system with exactly two failed (but different) components. They (Dreyfuss and Stulman [6]) then extended this to a situation where an arriving customer has failure of either one of two component types or simultaneous failure of both components.

This last model naturally presents another question. Suppose that our arriving customer arrives at time $t$ with component A working but component B failed. Further suppose that ahead of him in the queue is a customer who requires a component type A (either because he arrived with a failure of both components or because he arrived with component B working or component A failed). Under which circumstances would the average waiting time $W_t(x)$ in the system be reduced by allowing
cannibalization of the available working components from the newly arrived customer for use in satisfying the needs of the customers ahead of him in the queue. This process is called cannibalization.

Our objective in this paper will be to examine the model in Dreyfuss and Stulman [6] and present a heuristic as to when to allow cannibalization. We will see that when dis-assemble, reassemble and transit times are negligible we should always allow cannibalization. When these conditions are not met then we will present two system dependent heuristics for deciding whether to allow cannibalization of a component from an arriving customer should be allowed. The first heuristic will be more calculation intensive then the second but will give only marginally better results. While in this paper we will focus on the model in Dreyfuss and Stulman [6] where a customer come with at most two failed (but distinct) components the results should apply equally well do models where customers can arrive with up to a greater number of failed (but distinct) components.

Cannibalization has been studied extensively in the past but not with respect to this particular model and not with respect to the effect of cannibalization on the waiting time in the system.

Fisher [7] presented a paper that reviewed many of the issues that are present when considering cannibalization. Hirsch, Meisner and Boll [11] discuss a system which may be in many different states and the effect of cannibalization on the probability of the system being in the different states. Rolfe [14] looked at a special case of the approach developed by Hirsch, Meisner and Boll [11] where failed aircraft can only be repaired by cannibalization (no spares) and the objective was to find the expected number of operating aircraft after a certain number of missions. Hochberg [12] also continued the work of Hirsch, Meisner and Boll [11] but allowed for intermediate states of the operation of system. Simon [16] studied the multicomponent system introduced by Hirsch, Meisner and Boll [11] with an aim to improving the reliability of a system by a judicious policy of cannibalization.

Cheung and Hausman [3] considered a model similar to the one we are examining however their focus was on estimating the distribution and the mean of the number of customers in the queue and not on the effect of cannibalization on the waiting time in the system.

Cannibalization processes often lead to mathematical complexities which are difficult to solve analytically. A number of different authors conducted various different simulations studies on the effects of cannibalization in different situations. Salman, Cassady, Pohl, and Ormon [15] studied the comparison of cost of cannibalization versus maintaining a spare parts inventory. In their model there were a fixed number of machines each with n components and a Weibull failure rate. They did not study the average time in the system or the average time till complete repair. Fisher and Brennen [9] studied a maintenance facility that serviced 32 aircraft each with 6 components that may fail and that were repairable. The times to failure are therefore effected by the number of operating aircraft. They studied different cannibalization policies with regard to backorder levels and the fill rate (the probability of having enough components on hand to effect immediate repair). They did not discuss the effect of cannibalization on the waiting time distribution. Ormon and Cassady [13] did a simulation study of a closed system of n machines each consisting of 2 components. The effect of cannibalization was to minimize investment subject to production capacity constraints as well as the dual of this problem. Fisher [8] presented a simulation of a maintenance system called MNTMOD and its application to the models discussed by Fisher and Brennen [9].
2. Cannibalization and The Original Model

In the original model of Dreyfuss and Stulman [6] there was an assumption that the times to remove or install a component and all transit times were negligible (zero) as compared to all other times. Then if a new customer arrives with a working component it can be removed from an arriving customer and is immediately available to be placed on the available components shelf. From there it can be used in making a repair to the next customer requiring that component to arrive to the head of the queue. We referred to this as Model 1. Alternatively, instead of waiting for a customer to get to the head of the queue to be served, when a component becomes available it can be immediately installed in first customer in the queue requiring that component. We referred to this as Model 2.

A situation where all removal, installation and transit times are negligible is not so farfetched in practice. We might be dealing with electronic components that simply snap in and out and with an on-site repair facility so that the transit times are inconsequential.

Let \( \lambda_i \) be the arrival rate to of a customer with component \( i \) not working and \( \lambda \) be the overall arrival rate to the queue. Let \( G_i(x) \) be the repair time distribution for component \( i \). Then by allowing cannibalization, all we are actually doing is changing the repair time distribution to

\[
G_i^*(x) = \frac{\lambda_i}{\lambda} \cdot G_i(x) + (1 - \frac{\lambda_i}{\lambda}) \cdot 1 > G_i(x)
\]

and all the results in Dreyfuss and Stulman [6] are preserved. That is, the probability of component A being "repaired" in time \( x \) or less has increased. Since components are "repaired" faster it makes intuitive sense that the waiting time distribution of a customer in the system will then decrease.

Let us consider the following example of a two component case. A customer arrives with either one of two components or both components in need of repair. Using the notation of Dreyfuss and Stulman [6], we will refer to these components as components A and B.

Define \( \lambda^{AB} \) as the arrival rate for customers with both A and B in need of repair, \( \lambda^B \) as the arrival rate for customers with A functioning with B in need of repair and \( \lambda^A \) as the arrival rate for customers with B functioning with A in need of repair. Assume that \( \lambda^{AB}, \lambda^A \) and \( \lambda^B \) are independent Poisson arrival rates for the different types of customers. Then

\[
\lambda = \lambda^{AB} + \lambda^A + \lambda^B.
\]

Let the number of original spare in the system be denoted by \( n_A \) and \( n_B \) respectively.

For an example we have chosen \( \lambda^{AB} = 5, \lambda^A = 2, \lambda^B = 1, n_A = 10 \) and \( n_B = 11 \).

We have also chosen the repair time distributions to be \( G_A(x) \sim N(\mu = 1.8, \sigma = .5) \) and \( G_B(x) \sim N(\mu = 2, \sigma = .5) \). All these values where arbitrarily chosen.

Then the waiting time distribution for a randomly arriving customer with and without cannibalization is shown in the following Figure 2. The curve in the graph without cannibalization was calculated using...
the formulas developed in Dreyfuss and Stulman [6]. The curve in the graph with cannibalization used the same formulas with the repair times replace by (1).

![Cumulative Waiting Time Distribution](image)

**Figure 2**

Note the cumulative waiting time distribution curve allowing for cannibalization is everywhere above the equivalent curve not allowing cannibalization. Thus when dis-assemble and transport times are negligible it always pays to allow cannibalization.

For a second example we choose $\lambda^A = 5, \lambda^B = 2, \lambda^B = 1, n_A = 5$ and $n_B = 8$. That is, fewer initial spares in the system. This will greatly increase the waiting times in the system. The resulting waiting time distribution for a randomly arriving customer with and without cannibalization is shown in the following Figure 3.
Under these circumstances ($n_A$ and $n_B$ decreased) the benefit of allowing cannibalization will dramatically decrease the waiting times.

3. Cannibalization Models with Non-Zero Dis-Assembly and Transit Times: Heuristic 1

3.1 Heuristic and Discussion

In the original model ([4] and [6]) the underlying assumption was that the time to disassemble/re-assemble the various functioning components from the arriving customer and all transit times where negligible (zero) as compared to all other times. Under these assumptions we have seen that we should always allow cannibalization. In many situations this will not be the case.

It is intuitively clear that if the time to disassemble/re-assemble the various functioning components from the item and the time in transit are sufficiently large as compared to the time to repair the components it would never pay to cannibalize. It seems that there should be a threshold value for the times to disassemble/re-assemble the various functioning components from the item and transit times below which cannibalization would improve the throughput of the system (lessen the expected waiting time in the queue) and above which it would not.

We will see that the threshold value may be different depending on how much of the complete state of the system we wish to consider. In our analysis the actual complete state of the system when the new customer arrives is very complex. The decision to cannibalize a component may well be
dependent on the state of the queue, the state of repair of each type of component in repair as well as the position of each type of component in transit. We will examine two heuristics that focus exclusively on the component being considered for cannibalization. The first heuristic looks at the complete state of the system as it pertains specifically to the component under question. That includes the state of the queue with respect to that component and the state of the repair facility of that component and those components in transit. In the second heuristic we will propose a decision simply based on the state of the queue with respect to that component without considering the state of the repair facility of that component.

While the coming discussion is valid for customers who may require some of multiple components, we will (for the sake of the simulations presented) restrict ourselves to customers that can require only one or both of two components (A or B).

Without loss of generality our newly arrived customer has arrived with component A functioning and component B failed. At that time, the state of the system with respect to component A will be $\Omega_A$. The state of the system $\Omega_A$ will consist of information about

a. How many previous customers in front of the arriving customer who are in the queue will require a component A.
b. How many previous customers in front of arriving customer in the queue are in the process of disassembling a working component A and the distribution of the times needed to complete those dis-assemblies.
c. How many previous customers in front of arriving customer in the queue are in the process of disassembling a failed component A and the distribution of the times needed to complete those dis-assemblies.
d. How many components A are in transit to the component A repair facility and for how long they are already in transit.
e. How many components are in transit from the component A repair facility and for how long they are already in transit.
f. How many of these components are currently in repair and the distribution of times till those repairs are completed.

Clearly the state of the system $\Omega_A$ can be very complex and depends on many parameters.

Based on $\Omega_A$ we would wish to make a decision as to whether to cannibalize the new customer's working component A.

If our new customer's cannibalized component A will, in fact, be used to replace a missing component A for some customer ahead in the queue, then our new customer will himself have to wait for a replacement component A that would have been originally used further up the queue. But in a Fifo system our new customer would have waited at least that long anyway. We would therefore "expect" the average waiting time to decrease.

We are proposing that the decision should be made by considering the last customer in the queue who is in front of our arriving customer and who requires a component A. If the total expected amount of time till that customer gets that component A replaced is decreased if we cannibalize our arriving
customer's component A then we should do so. Otherwise we should not cannibalize the component A from our arriving customer.

Let us assume for the moment that the dis-assemble times and transit times are all deterministic. Then we would never choose to dis-assemble component A from our newly arriving customer for the benefit of another customer in the queue in front of him that is in himself in the process of dis-assembly of a working component A that we had previously decided to dis-assemble and cannibalize. This is so because that customer is ahead of our new customer in the dis-assembly process and can therefore be sooner satisfied with its own component A. This should be true even if the dis-assembly time is probabilistic so long as the mean remaining time to dis-assemble after dis-assembly has begun is a non-increasing function of time. This is almost always the case.

If the time to dis-assemble has an exponential distribution the expected remaining time to dis-assemble will be constant and equal to the expected dis-assembly time for component A of our new customer. So again we need not choose to dis-assemble component A from our new customer. Therefore, we would (almost) never make the decision of dis-assembling working component A from our newly arriving customer for the benefit of another customer ahead in the queue with working component A in dis-assembly.

3.2 Heuristic1 an Example

In this example all times are deterministic (other than the arrival times to the queue). Let the dis-assemble time and the reassemble time each equal 1. Let the transportation times in each direction each equal 0. Let the repair times be equal 2.

Our new customer arrives at the queue and observes \( \Omega_A \) as per Figure 4. In Figure 4 the newly arriving customer has component B failed (as indicated by the X) and component A functioning (as indicated by the \( \sqrt{\cdot} \)). He observes that there are ten customers in the queue in front of him (Q1-Q10) each with their own set of X's and \( \sqrt{\cdot} \)'s. He also observes Repair Facility A where there are eight components under repair (R1-R8) and the elapsed time that has passed since each of those components began repair.

He will now make the following calculation.

The component in R1 has essentially completed its repair and requires a time of 2 to transit back to the queue and satisfy customer Q1.

The component in R2 has will complete its repair in a time of 1 and requires a time of 2 to transit back to the queue for a total of 3. It will satisfy customer Q2.

The component in R3 has will complete its repair in a time of 2 and requires a time of 2 to transit back to the queue for a total of 4. It will satisfy customer Q4.

The component in R4 has will complete its repair in a time of 2.9 and requires a time of 2 to transit back to the queue for a total of 4.9. It will satisfy customer Q6.

The component in R5 has will complete its repair in a time of 3 and requires a time of 2 to transit back to the queue for a total of 5. It will satisfy customer Q8.
The component in R6 has will complete its repair in a time of 4 and requires a time of 2 to transit back to the queue for a total of 6. It will satisfy customer Q9.

Since the dis-assemble time of our new customer (Q11) is 1 which is less than 6 we would decide to cannibalize component A from Q11 and use it to repair the missing component A closest to the front of the queue. The remaining components under repair will then eventually affect repairs in the order of the queue and eventually also "repair" Q11.

At each new customer arrival with a functioning component A we would then decide to cannibalize that component A (or not) based on a similar calculation. While we have only discussed component A in the forgoing analysis the very same process would apply to component B (and if there were multiple components it would apply to each one individually).

### 3.3 Heuristic1 Simulation

Below are the results of a simulation (Figure 5). The simulation was done using Arena. The simulation was repeated 10 times each for 70 hours (the first 10 hours were considered warm-up times to simulate steady state results. The parameters were as above except that we allowed the dis-assemble time to vary.

Figure 5 show the results of the simulation allowing for cannibalization as per Heuristic1 as well as the results for the policy of always cannibalizing and a policy of never cannibalizing (using the formulas found in [6]). The results show that our Heuristic1 cannibalization always performed better (lower average waiting times in the system) than either a policy of never cannibalizing or always cannibalizing. As expected when dis-assemble time were low our heuristic gave the same average waiting times as
the policy of always cannibalizing. When the dis-assemble times were relatively large our heuristic gave the same average waiting times as the policy of never cannibalizing.

![Performance of cannibalization algorithms](image)

Our example assumed that all the times were deterministic. This need not be the case. When times are probabilistic then expected times and mean residual times can be used.

### 3.4 Heuristic1 Further Discussion

Suppose that the repair times were probabilistic having some repair time distribution while the dis-assemble time, assemble time and transit times are approximately deterministic (as may often be the case). Then at each decision moment when we need to decide whether to cannibalize a component from a newly arrived customer we could replace the deterministic residual time of each repair with the expected residual time of each component being repaired. If for example, the repair time distributions are Normal, Gamma or Lognormal densities then efficient formula for the mean residual times can be found in Govil and Aggarwal [10]. A general formula for the mean residual time can also be found in Govil and Aggarwal [10] as well as in many other sources. While there a positive probability that a component under repair will finish that repair earlier than a component who may have stared its repair earlier, it is generally unlikely and certainly not in expectation.

An important exception to the last remark would be an exponential repair time distribution. If the repair time densities are exponential the mean residual time will just be the mean of the exponential distribution itself. By the lack of memory property of the exponential distribution all the components that are under repair will have the same residual distribution until their repairs are completed. Thus the expected residual times of repair will not be ordered. Under this circumstance all the items under repair will be tied in expectation as to which of them finishes first, second, etc. In this case, if there...
are n items under repair and there are k customers needing the component in the queue in front of the arriving customer then the expected repair time of the item that will satisfy the k-th customer in the queue will just be the expected time of the k-th order statistic of the exponential distribution. A formula for this can be found in Balakrishnan and Cohen [1].


4.1 Heuristic2 and Discussion

In the first heuristic, when a new customer arrives with a component that can be cannibalized we evaluate the entire state of the system as it pertains to that component. This was referred to as $\Omega_1$. This includes the customers in the queue in front of the arriving customers as well as components of the appropriate type that are in repair or in transit. Based on $\Omega_1$, a decision is made to cannibalize component $i$ or not. We have seen above that this could be quite complicated and may be calculation intensive. For most repair time distributions, we will have to calculate the mean residual times for each component under repair. Each such calculation may require at least one numerical integration. Thus the actual calculations themselves may become quite cumbersome (but not impossible).

The question that we wished to examine, was how much does the portion of the information dealing with the status of the components under repair effect the decision to cannibalize a component from a newly arrived customer. How much would be lost if the decision to cannibalize or not was solely based on the information about the queue in front of the arriving customer and not on the status of components under repair.

We have already established that when the disassemble time is very low (and assuming transit times within the queue itself is zero or close to it) then we should always take the opportunity to cannibalize. When the disassemble time is very large it should never pay to disassemble. Since there is an average (steady) state of the repair status of the components it makes sense that for every state of the various parameters there should be some number $K$ such that if there are $K$ or above customers in the queue in front of our newly arrived customer that can use the cannibalized component it would pay to cannibalize. If there are less then $K$ such customers it would not pay to cannibalize.

4.2 Heuristic2 Simulations and Discussion

We ran a simulation using Arena for a number of different parameter sets. The simulation was repeated 10 times each for 70 hours (the first 10 hours were considered warm-up time to simulate steady state results). In all cases the pattern of results indicated that the waiting time distribution as a function of $K$ had a U shape with a clear minimum. This minimum should define the policy $K$ for when to cannibalize or not.

For example, consider the set of parameters as above (dis-assemble time=1). Simulations were performed for various values of $K$ and produced the following set of average waiting times (Figure 6).
We can see that for this set of parameters the average waiting times forms a U function with the minimum value at K=3. Thus a policy of cannibalizing whenever the number of customers in the queue requiring the component is three or more will be the best K type policy.

Now consider the very same set of parameters but allowing the disassemble time to vary. Simulations were run (as before) for an assortment of disassemble times. The Figure 7 shows the resulting step function of optimal policies K for the varying disassemble times.

As expected, the values of K are an increasing function of the disassemble times.

Finally, simulations were performed for various disassemble times to get the average waiting times based on Heuristic1 and similarly based on Heuristic2 (where in each case the optimal K was used). The Figure 8 shows the various results.

The curves produced by each of the two Heuristics are very close to one another. As expected when the disassemble times are either very small or very large the results are identical. However, for intermediate values Heuristic1 is lower (smaller average waiting times) then the results of Heuristic2. This is to be expected because Heuristic1 uses more information from the state of the system than does Heuristic2. However, the results are everywhere very close. Thus it appears that the advantage of Heuristic1 is only marginally better then Heuristic2.

Figure 6

![Performance of Heuristic 2 algorithm](image)

**Performance of Heuristic 2 algorithm**

<table>
<thead>
<tr>
<th>Number</th>
<th>Performance of Heuristic 2 algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.12</td>
</tr>
<tr>
<td>1</td>
<td>1.11</td>
</tr>
<tr>
<td>2</td>
<td>1.10</td>
</tr>
<tr>
<td>3</td>
<td>1.10</td>
</tr>
<tr>
<td>4</td>
<td>1.09</td>
</tr>
<tr>
<td>5</td>
<td>1.08</td>
</tr>
<tr>
<td>6</td>
<td>1.07</td>
</tr>
<tr>
<td>7</td>
<td>1.06</td>
</tr>
</tbody>
</table>

Average Waiting Time
Figure 7

Optimal decision of cannibalization algorithms

Performance of cannibalization algorithms
5. Final Remarks

In this study we examined a model for repair/replacement of multiple failed components for and arriving customer at a repair queue. We saw that when dis-assemble, assemble and transit times are negligible (as compared to component repair times) then a policy of component cannibalization should be instituted. However, when these times are not negligible then there is a threshold value of dis-assemble and assemble times below which cannibalization should be allowed and above which it should not be allowed.

These threshold values will be a complicated function of the state of the system when the new customer arrives at the queue. We examined two heuristics for the threshold values. One examined the entire state of the system only with respect to the component whose cannibalization is under consideration (the state of the queue, the state of components under repair and the state of components in transit). The second heuristic, which is less calculation intensive, only examines the state of the queue with respect to the component being considered for cannibalization. The second heuristic performed only slightly worse than first heuristic.

All of our simulations assumed that the arriving customer has failure of only up to two components. This allowed us to compare the results with the results on [6]. However, the logic of the two heuristics would be the same in an environment of more than two components. It may be of interest to study this "larger" model.

References


