IE 602: Course Project Report

Cheran Transport Corporation (CTC): Management of Maintenance Spares

Jigar Panot (163196001), Shiksha Singhal (183190001)

Kapsikar Suyog Pramod (183190008), Sumit Jain (183190011)

Key Terms:

1. **Fleet Size -** The number of buses with Cheran Transport Corporation.

2. **Float Size -** The total number of aggregates dedicated in the system for the purpose of exchange or replacement on a one-to-one basis. We assign it as a percentage of fleet size.

3. **Aggregates -** Also known as **assemblies**, consists of large number of components which can be replaced.

4. **Service Level -** Service Level measures the performance of a system. Certain goals are defined and the service level gives the percentage to which those goals can be achieved. For our system, we define service level *(SL)* as:

$$SL = \frac{Number\ of\ aggregates\ replaced}{Total\ demand\ of\ aggregate\ replacement}$$

Introduction

CTC is a road transport corporation with 24 branches, 3 divisional offices, 1 reconditioning unit (RCU) and a tyre retreading plant. The RCU is responsible for maintenance activities needed for efficient running of vehicles owned by the Corporation

CTC has a fleet size of 1412 buses and runs about 460 thousand kilometers per day. Such a substantial fleet calls for occasional replacement and maintenance of aggregates.
Earlier, there were two locations in CTC where the aggregates were stocked:

1. At branches for exchange in vehicle when failed.
2. At RCU for exchange when branches send failed aggregates for reconditioning.

Now, CTC has decided on a new policy regarding change in stocking location of maintenance spares, i.e., stocking location is changed from branches (24 in number) to divisional offices (3 in number) with the belief of increasing service level at the same float size and reducing cross-movement of spares between branches and reconditioning unit (RCU). A validation study is required to provide quantitative basis for this belief.
In addition to this, the float size has been fixed to be 10% of fleet size by CTC based on past experience. We enquire this number and would like to come up with a formal study to obtain optimal float size. Therefore,

The aim of this study is to:

1. Record changes in service level with changes in float size and find the optimal float size.

2. Compare the old policy with new policy in terms of service level at the pre-defined float size (10%).

To study above mentioned objective, we perform a **discrete event simulation**.

Discrete Event Simulation

Ideally, we should simulate for continuous time, however since the demand is sparse, we discretize the simulation and simulate parameters on a weekly basis. We simulate the demand assuming that demand comes at the beginning of each week. This consideration of a week as a time step stands valid because the commute for exchange of aggregates from branches (as per the old policy) or divisional offices (as per the new policy) to RCU happens once a week (as given in the case study).

It may be noted that although the management is easier when spares are stocked at branches but here our main focus is achieving maximum service level for a given float size.

Problem Formulation:

We model the problem as a *discrete event simulation*. We utilize the data for empirical demand given in Table 1 (see appendix) to generate stochastic demand. This demand and the turn-around time is utilized to calculate inventory levels at each week. In general, to have data sufficiency, we only consider Alternator, Gear Box and Self Starter for our analysis.

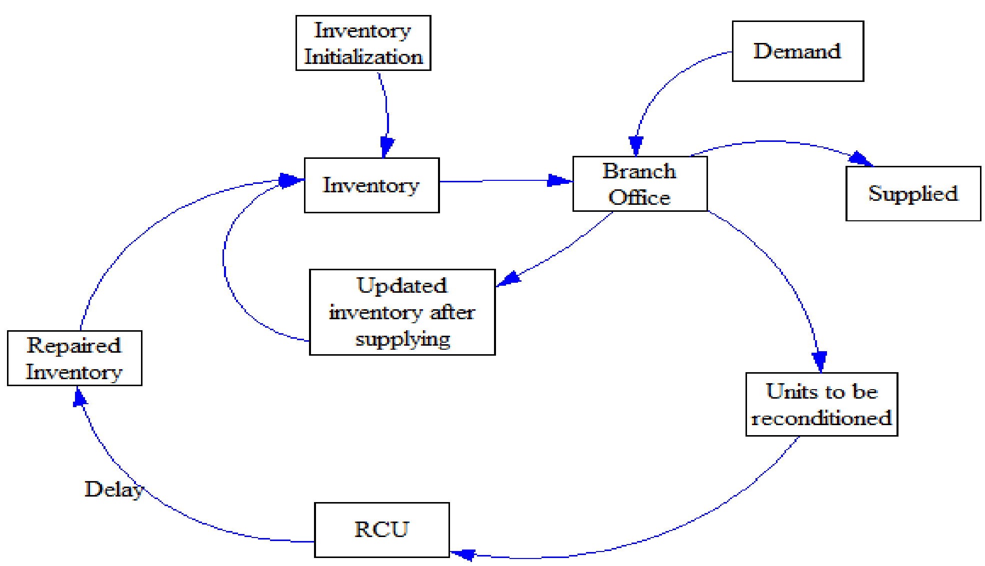

Figure 1: Old Policy

Figure 1 shows flow of aggregates as per the old policy. When the demand comes at branch, the aggregates are replaced from inventory and the faulty aggregates are sent to RCU for repair and maintenance. After a certain processing delay, the aggregates are sent to branch whose inventory is now updated as inventory left after fulfilling demand and processed aggregates sent from RCU. It may be noted that the demand and turn-around time is stochastic in nature. Therefore, we simulate the policy to identify average service level and also account for worst case scenarios.

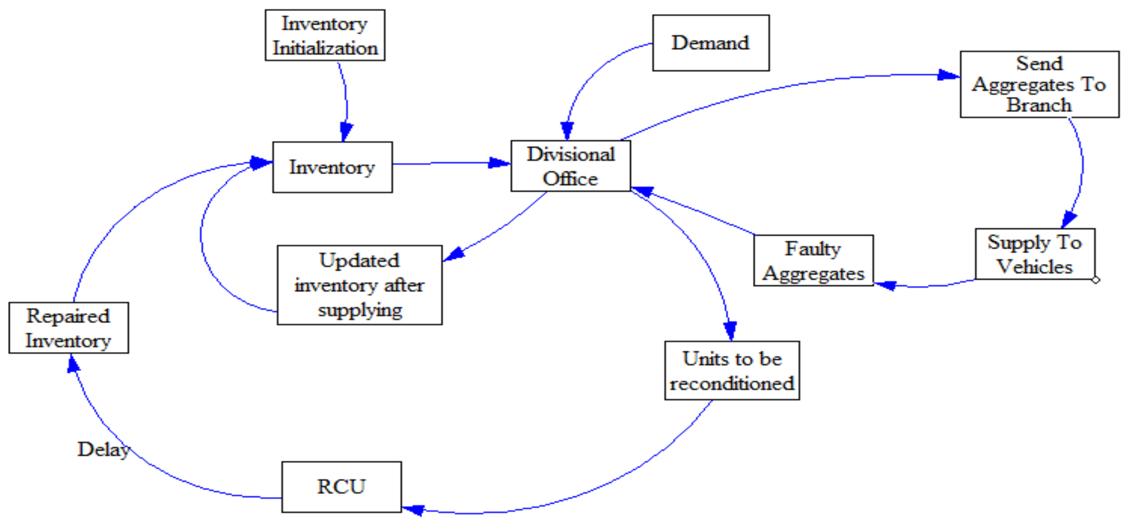

Figure 2: New Policy

Figure 2 shows flow of aggregates as per the new policy. When the demand comes at branch, the requested aggregates are sent from divisional offices to branches which are then replaced. The faulty aggregates received at the branches are also sent to divisional offices which then sends it to RCU for repair and maintenance. After a certain processing delay, the aggregates are sent to divisional offices whose inventory is now updated as inventory left after fulfilling demand and processed aggregates sent from RCU. Note that the processing delay is stochastic, similar to old policy owing to the stochastic nature of RCU repair and maintenance time.

A more detailed Mathematical model has been provided in the appendix.

Demand Generation:

1. **Old Policy**: Past demand data for each aggregate at each branch was given in Table 1 (data for 6 months). We consider that as mean demand for 26 weeks, to obtain the mean demand per week. Using that we simulate demand per week with Bernoulli distribution for each aggregate at each branch (see below).

2. **New Policy**: Similar to Old Policy, we simulate demand per week for each aggregate at each branch. The demand of all the branches that fall under control of same division in order to are then added to obtain the cumulative demand for each division.

3. **Example for demand generation:**
   A. For MFL I, demand of alternator for 6 months is given as 47 alternators
      Therefore,

$$\text{Weekly demand} = \frac{47}{26} > 1$$

      We generate 6 Bernoulli random numbers (one for each day) with mean $\frac{47}{(26*6)}$ and added them to get weekly demand.
      i.e.

$$p = \frac{47}{(26*6)}$$

$$\text{demand} = \begin{cases} 1 & \text{with probability p} \\ 0 & \text{with probability (1-p)} \end{cases}$$

   B. For MFL II, demand of alternator for 6 months is given as 10 alternators
      Therefore,
      Weekly demand = $\frac{10}{26} < 1$
      Thus, we generate a Bernoulli random number with mean $\frac{10}{26}$ to get the weekly demand.

$$p = \frac{10}{26}$$

$$\text{demand} = \begin{cases} 1 & \text{with probability p} \\ 0 & \text{with probability (1-p)} \end{cases}$$

Assumptions -

1. All the demand comes at the beginning of the week

2. Transit of aggregate between branch and divisional office is readily available with negligible transit time

3. Working week is assumed to be of 6 days.

Simulation Validation

We validate the simulation in two ways:

1. Objectively, with increase in float size service level increases for both old and new policies.

2. By hand calculation, we simulate the policies for 3 weeks, printed demand, inventories at branch and RCU. (Given initial inventory at branch) we calculate inventory at branch for remaining duration.

The simulation passes both test successfully. In addition to this, as a best practice, we have tried to write the code with modular functions such that the output at each step can be studied and verified with the hand calculations.

Results

We run the model for the duration of 26 weeks i.e. half a year with float size as 10% of fleet size, we found following service levels averaged over 100 sample paths

$$\begin{aligned} Service\ level_{old\ policy} &= 0.828 \\ Service\ level_{new\ policy} &= 0.955 \end{aligned}$$

We also wish to find the optimal float size, therefore we simulate the old and new policy for different values of float sizes.
Following are the service levels of old and new policies for different float sizes.
It can be seen in Figure 3 that the service level increases monotonically with increase in float size (also acts as validation for the simulation). In addition to this, the service level for both policy can be compared, it can be seen that for a given service level, we require more float in old policy than in the new one.

A service level of near 100% is achieved for the old policy at about 45% float. On the contrary, new policy requires only about 20% float.

Table 2 below summarizes the result of our simulation (Calculated by taking average over 50 sample paths and for duration of half a year)

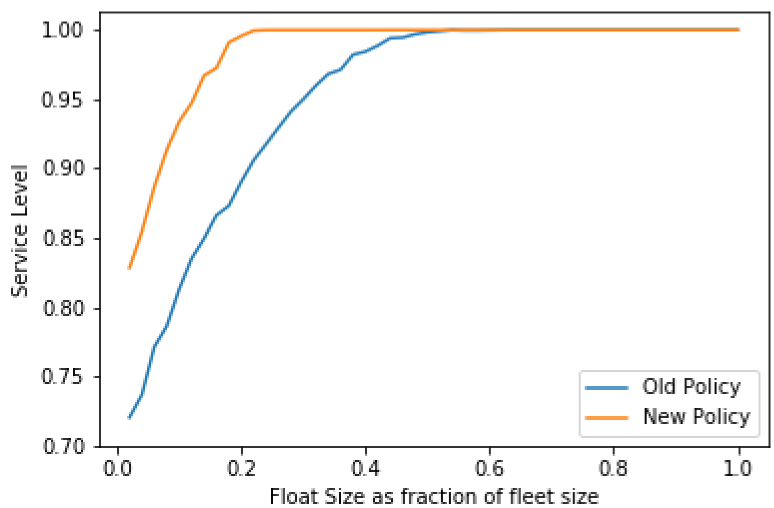

Figure 3: Service Level v/s Float Size

Table 2: Percentage Float Size v/s Service Level of Old and New Policy

Percentage Float size	Service Level Old Policy	Service Level New Policy
5	0.7802	0.8998
10	0.8311	0.9535
15	0.8732	0.9887
20	0.9116	0.9986
25	0.9377	1.0000
30	0.9613	1.0000
35	0.9772	1.0000
40	0.9891	1.0000
45	0.9964	1.0000
50	0.9989	1.0000
55	0.9999	1.0000
60	1.0000	1.0000
65	1.0000	1.0000

Conclusions

The simulations were carried out using the empirically available data as the basis. We found that

1. For a given float size, **a better service level can be achieved for the new policy**. Intuitively, this makes sense because we find a balance between centralization (one headquarter) and complete decentralization (branch wise). The float deficit at one branch can now be fulfilled by the surplus at another branch.

2. Although the service level has increased for the new policy, it brings **additional complexity** of distributing the repaired aggregates amongst different branches. In addition to this, there is additional transportation from division to branches. However, since the distances between branches and divisional office are small, we can ignore this additional complexity and assume commute is readily available.

3. A service level of 95% can be achieved for the new policy for a float size of 10%. However, this float size gives only about 83% service level for the old policy. For both policies the service level increases monotonically with increase in float size. We **propose a float size of 10%** while using new policy.

4. We also simulated the unfulfilled demand for each policy for every week. For new policy, the worst case unfulfilled demand (division level) is 8 aggregates (Alternator). For old policy, the worst case unfulfilled demand (branch level) is 6 aggregates (Alternator). However these numbers cannot be compared directly since a division serves more than one branch. Therefore, we calculate the variability in unfulfilled demand which is as follows:

   Old policy - 1.27 (Worst Case Sample Path)

   New Policy - 0.72 (Worst Case Sample Path)

   Therefore, we can say that the **new policy results in smoother operations.**

APPENDIX

Table 1 : Branch related statistics									
	Branch Code	Number of vehicles	Scheduled kilometres per month	Aggregates					
				Alternator		Gearbox		Self Starter	
				R	I	R	I	R	I
Division – 1	MFL I	82	991349	47	47	31	30	70	66
	MFL II	35	503955	10	10	9	9	11	11
	TS-I	77	782564	58	61	20	19	62	65
	TS-II	69	761126	64	63	48	50	91	89
	TS-III	71	760960	67	65	42	36	73	73
	MM-I	24	265562	25	26	13	19	26	20
	UKM	69	703160	8	11	3	3	4	8
	MTP-I	42	458809	57	61	34	35	73	67
	MTP-II	54	608827	2	7	1	2	3	10
	OPR	72	776861	50	50	30	29	82	77
Division – 2	POY-I	56	614600	35	37	36	32	66	62
	POY-II	70	913521	45	46	13	12	72	74
	POY-III	51	546648	31	29	24	25	43	44
	VLP	41	319381	27	31	30	34	34	36
	UDT	75	886876	46	47	19	20	64	66
	PLN	25	368255	7	10	8	9	32	10
	PDM	34	389013	17	13	10	10	16	14
	KMP	35	378352	4	4	1	2	5	7
	TPR	93	1138980	69	67	30	30	68	64
Division – 3	OOTY-I	73	520976	25	31	47	44	65	60
	OOTY-II	75	558639	45	37	48	39	46	45
	CNR	77	541764	57	59	42	46	70	73
	KTG	60	408313	26	25	49	49	44	37
	GDR	51	366394	40	32	31	29	40	37
	Total	1294	1,45,64,844	R – Receipt I– Issue (Cumulative data for six months)					

<u>Mathematical Model</u>

Variables:

IB_{ijt} = Inventory of j^{th}aggregate at i^{th} branch at time t

IRP_{jt} = Processed Inventory of j^{th}aggregate at RCU at time t

$IRUP_{jt}$ = Unprocessed Inventory of j^{th}aggregate at RCU at time t

D_{ijt} = Demand of j^{th}aggregate at i^{th} branch at time t

O_{ijt} = Total cumulative order of j^{th}aggregate at i^{th} branch at time t

UD_{ijt} = Unfulfilled demand at i^{th} branch of j^{th}aggregate at time t

MH_{jt} = Man hours available per week (6 days) of j^{th}aggregate at time t

ST_{jt} = Standard time for reconditioning of j^{th}aggregate at time t

wA_{ijt} = Weight of j^{th}aggregate at i^{th} branch at time t

Rem_{jt} = Remaining aggregates at RCU of j^{th}aggregate at time t

b_{ijt} = Number of allotted j^{th}aggregate to i^{th} branch at time t

SL = Service Level

Algorithm (Old Policy):

Figure 1 briefly explains flow of the inventory and the basis on which this algorithm runs

1. Start
2. Float Size is read from **csv** file.
3. Initial Float Size = Fleet Size * Some Percentage (for ex 10%)
4. Begin **for loop** for $t >= 1$
5. Begin **for loop** for $i = 0 \ to \ 24 \ \forall j$
6. Simulate demand D_{ijt}
7. Update IB_{ijt} = max{$IB_{ij(t-1)}$ - $D_{ijt}, 0$ } $\forall i, j$
8. Update O_{ijt} = $O_{ij(t-1)}$ + D_{ijt} $\forall i, j$
9. Update UD_{ijt} = max{D_{ijt}-$IB_{ijt}, 0$} $\forall i, j$
10. End **for loop**
11. Update $IRUP_{jt}$= $IRUP_{jt}$+ $\sum_i \ D_{ijt} \ \forall j$
12. Update IRP_{jt}= IRP_{jt}+ min($floor(MH_{jt}/ST_{jt}), IRUP_{jt}) \ \forall j$
13. Update $IRUP_{j(t+1)}$= $IRUP_{jt}$- min($floor(MH_{jt}/ST_{jt}), IRUP_{jt}) \ \forall j$

   Allotting spares to each branch in the ratio of their demands

14. Update wA_{ijt} = $O_{ijt}/ \sum_i \ O_{ijt} \ \forall i, j$
15. Update Rem_{jt}= $MH_{jt}/ST_{jt} - floor(\sum_i \ wA_{ijt}$*$MH_{jt}/ST_{jt}) \ \forall j$
16. Update b_{ijt} = $floor(\sum_i \ wA_{ijt}$*$MH_{jt}/ST_{jt}) \ \forall i, j$

Since we are using floor function, so RCU may still have some processed aggregates, so allocating them sequentially to each of the branches whose $wA_{ijt} > 0$

17. Begin **while loop** Rem_{jt}>0
18. Begin **for loop** i<= 24 (Number of stations)
19. Begin **if** $wA_{jt}[i]$>0, then $b_{jt}[i]$ = $b_{jt}[i]\ +\ 1\ \forall i,j$
20. Rem_{jt}= $Rem_{jt} - 1\ \forall j$
21. Begin **if** Rem_{jt}<= 0, then **break**
22. End **if**
23. End **if**
24. End **while loop**
25. End **for loop**
26. Begin **if** IRP_{jt}> 0, then
27. Begin **for loop** for $i\ =\ 1\ to\ 24$
28. $IB_{ij(t+1)}$ = $IB_{ij(t+1)}$ + b_{ijt}
29. End **for loop**
30. $IRP_{j(t+1)}$ = $IRP_{jt} - \sum_i\ b_{ijt}$
31. End **if**
32. Begin **for loop** for $i\ =\ 0\ to\ 24$
33. $O_{ij(t+1)}$ = $O_{ijt}\ -\ b_{ijt}\ \forall i,j$

34. End **for loop**

35. SL = 1 - (Total Unfulfilled Demand/ Total Demand)

Algorithm (New Policy):

The algorithm for old policy can be modified suitably. The changes are mentioned below:

1. The demand has been added of different branches falling under same division.

2. Instead of 24 branches, algorithm runs for 3 divisions

3. Also, to calculate Initial Float Size, the Fleet Size read from **csv** file has been added of different branches falling under same division.