2018

SEMESTER 2

ENGSCI 355 Simulation Report

Connor McDowall
Josh Beckett
Alexander Zhao

October 15, 2018

Contents

Listings

List of Figures

1 Problem Description/Background

Kemito Pipfruit are a logistics company. Their operations are the transhipment of produce (avocados and apples) from suppliers to packhouses and packhouses to markets. Our company delivered a packing machine investment plan to minimise the acquisition and installation machine cost whilst able to meet historical demand. The plan was the output from a linear optimisation model. See 7.3 for the plan.

Kemito Pipfruit want to build a model to simulate the transhipment of both their avocado and apple supply chains. The purpose of the simulation is to investigate the role uncertainty plays in their operations and the effect on their machine investment plan. In particular, both the uncertainty in transhipment processes and supply chain interactions are of interest. Our conceptual model 7.6 set the plan on how to build the simulation model.

The companys transhipment operations have temporal, capacity and loading constraints. Trucks, with a capacity of 100 units of produce, arrive at the suppliers at 7am to begin loading fruit. After loading is complete, the fruit is transported to the relevant packhouses for unloading before packing can begin. After packing, the fruit is loaded into another truck, shipped to the relevant market for unloading. Kemito Pipfruit aim to have all fruit delivered to the relevant market by 5pm. Demand must be met at each market for each period and supply not exceeded (7.4). Loading bays at each destination (supplier, packhouse and market) have the capacity to load or unload one truck at a time.

Kemito Pipfruit wish to investigate the submitted packing machine investment plan. The company seeks an assessment on how suitable the plan is. The assessment is in terms of the plans cost and the ability to deliver fruit on time under supply chain uncertainties. The existing plan was built on the following considerations; transportation and machine costs, averaging processing rates and the historical demands per period.

2 Assumptions

The assumptions made in the conceptual model (7.6) informed our simulation model. After application, additional assumptions for our simulation model include:

- Machines only pack one unit of produce at a time.
- All servers process queues on a first in first out basis (FIFO) except the packhouse loading bays.
- The packhouses only have one loading/unloading bay. Trucks are sent to the load up at the packhouse if there is sufficient produce available. Loading is prioritised over unloading.
- Queues have an infinite capacity. Trucks can queue on the street and warehouses are large enough to store produce.
- A time series component controls the operating period, controlled by a time series threshold. The operating period is a 7 day week, 7am to 5pm each day. Operations pause overnight.
- A time series component controls when trucks reach the suppliers, controlled by a time series threshold. Trucks arrive at the suppliers at 7am, 10am and 1pm. Trucks are assumed to follow a schedule.
- We excluded rare events, such as congestion in Auckland's traffic causing transportation delays and fruit fly invasions decreasing supply.

3 Data

- The loading/unloading processes have a deterministic time of 30 seconds per truck.
- Both the intial investment and distribution plans are from the optimisation part of the project. See 7.3 and 7.5 respectively.
- A log normal distribution was assumed to model the variable packing times of the different machines. For large packing machines μ = 1.7963584 σ = 0.4397938 and medium packing machines μ = 2.079229 σ = 0.450827 (all units in minutes).
- The distribution plan (7.5) was converted into text files to assign what produce each truck would receive and transport between an origin and destination. Each truck only has a capacity for 100 units of produce.

- The transportation times were derived from the route's transports costs between source and destination. The travel cost were based off a normal distribution were the mean is $5 + \frac{\text{Travel Cost}}{3}$ and standard deviation is $\frac{TravelCost}{5}$. Travel costs can be found in 7.4.
- Historical supply and demand for produce used in the simulation are found in 7.4.

4 Approach and Model

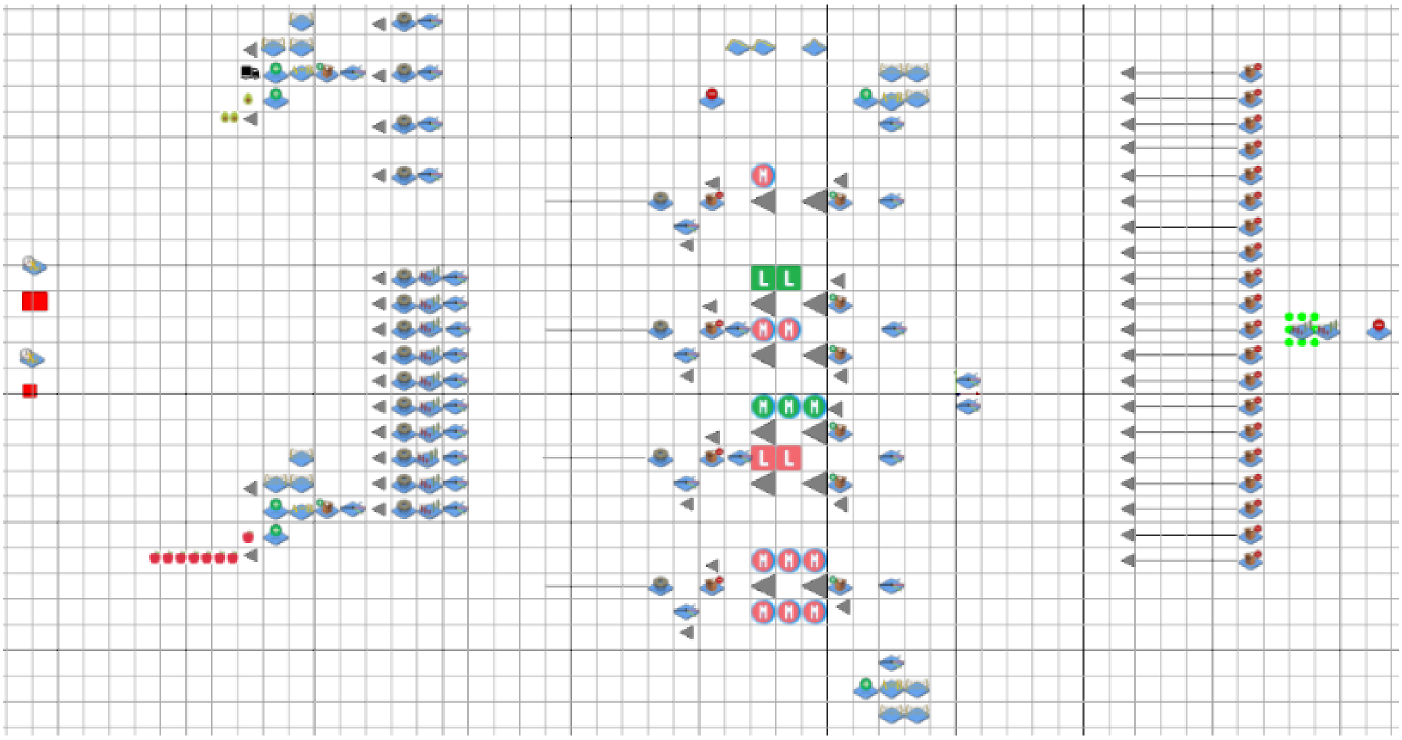


Figure 1: Model Overview

JaamSim (7.8) was used to model the transhipment operations of Kemito Pipfruit. Figure 1 shows an overview of the entire simulation model. Trucks (entity containers) are generated on the left, loaded with a produce type, flowing through the model. Produce is unloaded at a relevant packhouse, packed, loaded, then transported to the market. The trucks are unpacked and discharged through a sink on the far right. The apple and avocados are mutually exclusive and therefore can be handled seperately. The following approach is applied to both apples and avocados for each supplier, packhouse and market combination. The following approach is the implementation of our conceptual model.

4.1 Set Up of Entity Containers

Produce was modeled by entities and trucks were modelled by entity containers. Attributes are defined and set at zero for the truck. The attributes are: number of units transported (Num), destination (Des), source (Start), Travel Time (TravelTime) and Produce Type (Type).

Truck (entity container) attributes were assigned values using an assign component. Values were added to the model using the file to matrix components and prepared text files 7.9. There are three files to matrices per produce type: Travel time standard deviation. Travel time mean. The produce quantity transported on the truck between the source and destination. The assignment is shown in figure 9.

The add to component shown in figure 10 reads the capacity attribute for the container and packs the truck with a type of produce. There are two separate add to components, one for avocados and apples respectively. The add to component packs the produce instantaneously.

4.2 Load up of Supply

A branch sends the truck to the relevant server (supplier) based on its start attribute to simulate the loading for a deterministic loading time of 30 seconds, shown in figure 11.

4.3 Transhippment to Packhouse

After the truck leaves the designated supplier branch, it is sent along to an entity delay. This delay simulates the travel time between the supplier and packhouse. The travel time, calculated as mentioned in the data section, is an attribute assigned to the truck. The truck arrives at the loading bay server and is rerouted via a branch to be unpacked at a removefrom component. This component simulates the unloading at the packhouse. If the truck arrives while the server is engaged, the truck is rerouted to a queue. The server (loading bay) alternates between loading and unloading based on an assumed prioritisation.

4.4 Packing at Packhouse

After the unloading via the remove from component, the truck leaves the system via an entity sink. If more than one produce type is processed at a packhouse (apples or avocados), a branch is used to assign these to different queues for packing. These are the longest queues of our model.

When the packing machine is available, produce is rerouted into new server components (packing machines). The machine has an assigned service time based on a lognormal distribution for the relevant size, derived from the packing-times.csv data and data manipulation in R. The distributions are shown in 7.10.

4.5 Loading up Packed Produce

After packing, packed produce is stored in a relevant queue. A truck is sent to the packhouse for loading only if there is a sufficent quantity for packing in the queue. Loading trucks are prioritised over trucks unloading. The produce is loaded into the trucks instantaneously. The truck is then sent to a loading bay queue to simulate the deterministic loading bay time of 30 seconds.

4.6 Transhippment to Market

After loading, the entity container is sent to a sending branch (apples or avocados) through a branch at the relevant packhouse. From the sending branches, the trucks are transported to an entity delay to simulate the travel times derived using the aforementioned method in the data section.

4.7 Unloading at Market

After this entity delay, it goes to the markets associated unpack component, unpacked at a deterministic rate explained in the data section. If a truck is using the bay, the new truck is rerouted to a queue. After the truck is unloaded, it is sent to an entity sink.

4.8 Simulating Daily Periods and Truck Arrival at Suppliers

A time series component was implemented to simulate the 7am-5pm operating hours. We used Boolean values to switch between operating and non-operating states. A time series threshold controls the activation of this operating timeframe and subsequently all processes in the model. Figure 16 shows the time series for setting the operating period.

At 5pm, if a process is in progress (packing, transporting or unloading/loading), the current operation will continue. The remaining work will be completed and sent to the next stage before closure.

A second time series controls the intervals trucks are sent to suppliers, set at 7am, 10am and 1pm as shown in figure 17. This was to prevent too many trucks arriving at the supplier at once. A time series threshold controls the activation of these states, therefore the times the trucks arrive.

4.9 Collecting Statistics: Measures of Success and Failure

The average waiting time and queue length of produce before getting packed at the packhouse, total time of produce in the system and maximum number of produce in a queue in the system are recorded to measure success and failure. These statistics components use the produces state assignments to determine both the waiting times and quantity of produce in the system.

Finally, our model was set up to simulate a seven-day working week with 100 replications to simulate 100 weeks per historical demand. Operations pause overnight. No new processes start overnight (5pm to 7am) but are finished if already started. We repeated this methodology, simulating each historical demand per period. The simulation takes 13.67 seconds to solve 100 replications (per historical). The time to simulate all 10 historical periods is approximately 2 minutes and 17 seconds.

5 Experiments/Results

We conducted 100 replications of the simulation for each historical period using the investment plan recommended from the results of our optimisation model. We were interested in the total time, waiting time and queue length of produce in the system due to the recommended investment plan. Abnormalities lead to an inability to simulate historical periods five and six. The remaining historical periods were simulated. The following measures of success and failure are the averages over the 100 repetitions per period.

The aggregate average waiting times varied per historical period. Produce waited between an approximate 21.77 hours to 29.63 hours. The maximum waiting time recorded for one or more units is between 75.73 hours and 99.28 hours. These are both adequate as produce can spend a maximum of 168 hours in the system as waiting overnight while operations cease is included. See figure 2 for a graph of waiting times.

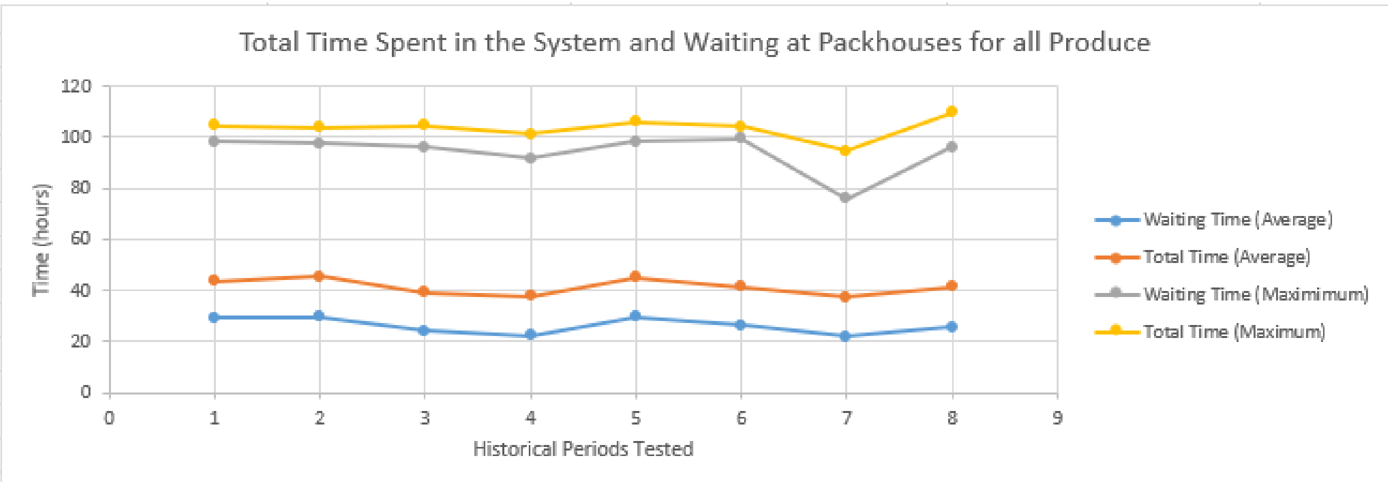


Figure 2: Average Waiting Time Results across 100 simulations per period

The produce spends an average total time in the system between 37.28 and 45.25 hours in the system. The average maximum recorded total time in the system is between 94.50 and 109.69 hours. Both are valid as the system has a limit of 168 hours, the number of hours in a seven-day week. See figure 2 the total waiting times per period.

The average queue length varied per packhouse due to the variation in number, size and type of packing machine at each packhouse. The average queue lengths for each packhouse (waiting to be packed) are: PH1(Apples) 46 to 47 units. PH2 (Apples) 45 to 239 units. PH2 (Avocados) 168 to 203 units. PH3 (Avocados) 199 to 333 units. PH3 (Apples) 112 to

279 units. PH4 (Apples) 57 to 74 units. These queue lengths are satisfactory as the largest equates to 4 trucks waiting for unloading. See figure 3 for a graph of average queue lengths. All market demands were met across the eight simulated periods.

We compared the quantity of produce passing through the sink to the demand in each period. There was small discrepancy however the largest is 0.2% off. This is attributed to the quantity through the sink never exceeding demand but could only be below. See figure 6 in the appendix for a comparison plot.

Based on all these parameters of success, our investment plan is plausible, therefore validated.

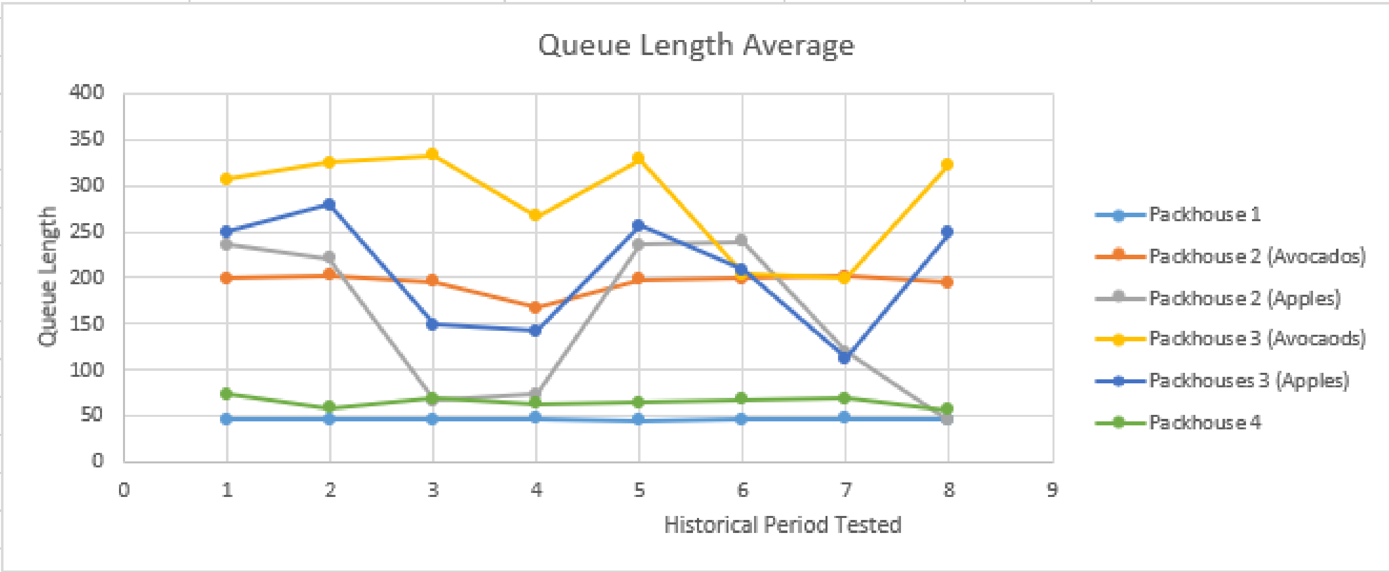


Figure 3: Average Queue Length Results across 100 simulations per period

See 7.1 for all output values of the simulation.

6 Conclusions/Recommendations

- Our investment plan satisfies the constraints of the simulation model.
- Only eight of the ten historical periods could be simulated. The average waiting times and total time in the system between periods were fairly consistent, therefore, assumed similar for both periods five and six.
- Our simulation was developed using the conceptual model but some measures of success and failure were not included. They are listed in 7.6.
- Market demand for all historical periods is met.
- Produce spent between 22 to 33 hours on average waiting in the packhouses to be packed.
- Produce spent on average 37 to 45 hours in the system (Supplier to Market).
- All produce was delivered to the markets at the end of the seven day working week within the operating hours.
- The average queue length at the packhouse waiting to be packed varied considerably based on the produce type and packhouse location but did not exceed 333 units.
- Under our current assumptions, the current investment plan is valid.
- Packhouse 2 and 3 are heavily used, resulting in large queues. Add more machines to packhouses 1 and 4 as currently under utilised will lighten the load.

7 Appendix

7.1 Experiments/Results

Period	Average Waiting Time (Hours)	Average Total Time (Hours)	Maximum Waiting Time (Hours)	Maximum Total Time (Hours)	Number Processed
1	29.13464215	43.64229899	98.23316829	104.394882	6411
2	29.6867968	45.25019213	97.45421522	103.6063611	6410
3	24.16961037	38.98218487	96.012376	104.4688572	5974
4	22.14331029	37.51389576	91.78405789	101.1063533	5768
7	29.62919504	44.94050513	98.38130198	106.0471609	6409
8	26.14494484	41.24003354	99.28304774	104.1099589	6204
9	21.77149303	37.28497828	75.72917459	94.49924969	5766
10	25.65801692	41.31754453	96.14145591	109.6872358	5975

Figure 4: Average and Total Waiting Times across 100 simulations per period

Period	PH1 Average Queue Length	PH2 (Avocado) Average Queue Length	PH2 (Apple) Average Queue Length	PH3 (Avocado) Average Queue Length	PH3 (Apple) Average Queue Length	PH4 (Apple) Average Queue Length
1	46.26736779	199.3869534	235.3101761	307.0158608	249.8359749	73.98242196
2	46.37193848	202.9477028	221.0122142	324.9126371	278.8992105	58.5489605
3	46.38827377	195.5429814	66.40162202	333.3377358	148.8140593	68.97516306
4	46.66620714	167.6707529	73.97862743	267.2202808	141.8858673	62.8319179
7	46.20310537	198.0039964	236.2669418	328.4323477	256.929124	64.48300262
8	46.41780645	199.256751	239.3153361	203.9847501	208.6950107	67.82580858
9	46.66988553	201.1734922	119.7450912	198.8419816	111.8546452	68.94364707
10	46.36670063	194.4986376	44.65499824	322.5050109	247.9100279	56.60421445

Figure 5: Average Queue Length Table across 100 simulations per period

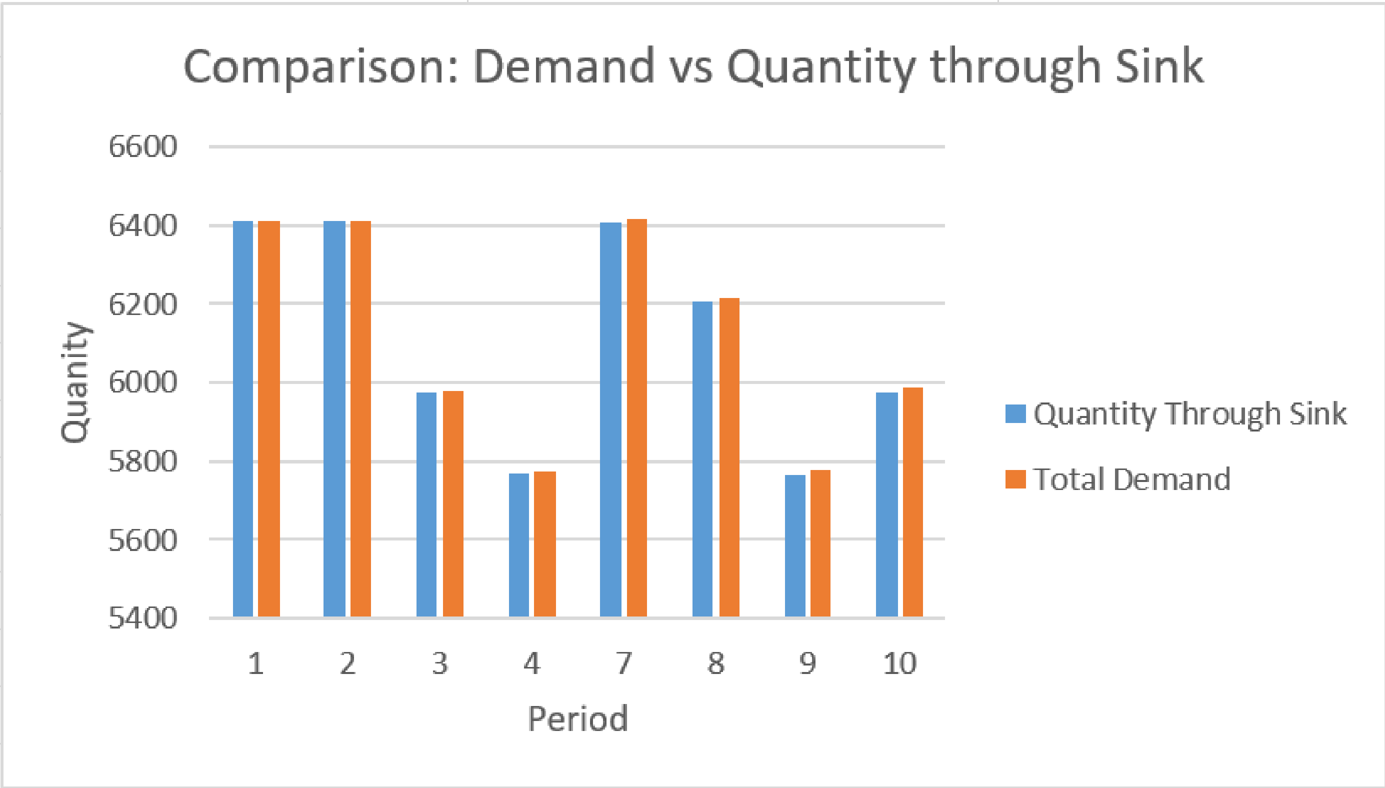


Figure 6: Average Delivery to Market accross 100 repetitions vs Historical Demand

Period	PH1 Maximum Queue Length	PH2 (Avocado) Maximum Queue Length	PH2 (Apple) Maximum Queue Length	PH3 (Avocado) Maximum Queue Length	PH3 (Apple) Maximum Queue Length	PH4 (Apple) Maximum Queue Length
1	260.36	476.49	554.88	766.1	581.02	496.61
2	260.96	480.79	543.69	781.42	648.07	394.78
3	260.52	471.4	282	790.46	411.83	496.16
4	261.49	449.72	300.52	745.44	407.97	410.84
7	260.15	475.19	557.21	782.55	593.2	514.89
8	260.99	472.96	567.47	668.78	516.98	498.11
9	261.68	474.08	398.35	664.37	364.12	501.81
10	260.67	471.79	251.24	791.83	569.66	465.8

Figure 7: Maximum Queue Length Results across 100 simulations per period

7.2 File Images

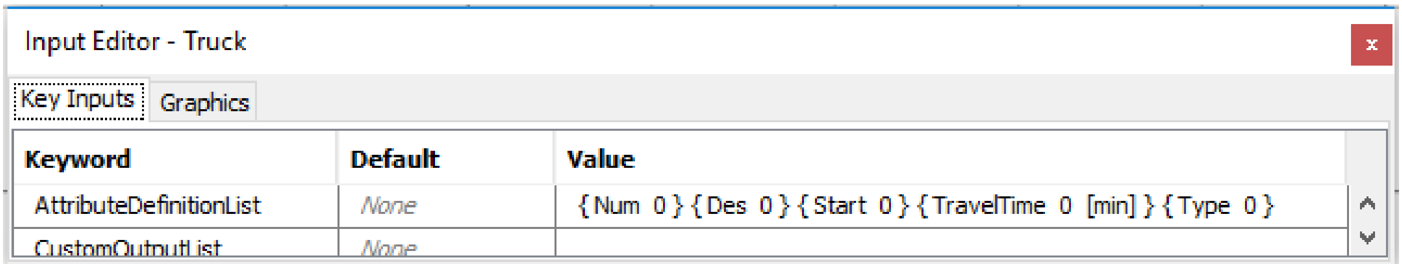


Figure 8: EntityContainer

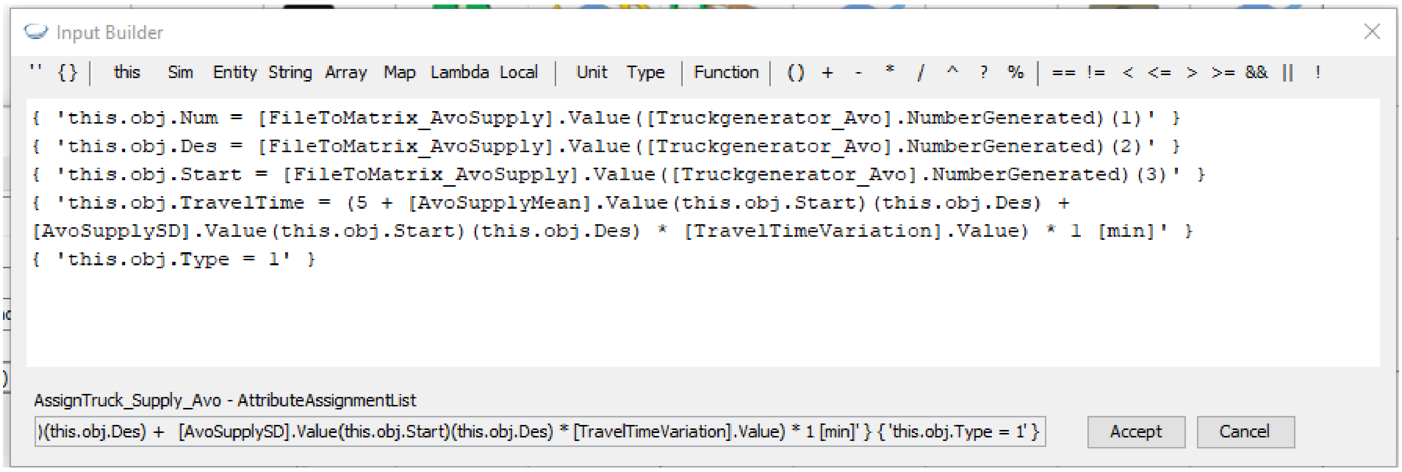


Figure 9: Assign Component to Assign Attributes to the Truck Entity Container

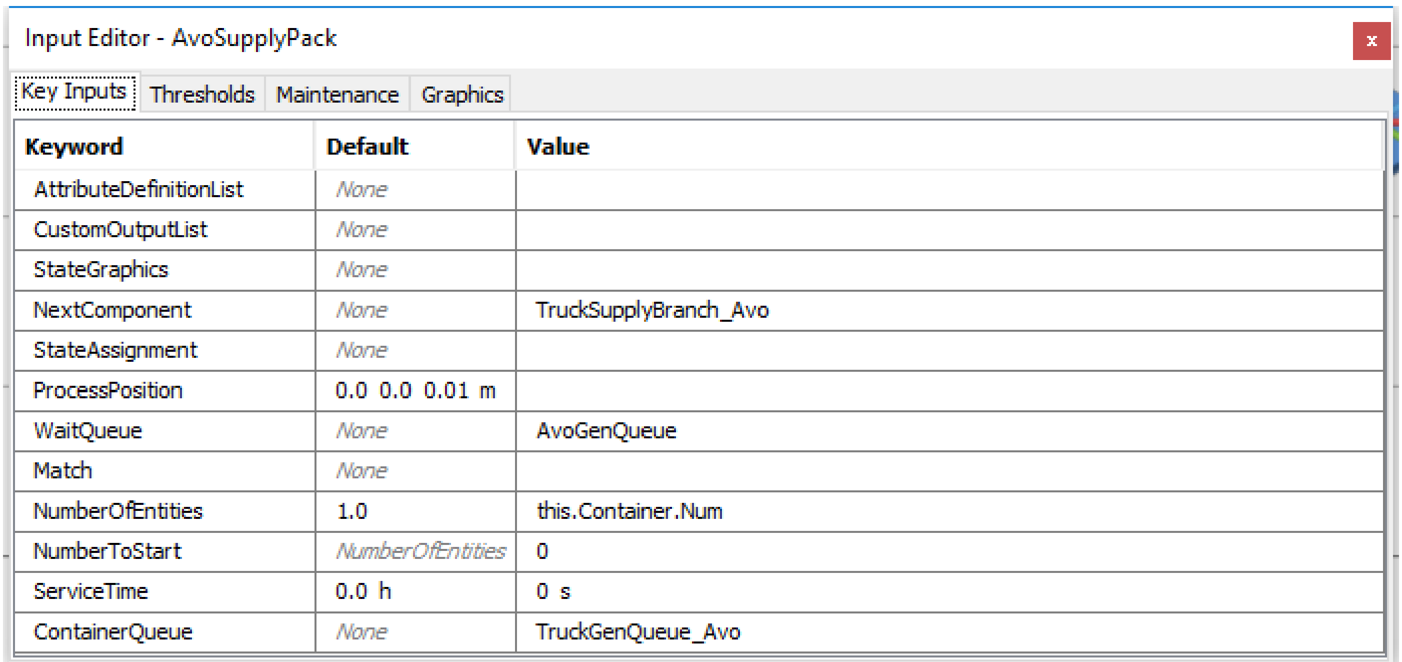
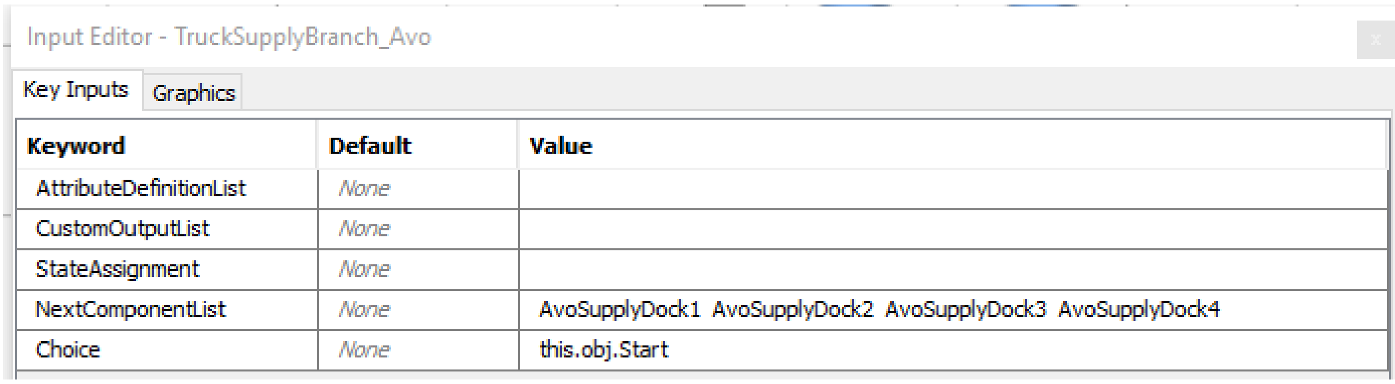


Figure 10: AvoSupplyPack Add To Component

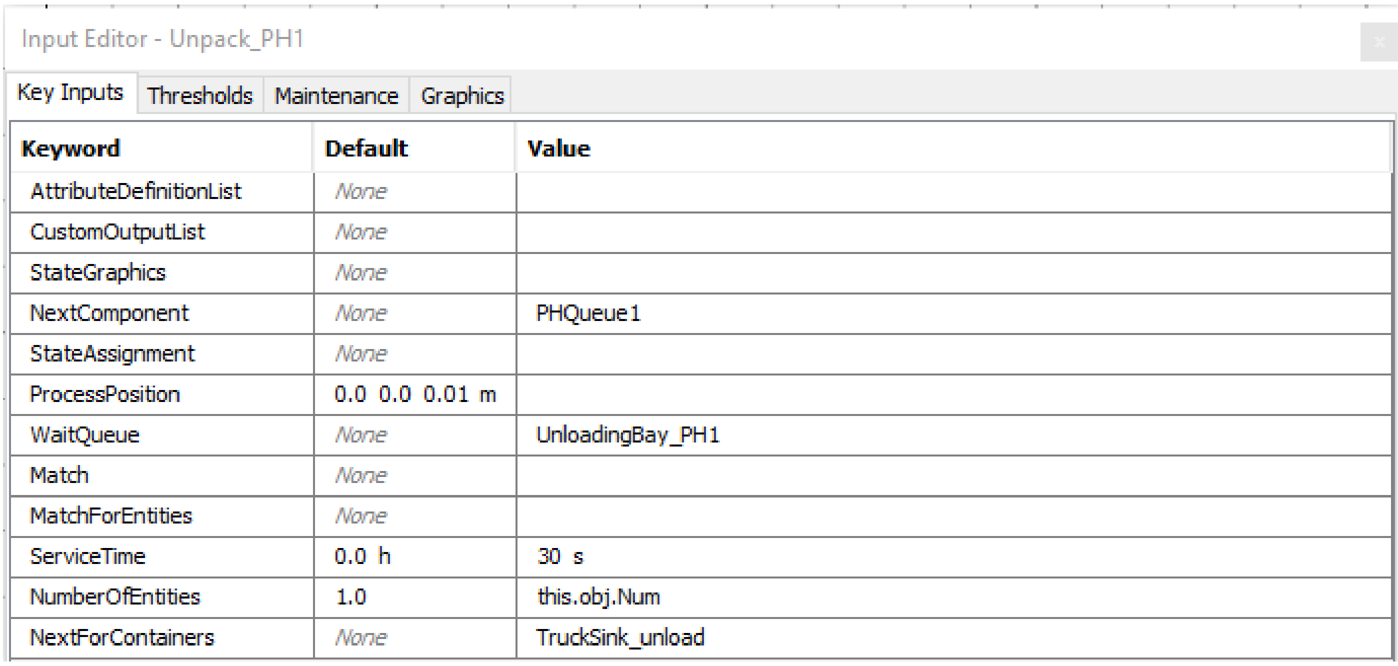


Input Editor - TruckSupplyBranch_Avo

Key Inputs | Graphics

Keyword	Default	Value
AttributeDefinitionList	None	
CustomOutputList	None	
StateAssignment	None	
NextComponentList	None	AvoSupplyDock1 AvoSupplyDock2 AvoSupplyDock3 AvoSupplyDock4
Choice	None	this.obj.Start

Figure 11: TruckSupplyBranchAvo Branch Component

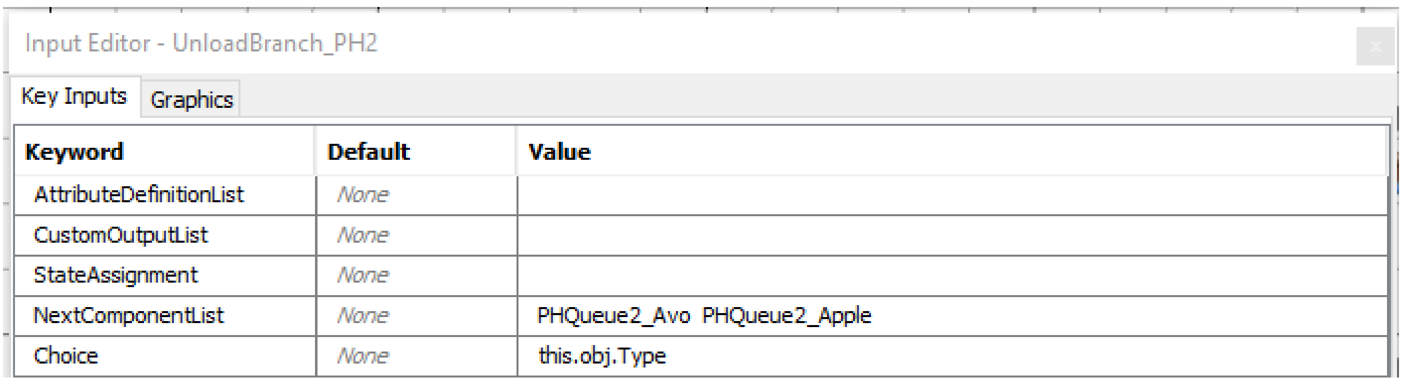


Input Editor - Unpack_PH1

Key Inputs | Thresholds | Maintenance | Graphics

Keyword	Default	Value
AttributeDefinitionList	None	
CustomOutputList	None	
StateGraphics	None	
NextComponent	None	PHQueue1
StateAssignment	None	
ProcessPosition	0.0 0.0 0.01 m	
WaitQueue	None	UnloadingBay_PH1
Match	None	
MatchForEntities	None	
ServiceTime	0.0 h	30 s
NumberOfEntities	1.0	this.obj.Num
NextForContainers	None	TruckSink_unload

Figure 12: Unpack PH1 Remove From Component

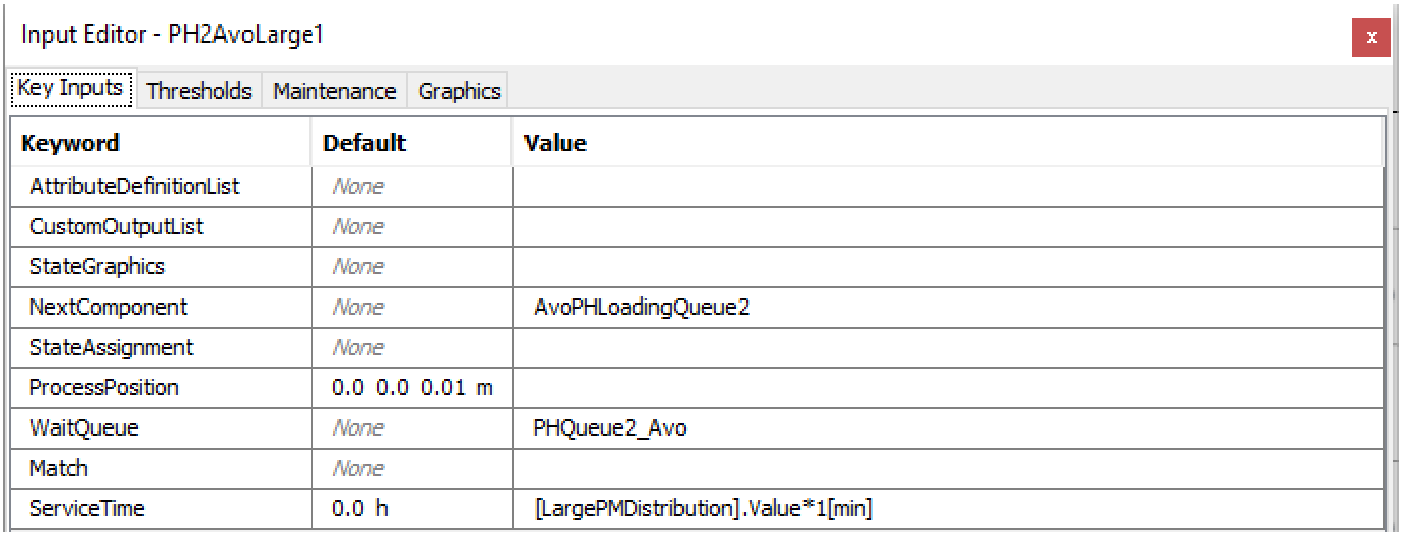


Input Editor - UnloadBranch_PH2

Key Inputs | Graphics

Keyword	Default	Value
AttributeDefinitionList	None	
CustomOutputList	None	
StateAssignment	None	
NextComponentList	None	PHQueue2_Avo PHQueue2_Apple
Choice	None	this.obj.Type

Figure 13: UnloadBranch PH2 Branch Component



Input Editor - PH2AvoLarge1

Key Inputs | Thresholds | Maintenance | Graphics

Keyword	Default	Value
AttributeDefinitionList	None	
CustomOutputList	None	
StateGraphics	None	
NextComponent	None	AvoPHLoadingQueue2
StateAssignment	None	
ProcessPosition	0.0 0.0 0.01 m	
WaitQueue	None	PHQueue2_Avo
Match	None	
ServiceTime	0.0 h	[LargePMDistribution].Value*1[min]

Figure 14: PH2AvoLarge1 Server Component for Packing

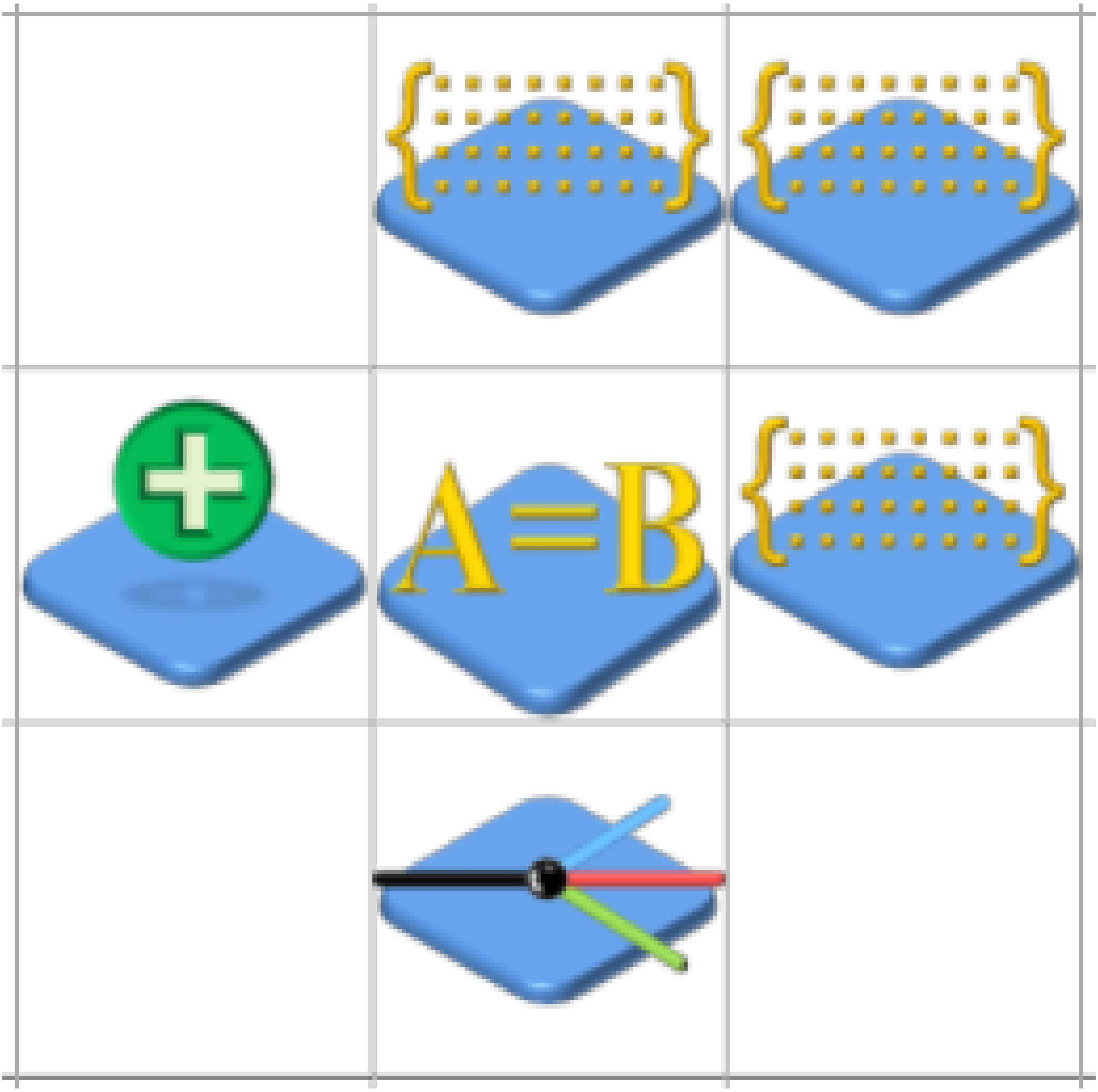


Figure 15: Layout for Truck Arrival

Input Editor - TimeSeries1

Key Inputs | Graphics

Keyword	Default	Value
AttributeDefinitionList	*None*	
CustomOutputList	*None*	
UnitType	*None*	DimensionlessUnit
Value	*None*	{ 0 h 0 } { 7 h 1 } { 17 h 0 }
CycleTime	Infinity h	24 h

Figure 16: TimeSeries1 for Operating Day Control

Input Editor - TruckSendingTimeseries

Key Inputs | Graphics

Keyword	Default	Value
AttributeDefinitionList	*None*	
CustomOutputList	*None*	
UnitType	*None*	DimensionlessUnit
Value	*None*	{ 0 h 0 } { 7 h 1 } { 7.0000001 h 0 } { 10 h 1 } { 10.000001 h 0 } { 13 h 1 } { 13.000000001 h 0 }
CycleTime	Infinity h	24 h

Figure 17: TruckSendingTimeseries

7.3 Optimisation Report

2018

Semester 2

Kemito Pipfruit: Optimisation

Connor McDowall
Josh Beckett
Alexander Zhao

September 10, 2018

Listings

Contents

List of Figures

List of Tables

1 Problem Description/Background

Kemito Pipfruit pack and distribute apples and avocados. They have a number of suppliers who provide produce, pack produce at Kemitos packhouses and ship the produce to various markets. Kemito wish to invest in new automated packing machines in their four packhouse locations. They wish to facilitate the transhipment of produce and meet the markets demand. Our objective is to decide the produce setting, size and number of packing machines to build at each packhouse. Our ancillary objective is to use optimisation to minimise the cost of produce transhipment from; supplier to packhouse and packhouse to market. The transhipment of produce and investment in machines for apples and avocados are mutually exclusive, therefore, can be treated as separate problems. In addition, demand varies at each market per period and is not known beforehand.

2 Data

The data given included: the fixed supply (units/period) for four avocado and ten apple producers, fixed per period. The historical, variable demand (units/period) for five avocado markets and fifteen apple markets for ten periods. The transportation costs per unit for apples and avocados from supplier to packhouse and packhouse to market. To conclude, the average packing rate (units/period) and cost (000/machine) of packing machine size (small, medium and large) completes the set of data.

The variable, historical demand for the twenty markets over the ten periods created uncertainty. The periods beginning, duration and correlation with other periods was unknown. These uncertainties created difficulties in formulating the model as there appeared to be no pattern per period or any indication of the likely cause.

We considered taking the peak value of each market demand across all periods but lead to a mass shortage of produce, unable to satisfy the demand of each market. Also, the cost of this solution would be exorbitant. Averaging the data across the periods was also considered. This resulted in the demand not being meet for several time periods while not considering fluctuating demand. We considered using a weighting system to penalise or omit unlikely periods, however, we did not have the industry expertise to deem what was an unlikely scenario. We agreed to use the data to build a robust solution by considering all periods.

3 Assumptions

We made the following assumptions to simplify our model formulation:

- Meeting market demand is a priority. This meant we solved our model to ensure that all the different market demands' for each period were met.
- Suppliers contracts must be honoured meaning we will not take more than what the producers can provide and we will not seek out contracts with others. The supply from each supplier is fixed for any period.
- No wastage at packhouses meaning produce flow is conserved. This may be unrealistic as human error, mechanical failure or transporation may create wastage.
- Minimising the cost of operation is our main driver. We are not concerned with the profitability of produce. We focus on the optimal locations for packing machines and the transportation of fruit between suppliers to packhouses and packhouses to markets.
- The location of packing machines is permanent. Machines cannot be decommissioned or transported to new locations. This ensures that our solution is very robust and can handle different levels of demand.

4 Model Formulation

Our model was formulated as a naturally integer linear programme, written in AMPL and solved using Gurobi. (Note: AMPL uses names for index notation rather than numbers).

4.1 Data

4.1.1 Sets and Parameters

Due the mutually exclusive nature of produce transhipment, two data files were defined from the data. A file for each fruit. Multiple sets were set in both files. These sets are the suppliers, periods, markets, pack machine sizes and packhouse locations. These sets function as objects to assign parameters to individual sets and/or a combination of sets. Arcs were created between suppliers to packhouses and packhouses to markets as an additional set. Each set was assigned relevant parameters. These parameters are the number of periods, the supply of each supplier. the demand of each market for each period, the pack rate for each pack machine size, the cost for each packing machine size, and the transportation costs between every supplier to every packhouse and every packhouse to every market. Arcs were also assigned lower and upper limits. These sets and parameters defined for the model can be found in the appendix (7.1).

4.2 Model

4.2.1 Variables

Flow and Built are the two decision variables. Flow is the number of units of produce shipped in the arc for a period. Built is the number of machines of each size built at the packhouse location. See the variables below (4.2.1).

- var $Flow_{ijp} \geq 0$, integer where i = origin in arc, j = destination in arc, p = period.
- var $Built_{mh} \geq 0$ where m = packmachine and h = packhouse.

4.2.2 Objective Function

Our objective function is to minimise the combined cost of installing the required number and size of packmachines at each packhouses, with transporting produce flow between arcs across all periods. See the function below (4.2.2).

$$\text{Min} \sum_i \sum_j \sum_p Cost_{ij} \times Flow_{ijp} + \sum_m \sum_h numPeriods \times packcost_m \times Built_{mh}$$

where i = origin, j = destination, p = period, m = packmachine, h = packhouse.

4.2.3 Constraints

Four constraints bind the model; Demand for all produce must be met at all markets. The total produce transported to packhouses must be less than or equal to supply. Aggregate flow into each packhouse must equal aggregate flow out of that packhouse, conserving the flows. Finally, the capacity of each packhouse's combined number of machines may not be exceeded by the flows in. The constraints are expressed mathematically below (4.2.3).

- Demand: $\sum_j Flow_{hjp} \geq demand_{jp}$
- Supply: $\sum_i Flow_{ihp} \leq supply_{ip}$
- Conserve: $\sum_i Flow_{ihp} = \sum_j Flow_{hjp}$
- Capacity: $\sum_m Built_{mh} \times rate_m \geq \sum_i Flow_{ihp}$

where i = supplier, j = market, h = packhouse, p = period and m = machine. See the whole AMPL Implementation of the model in 7.3 of the appendix.

5 Results

The machine investment plan explains the number and size of machines to install in each of your four packhouses. Table 1 outlines the proposed investment plan for both apples and avocados. The transhippment flows of apples and avocados varied due to the fluxuating demand at the markets across periods. The flows in each period were important to consider for machine installation but not the reported cost in the conclusions and recommendations 6. Future demand will likely be different. The transhippment flows for both produce in period one are displayed in the appendix 7.2.

Packhouse	Apple: Large	Apple: Medium	Avocado: Large	Avocado: Medium
One	-	1	-	-
Two	-	2	2	-
Three	2	-	-	3
Four	-	6	-	-

Table 1: Apple Machine Investment Plan

6 Conclusions

6.1 Recommendations

Based on the aforementioned results, Kemito Pipfruit should:

- Install one medium machine set to pack apples at Packhouse One.
- Install two medium and two large machines set to pack apples and avocados respectively at Packhouse Two.
- Install three medium and two large machines set to pack avocados and apples respectively at Packhouse Three.
- Install six medium machines set to pack apples at Packhouse Four.

The investment plan will cost **$440,000**. The model delivers a robust solution. Market demand is met in each period while minimising machine acquisition, installation and produce transhippment. See table 2 for the cost per unit for each machine size.

	Small	Medium	Large
Cost($)	10000	25000	35000

Table 2: Machine Size Cost

6.2 Improvements

We have delivered the best model based on the data you provided. With more data, we could formulate a model to provide a more robust solution. In particular:

- Using produce pricing to maximise the profit of your transhipment operations.
- Factoring in different product segments within apples and avocados.
- Factoring in produce wastage and conversion rates in transportation and packing.
- Use data to forecast period demand combined with potentially using futures contacts.
- Factoring in decommissioning and reinstalling packing machines in different packhouses.
- Using penalty costs for not meeting supply or demand, based on your existing contracts.

7.4 Optimisation Brief

Optimisation Project

ENGSCI 355, S2 2018

Problem Description

Kemito Pipfruit pack and distribute apples and avocadoes. They have a number of suppliers that provide them with produce that is then packed and shipped to a number of markets. Kemito is investing in new, automated packing machines at their 4 packhouses. Their two lines of produce, apples and avocadoes, are completely separate so they need a distribution and (packing machine) investment plan for each line. There are 4 suppliers and 5 markets for avocadoes and 10 suppliers and 15 markets for apples. In addition, although Kemito has guaranteed contracts with their suppliers, the demand in each market is not known beforehand. Kemito has 10 periods of historical data for the demand in each market for both avocadoes and apples.

Supply/Demand Data

The supply and demand data for apples and avocadoes is given in Tables 1 and 2. Note that avocado data is given first as it has lower volume and less suppliers/markets.

Table 1. Supply Demand data for Avocadoes

			Demand (Units/Period – Historical)									
Supplier	**Supply (Units/ Period)**	**Market**	1	2	3	4	5	6	7	8	9	10
S1	531	D1	6	1953	1976	262	1101	145	10	109	335	719
S2	285	D2	1609	12	58	131	407	1159	306	98	1240	224
S3	983	D3	326	77	8	524	67	160	1665	106	58	1077
S4	264	D4	85	9	7	765	64	180	5	1439	70	20
		D5	35	9	13	173	216	210	74	102	152	20

Table 2. Supply Demand data for Apples

			Demand (Units/Period – Historical)									
Supplier	**Supply (Units/ Period)**	**Market**	1	2	3	4	5	6	7	8	9	10
S1	69	D1	173	12	1138	1854	116	4	26	868	141	180
S2	10	D2	50	715	67	82	101	2	2	38	125	172
S3	841	D3	114	12	233	71	52	5	1754	10	100	74
S4	195	D4	17	32	884	120	32	5	3	10	431	93
S5	945	D5	78	17	221	66	32	2	4	10	278	57
S6	357	D6	209	12	524	66	72	3	2	49	1286	53
S7	364	D7	21	42	146	225	29	2	2	36	100	2266
S8	968	D8	1644	10	81	74	84	6	11	10	193	53
S9	594	D9	32	11	111	254	131	2	6	14	306	97
S10	14	D10	29	19	62	84	45	14	2	3178	104	89
		D11	47	10	74	71	2475	4218	15	14	193	53
		D12	195	351	121	467	32	2	4	11	100	55
		D13	1570	12	97	336	655	5	16	14	104	304
		D14	16	2846	60	77	30	2	14	52	100	80
		D15	155	249	93	66	29	76	2488	36	350	289

Packhouse Data

There are three different sized automated packing machines that Kemito are considering. Each packhouse can contain as many of each type of machine as necessary, but machines are pre-configured for apples or avocadoes, not both.

The data on the machines is given in Table 3.

Table 3. Data for Packing Machines

Size	**Average Packing Rate (Units/Period)**	**Cost ($1,000s)**
Small	100	10
Medium	375	25
Large	500	35

The transportation cost from the suppliers and markets to/from the packhouses are given in Tables 4 and 5 (for avocadoes and apples respectively).

Table 4. Transportation Cost to/from packhouses for Avocado suppliers/markets

Cost ($/unit) From/To	T1	T2	T3	T4
S1	21	84	42	93
S2	38	61	5	51
S3	67	9	74	89
S4	48	4	11	18
D1	77	73	16	64
D2	97	33	40	91
D3	60	66	14	90
D4	96	46	63	44
D5	44	97	52	70

Table 5. Transportation Cost to/from packhouses for Apple suppliers/markets

Cost ($/unit) From/To	T1	T2	T3	T4
S1	65	34	44	38
S2	3	35	79	35
S3	68	10	3	32
S4	80	90	80	2
S5	73	98	36	9
S6	80	56	47	48
S7	20	63	72	67
S8	87	47	72	20
S9	24	68	83	1
S10	32	20	96	36
D1	93	51	99	41
D2	66	92	71	46
D3	42	90	10	53
D4	19	57	64	29
D5	58	15	2	59
D6	24	87	83	1
D7	59	72	29	61
D8	97	99	48	29
D9	22	78	39	57
D10	84	20	68	19
D11	51	8	39	83
D12	2	14	99	38
D13	85	14	6	48
D14	7	93	1	71
D15	92	40	79	75

7.5 Distribution Plan

```
Gurobi 8.0.0: optimal solution; objective 7507463
3294 simplex iterations
184 branch-and-cut nodes
TotalCost = 7507460

Built [*,*,AP] (tr)
:     LARGE  MEDIUM  SMALL    :=
PH1     0       1       0
PH2     0       2       0
PH3     2       0       0
PH4     0       6       0

 [*,*,AV] (tr)
:     LARGE  MEDIUM  SMALL    :=
PH1     0       0       0
PH2     2       0       0
PH3     0       3       0
PH4     0       0       0
;

Flow [*,*,AP,1] (tr)
# $13 = AVS3
# $14 = AVS4
:     APS1  APS10  APS2  APS3  APS4  APS5  APS6  APS7  APS8  APS9  AVS1  AVS2  $13  $14  :=
PH1      0      0    10     0     0     0     0   364     0     0     0     0    0    0
PH2     69     14     0   191     0     0     0     0   476     0     0     0    0    0
PH3      0      0     0   650     0     0   350     0     0     0     0     0    0    0
PH4      0      0     0     0   195   945     0     0   492   594     0     0    0    0

:          PH1    PH2    PH3    PH4     :=
APD1         0      0      0    173
APD10        0      0      0     29
APD11        0     47      0      0
APD12      195      0      0      0
APD13        0    548    922    100
APD14       16      0      0      0
APD15        0    155      0      0
APD2         0      0      0     50
```

APD3	114	0	0	0
APD4	17	0	0	0
APD5	0	0	78	0
APD6	0	0	0	209
APD7	0	0	0	21
APD8	0	0	0	1644
APD9	32	0	0	0
AVD1	0	0	0	0
AVD2	0	0	0	0
AVD3	0	0	0	0
AVD4	0	0	0	0
AVD5	0	0	0	0

```
[*,*,AP,2] (tr)
# $13 = AVS3
# $14 = AVS4
```

:	APS1	APS10	APS2	APS3	APS4	APS5	APS6	APS7	APS8	APS9	AVS1	AVS2	$13	$14	:=
PH1	0	0	10	0	0	0	0	364	0	1	0	0	0	0	
PH2	69	14	0	191	0	0	0	0	451	0	0	0	0	0	
PH3	0	0	0	650	0	0	350	0	0	0	0	0	0	0	
PH4	0	0	0	0	195	945	0	0	517	593	0	0	0	0	

:	PH1	PH2	PH3	PH4	:=
APD1	0	12	0	0	
APD10	0	19	0	0	
APD11	0	10	0	0	
APD12	0	351	0	0	
APD13	0	12	0	0	
APD14	375	2	1000	1469	
APD15	0	249	0	0	
APD2	0	0	0	715	
APD3	0	0	0	12	
APD4	0	0	0	32	
APD5	0	17	0	0	
APD6	0	0	0	12	
APD7	0	42	0	0	
APD8	0	0	0	10	
APD9	0	11	0	0	

AVD1	0	0	0	0
AVD2	0	0	0	0
AVD3	0	0	0	0
AVD4	0	0	0	0
AVD5	0	0	0	0

[*,*,AP,3] (tr)
$14 = AVS4

:	APS1	APS10	APS2	APS3	APS4	APS5	APS6	APS7	APS8	APS9	AVS1	AVS2	AVS3	$14	:=
PH1	0	0	10	0	0	0	0	364	0	1	0	0	0	0	
PH2	69	14	0	0	0	0	0	0	336	0	0	0	0	0	
PH3	0	0	0	841	0	27	0	0	0	0	0	0	0	0	
PH4	0	0	0	0	195	918	0	0	544	593	0	0	0	0	

:	PH1	PH2	PH3	PH4	:=
APD1	0	69	0	1069	
APD10	0	62	0	0	
APD11	0	74	0	0	
APD12	0	121	0	0	
APD13	0	0	97	0	
APD14	0	0	60	0	
APD15	0	93	0	0	
APD2	0	0	0	67	
APD3	0	0	233	0	
APD4	375	0	0	509	
APD5	0	0	221	0	
APD6	0	0	0	524	
APD7	0	0	146	0	
APD8	0	0	0	81	
APD9	0	0	111	0	
AVD1	0	0	0	0	
AVD2	0	0	0	0	
AVD3	0	0	0	0	
AVD4	0	0	0	0	
AVD5	0	0	0	0	

[*,*,AP,4] (tr)
$14 = AVS4

:	APS1	APS10	APS2	APS3	APS4	APS5	APS6	APS7	APS8	APS9	AVS1	AVS2	AVS3	$14	:=
PH1	0	0	10	0	0	0	0	364	0	1	0	0	0	0	
PH2	69	14	0	0	0	0	0	0	364	0	0	0	0	0	
PH3	0	0	0	841	0	0	0	0	0	0	0	0	0	0	
PH4	0	0	0	0	195	945	0	0	517	593	0	0	0	0	

:	PH1	PH2	PH3	PH4	:=
APD1	0	0	0	1854	
APD10	0	30	0	54	
APD11	0	71	0	0	
APD12	187	280	0	0	
APD13	0	0	336	0	
APD14	0	0	77	0	
APD15	0	66	0	0	
APD2	0	0	0	82	
APD3	0	0	71	0	
APD4	0	0	0	120	
APD5	0	0	66	0	
APD6	0	0	0	66	
APD7	0	0	225	0	
APD8	0	0	0	74	
APD9	188	0	66	0	
AVD1	0	0	0	0	
AVD2	0	0	0	0	
AVD3	0	0	0	0	
AVD4	0	0	0	0	
AVD5	0	0	0	0	

```
 [*,*,AP,5] (tr)
# $13 = AVS3
# $14 = AVS4
```

:	APS1	APS10	APS2	APS3	APS4	APS5	APS6	APS7	APS8	APS9	AVS1	AVS2	$13	$14	:=
PH1	0	0	10	0	0	0	0	364	0	1	0	0	0	0	
PH2	69	14	0	0	0	0	0	0	667	0	0	0	0	0	
PH3	0	0	0	841	0	0	159	0	0	0	0	0	0	0	
PH4	0	0	0	0	195	945	0	0	57	593	0	0	0	0	

:	PH1	PH2	PH3	PH4	:=

APD1	0	0	0	116
APD10	0	0	0	45
APD11	182	750	968	575
APD12	32	0	0	0
APD13	0	0	0	655
APD14	30	0	0	0
APD15	0	0	0	29
APD2	0	0	0	101
APD3	0	0	0	52
APD4	0	0	0	32
APD5	0	0	32	0
APD6	0	0	0	72
APD7	0	0	0	29
APD8	0	0	0	84
APD9	131	0	0	0
AVD1	0	0	0	0
AVD2	0	0	0	0
AVD3	0	0	0	0
AVD4	0	0	0	0
AVD5	0	0	0	0

[*,*,AP,6] (tr)
$13 = AVS3
$14 = AVS4

:	APS1	APS10	APS2	APS3	APS4	APS5	APS6	APS7	APS8	APS9	AVS1	AVS2	$13	$14	:=
PH1	0	0	10	0	0	0	0	364	0	1	0	0	0	0	
PH2	69	14	0	189	0	0	0	0	478	0	0	0	0	0	
PH3	0	0	0	652	0	0	348	0	0	0	0	0	0	0	
PH4	0	0	0	0	195	945	0	0	490	593	0	0	0	0	

:	PH1	PH2	PH3	PH4	:=
APD1	0	0	0	4	
APD10	0	0	0	14	
APD11	369	750	998	2101	
APD12	2	0	0	0	
APD13	0	0	0	5	
APD14	2	0	0	0	
APD15	0	0	0	76	

```
APD2       0     0     0     2
APD3       0     0     0     5
APD4       0     0     0     5
APD5       0     0     2     0
APD6       0     0     0     3
APD7       0     0     0     2
APD8       0     0     0     6
APD9       2     0     0     0
AVD1       0     0     0     0
AVD2       0     0     0     0
AVD3       0     0     0     0
AVD4       0     0     0     0
AVD5       0     0     0     0

 [*,*,AP,7] (tr)
# $13 = AVS3
# $14 = AVS4
:    APS1 APS10 APS2  APS3  APS4  APS5  APS6  APS7  APS8  APS9 AVS1 AVS2 $13 $14 :=
PH1     0     0   10     0     0     0     0   364     0     0    0    0   0   0
PH2    69    14    0   190     0     0     0     0   477     0    0    0   0   0
PH3     0     0    0   651     0     0   349     0     0     0    0    0   0   0
PH4     0     0    0     0   195   945     0     0   491   594    0    0   0   0

:        PH1   PH2   PH3   PH4     :=
APD1       0     0     0    26
APD10      0     0     0     2
APD11      0    15     0     0
APD12      4     0     0     0
APD13      0     0     0    16
APD14     14     0     0     0
APD15      0   735     0  1753
APD2       0     0     0     2
APD3     350     0   996   408
APD4       0     0     0     3
APD5       0     0     4     0
APD6       0     0     0     2
APD7       0     0     0     2
APD8       0     0     0    11
```

APD9	6	0	0	0
AVD1	0	0	0	0
AVD2	0	0	0	0
AVD3	0	0	0	0
AVD4	0	0	0	0
AVD5	0	0	0	0

[∗,∗,AP,8] (tr)
$13 = AVS3
$14 = AVS4

:	APS1	APS10	APS2	APS3	APS4	APS5	APS6	APS7	APS8	APS9	AVS1	AVS2	$13	$14	:=
PH1	0	0	10	0	0	0	0	364	0	0	0	0	0	0	
PH2	69	14	0	215	0	0	0	0	452	0	0	0	0	0	
PH3	0	0	0	626	0	0	350	0	0	0	0	0	0	0	
PH4	0	0	0	0	195	945	0	0	516	594	0	0	0	0	

:	PH1	PH2	PH3	PH4	:=
APD1	252	0	0	616	
APD10	0	750	794	1634	
APD11	0	0	14	0	
APD12	11	0	0	0	
APD13	0	0	14	0	
APD14	0	0	52	0	
APD15	0	0	36	0	
APD2	38	0	0	0	
APD3	0	0	10	0	
APD4	10	0	0	0	
APD5	0	0	10	0	
APD6	49	0	0	0	
APD7	0	0	36	0	
APD8	0	0	10	0	
APD9	14	0	0	0	
AVD1	0	0	0	0	
AVD2	0	0	0	0	
AVD3	0	0	0	0	
AVD4	0	0	0	0	
AVD5	0	0	0	0	

```
 [*,*,AP,9] (tr)
# $14 = AVS4
:    APS1  APS10  APS2   APS3   APS4   APS5  APS6   APS7   APS8   APS9  AVS1  AVS2  AVS3  $14  :=
PH1     0      0    10      0      0      0     0    364      0      1     0     0     0    0
PH2    69     14     0     98      0      0     0      0    362      0     0     0     0    0
PH3     0      0     0    743      0      0     0      0      0      0     0     0     0    0
PH4     0      0     0      0    195    945     0      0    517    593     0     0     0    0

:         PH1   PH2   PH3   PH4     :=
APD1        0     0     0    141
APD10       0     0     0    104
APD11       0   193     0      0
APD12     100     0     0      0
APD13       0     0   104      0
APD14       0     0   100      0
APD15       0   350     0      0
APD2        0     0     0    125
APD3        0     0   100      0
APD4       30     0     0    401
APD5        0     0   278      0
APD6        0     0     0   1286
APD7        0     0   100      0
APD8        0     0     0    193
APD9      245     0    61      0
AVD1        0     0     0      0
AVD2        0     0     0      0
AVD3        0     0     0      0
AVD4        0     0     0      0
AVD5        0     0     0      0

 [*,*,AP,10] (tr)
# $2 = APS10
# $7 = APS6
# $11 = AVS1
# $12 = AVS2
# $13 = AVS3
# $14 = AVS4
:    APS1   $2  APS2   APS3   APS4   APS5   $7   APS7   APS8   APS9  $11  $12  $13  $14   PH1  :=
```

APD1	0
APD10	0
APD11	0
APD12	55
APD13	0
APD14	80
APD15	0
APD2	0
APD3	0
APD4	93
APD5	0
APD6	0
APD7	49
APD8	0
APD9	97
AVD1	0
AVD2	0
AVD3	0
AVD4	0

AVD5

	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0
PH1	0	0	10	0	0	0	0	364	0	0	0	0	0	0	.
PH2	69	14	0	0	0	0	0	0	259	0	0	0	0	0	.
PH3	0	0	0	841	0	159	0	0	0	0	0	0	0	0	.
PH4	0	0	0	0	195	786	0	0	624	594	0	0	0	0	.

:	PH2	PH3	PH4	:=
APD1	0	0	180	
APD10	0	0	89	
APD11	53	0	0	
APD12	0	0	0	
APD13	0	304	0	
APD14	0	0	0	
APD15	289	0	0	
APD2	0	0	172	
APD3	0	74	0	
APD4	0	0	0	
APD5	0	57	0	
APD6	0	0	53	
APD7	0	565	1652	
APD8	0	0	53	
APD9	0	0	0	
AVD1	0	0	0	
AVD2	0	0	0	
AVD3	0	0	0	
AVD4	0	0	0	
AVD5	0	0	0	

[*,*,AV,1] (tr)
$1 = APS1
$2 = APS10
$3 = APS2
$4 = APS3
$5 = APS4
$6 = APS5
$7 = APS6
$8 = APS7

```
#  $9  =  APS8
#  $10  =  APS9
:           $1    $2    $3    $4    $5    $6    $7    $8    $9   $10    AVS1    AVS2    AVS3    AVS4   PH1      PH2  :=
APD1         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD10        .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD11        .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD12        .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD13        .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD14        .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD15        .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD2         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD3         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD4         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD5         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD6         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD7         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD8         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
APD9         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
AVD1         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
AVD2         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0      915
AVD3         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
AVD4         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0       85
AVD5         .     .     .     .     .     .     .     .     .     .       .       .       .       .     0        0
PH1          0     0     0     0     0     0     0     0     0     0       0       0       0       0     .        .
PH2          0     0     0     0     0     0     0     0     0     0       0       0     983      17     .        .
PH3          0     0     0     0     0     0     0     0     0     0     529     285       0     247     .        .
PH4          0     0     0     0     0     0     0     0     0     0       0       0       0       0     .        .

:             PH3  PH4       :=
APD1            0    0
APD10           0    0
APD11           0    0
APD12           0    0
APD13           0    0
APD14           0    0
APD15           0    0
APD2            0    0
APD3            0    0
```

```
APD4       0    0
APD5       0    0
APD6       0    0
APD7       0    0
APD8       0    0
APD9       0    0
AVD1       6    0
AVD2     694    0
AVD3     326    0
AVD4       0    0
AVD5      35    0

 [*,*,AV,2] (tr)
# $1 = APS1
# $2 = APS10
# $3 = APS2
# $4 = APS3
# $5 = APS4
# $6 = APS5
# $7 = APS6
# $8 = APS7
# $9 = APS8
# $10 = APS9
:       $1  $2  $3  $4  $5  $6  $7  $8  $9  $10  AVS1  AVS2  AVS3  AVS4  PH1  PH2 :=
APD1     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD10    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD11    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD12    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD13    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD14    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD15    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD2     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD3     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD4     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD5     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD6     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD7     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD8     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
```

APD9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	873
AVD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	12
AVD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	77
AVD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	9
AVD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	9
PH1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.
PH2	0	0	0	0	0	0	0	0	0	0	0	0	980	0	.	.
PH3	0	0	0	0	0	0	0	0	0	0	531	285	0	264	.	.
PH4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.

:	PH3	PH4	:=
APD1	0	0	
APD10	0	0	
APD11	0	0	
APD12	0	0	
APD13	0	0	
APD14	0	0	
APD15	0	0	
APD2	0	0	
APD3	0	0	
APD4	0	0	
APD5	0	0	
APD6	0	0	
APD7	0	0	
APD8	0	0	
APD9	0	0	
AVD1	1080	0	
AVD2	0	0	
AVD3	0	0	
AVD4	0	0	
AVD5	0	0	

```
 [*,*,AV,3] (tr)
# $1 = APS1
# $2 = APS10
# $3 = APS2
# $4 = APS3
```

```
#  $5  =  APS4
#  $6  =  APS5
#  $7  =  APS6
#  $8  =  APS7
#  $9  =  APS8
#  $10  =  APS9
:          $1    $2    $3    $4    $5    $6    $7    $8    $9   $10    AVS1   AVS2   AVS3   AVS4  PH1    PH2  :=
APD1        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD10       .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD11       .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD12       .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD13       .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD14       .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD15       .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD2        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD3        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD4        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD5        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD6        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD7        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD8        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
APD9        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      0
AVD1        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0    896
AVD2        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0     58
AVD3        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      8
AVD4        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0      7
AVD5        .     .     .     .     .     .     .     .     .     .      .      .      .      .     0     13
PH1         0     0     0     0     0     0     0     0     0     0      0      0      0      0     .      .
PH2         0     0     0     0     0     0     0     0     0     0      0      0    982      0     .      .
PH3         0     0     0     0     0     0     0     0     0     0    531    285      0    264     .      .
PH4         0     0     0     0     0     0     0     0     0     0      0      0      0      0     .      .

:            PH3   PH4      :=
APD1           0     0
APD10          0     0
APD11          0     0
APD12          0     0
APD13          0     0
```

```
APD14          0     0
APD15          0     0
APD2           0     0
APD3           0     0
APD4           0     0
APD5           0     0
APD6           0     0
APD7           0     0
APD8           0     0
APD9           0     0
AVD1        1080     0
AVD2           0     0
AVD3           0     0
AVD4           0     0
AVD5           0     0

 [*,*,AV,4] (tr)
# $1 = APS1
# $2 = APS10
# $3 = APS2
# $4 = APS3
# $5 = APS4
# $6 = APS5
# $7 = APS6
# $8 = APS7
# $9 = APS8
# $10 = APS9
:       $1  $2  $3  $4  $5  $6  $7  $8  $9  $10  AVS1  AVS2  AVS3  AVS4  PH1   PH2 :=
APD1     .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD10    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD11    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD12    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD13    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD14    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD15    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD2     .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD3     .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
APD4     .   .   .   .   .   .   .   .   .   .    .     .     .     .     0     0
```

APD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	131
AVD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	765
AVD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
PH1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.
PH2	0	0	0	0	0	0	0	0	0	0	0	0	896	0	.	.
PH3	0	0	0	0	0	0	0	0	0	0	410	285	0	264	.	.
PH4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.

:	PH3	PH4	:=
APD1	0	0	
APD10	0	0	
APD11	0	0	
APD12	0	0	
APD13	0	0	
APD14	0	0	
APD15	0	0	
APD2	0	0	
APD3	0	0	
APD4	0	0	
APD5	0	0	
APD6	0	0	
APD7	0	0	
APD8	0	0	
APD9	0	0	
AVD1	262	0	
AVD2	0	0	
AVD3	524	0	
AVD4	0	0	
AVD5	173	0	

[*,*,AV,5] (tr)

```
#  $1  =  APS1
#  $2  =  APS10
#  $3  =  APS2
#  $4  =  APS3
#  $5  =  APS4
#  $6  =  APS5
#  $7  =  APS6
#  $8  =  APS7
#  $9  =  APS8
#  $10  =  APS9
```

:	$1	$2	$3	$4	$5	$6	$7	$8	$9	$10	AVS1	AVS2	AVS3	AVS4	PH1	PH2	:=
APD1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD10	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD11	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD12	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD13	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD14	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD15	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
APD9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0	
AVD1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	21	
AVD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	407	
AVD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	67	
AVD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	64	
AVD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	216	
PH1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	
PH2	0	0	0	0	0	0	0	0	0	0	0	0	775	0	.	.	
PH3	0	0	0	0	0	0	0	0	0	0	531	285	0	264	.	.	
PH4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.	

:	PH3	PH4	:=
APD1	0	0	

```
APD10       0    0
APD11       0    0
APD12       0    0
APD13       0    0
APD14       0    0
APD15       0    0
APD2        0    0
APD3        0    0
APD4        0    0
APD5        0    0
APD6        0    0
APD7        0    0
APD8        0    0
APD9        0    0
AVD1     1080    0
AVD2        0    0
AVD3        0    0
AVD4        0    0
AVD5        0    0
```



```
 [*,*,AV,6] (tr)
# $1 = APS1
# $2 = APS10
# $3 = APS2
# $4 = APS3
# $5 = APS4
# $6 = APS5
# $7 = APS6
# $8 = APS7
# $9 = APS8
# $10 = APS9
:       $1  $2  $3  $4  $5  $6  $7  $8  $9  $10  AVS1  AVS2  AVS3  AVS4  PH1  PH2 :=
APD1     .   .   .   .   .   .   .   .   .   .    .     .     .     .     0    0
APD10    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0    0
APD11    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0    0
APD12    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0    0
APD13    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0    0
APD14    .   .   .   .   .   .   .   .   .   .    .     .     .     .     0    0
```

APD15	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	820
AVD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	180
AVD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
PH1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.
PH2	0	0	0	0	0	0	0	0	0	0	0	0	983	17	.	.
PH3	0	0	0	0	0	0	0	0	0	0	322	285	0	247	.	.
PH4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.

:	PH3	PH4	:=
APD1	0	0	
APD10	0	0	
APD11	0	0	
APD12	0	0	
APD13	0	0	
APD14	0	0	
APD15	0	0	
APD2	0	0	
APD3	0	0	
APD4	0	0	
APD5	0	0	
APD6	0	0	
APD7	0	0	
APD8	0	0	
APD9	0	0	
AVD1	145	0	
AVD2	339	0	
AVD3	160	0	

```
AVD4       0    0
AVD5     210    0

 [*,*,AV,7] (tr)
# $1 = APS1
# $2 = APS10
# $3 = APS2
# $4 = APS3
# $5 = APS4
# $6 = APS5
# $7 = APS6
# $8 = APS7
# $9 = APS8
# $10 = APS9
:       $1  $2  $3  $4  $5  $6  $7  $8  $9  $10  AVS1  AVS2  AVS3  AVS4  PH1  PH2 :=
APD1     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD10    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD11    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD12    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD13    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD14    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD15    .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD2     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD3     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD4     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD5     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD6     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD7     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD8     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
APD9     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
AVD1     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    0
AVD2     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0  306
AVD3     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0  595
AVD4     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0    5
AVD5     .   .   .   .   .   .   .   .   .   .     .     .     .     .    0   74
PH1      0   0   0   0   0   0   0   0   0   0     0     0     0     0    .    .
PH2      0   0   0   0   0   0   0   0   0   0     0     0   980     0    .    .
PH3      0   0   0   0   0   0   0   0   0   0   531   285     0   264    .    .
```

```
PH4        0    0    0    0    0    0    0    0    0    0      0      0      0      0     .     .

:          PH3  PH4      :=
APD1         0    0
APD10        0    0
APD11        0    0
APD12        0    0
APD13        0    0
APD14        0    0
APD15        0    0
APD2         0    0
APD3         0    0
APD4         0    0
APD5         0    0
APD6         0    0
APD7         0    0
APD8         0    0
APD9         0    0
AVD1        10    0
AVD2         0    0
AVD3      1070    0
AVD4         0    0
AVD5         0    0

 [*,*,AV,8] (tr)
# $1 = APS1
# $2 = APS10
# $3 = APS2
# $4 = APS3
# $5 = APS4
# $6 = APS5
# $7 = APS6
# $8 = APS7
# $9 = APS8
# $10 = APS9
:       $1   $2   $3   $4   $5   $6   $7   $8   $9  $10   AVS1   AVS2   AVS3   AVS4  PH1    PH2  :=
APD1     .    .    .    .    .    .    .    .    .    .      .      .      .      .    0      0
APD10
```

	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD11																
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD12																
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD13																
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD14																
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD15																
	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD7	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	1000
AVD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
PH1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.
PH2	0	0	0	0	0	0	0	0	0	0	0	0	983	17	.	.
PH3	0	0	0	0	0	0	0	0	0	0	322	285	0	247	.	.
PH4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.

:	PH3	PH4	:=
APD1	0	0	
APD10	0	0	
APD11	0	0	
APD12	0	0	
APD13	0	0	
APD14	0	0	
APD15	0	0	
APD2	0	0	

```
APD3      0    0
APD4      0    0
APD5      0    0
APD6      0    0
APD7      0    0
APD8      0    0
APD9      0    0
AVD1    109    0
AVD2     98    0
AVD3    106    0
AVD4    439    0
AVD5    102    0

 [*,*,AV,9] (tr)
# $1 = APS1
# $2 = APS10
# $3 = APS2
# $4 = APS3
# $5 = APS4
# $6 = APS5
# $7 = APS6
# $8 = APS7
# $9 = APS8
# $10 = APS9
:      $1  $2  $3  $4  $5  $6  $7  $8  $9  $10  AVS1  AVS2  AVS3  AVS4  PH1  PH2 :=
APD1    .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD10   .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD11   .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD12   .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD13   .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD14   .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD15   .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD2    .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD3    .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD4    .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD5    .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD6    .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
APD7    .   .   .   .   .   .   .   .   .   .    .     .     .     .    0    0
```

APD8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
APD9	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	930
AVD3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
AVD4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	70
AVD5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	0	0
PH1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.
PH2	0	0	0	0	0	0	0	0	0	0	0	0	983	17	.	.
PH3	0	0	0	0	0	0	0	0	0	0	323	285	0	247	.	.
PH4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.	.

:	PH3	PH4	:=
APD1	0	0	
APD10	0	0	
APD11	0	0	
APD12	0	0	
APD13	0	0	
APD14	0	0	
APD15	0	0	
APD2	0	0	
APD3	0	0	
APD4	0	0	
APD5	0	0	
APD6	0	0	
APD7	0	0	
APD8	0	0	
APD9	0	0	
AVD1	335	0	
AVD2	310	0	
AVD3	58	0	
AVD4	0	0	
AVD5	152	0	

```
 [*,*,AV,10] (tr)
# $1 = APS1
# $2 = APS10
# $3 = APS2
```

```
#  $4  =  APS3
#  $5  =  APS4
#  $6  =  APS5
#  $7  =  APS6
#  $8  =  APS7
#  $9  =  APS8
#  $10  =  APS9
:          $1    $2    $3    $4    $5    $6    $7    $8    $9   $10    AVS1    AVS2    AVS3    AVS4   PH1      PH2  :=
APD1        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD10       .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD11       .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD12       .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD13       .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD14       .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD15       .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD2        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD3        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD4        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD5        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD6        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD7        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD8        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
APD9        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
AVD1        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0        0
AVD2        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0      224
AVD3        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0      716
AVD4        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0       20
AVD5        .     .     .     .     .     .     .     .     .     .      .       .       .       .      0       20
PH1         0     0     0     0     0     0     0     0     0     0      0       0       0       0      .        .
PH2         0     0     0     0     0     0     0     0     0     0      0       0     980       0      .        .
PH3         0     0     0     0     0     0     0     0     0     0    531     285       0     264      .        .
PH4         0     0     0     0     0     0     0     0     0     0      0       0       0       0      .        .

:            PH3  PH4        :=
APD1           0    0
APD10          0    0
APD11          0    0
APD12          0    0
```

APD13	0	0
APD14	0	0
APD15	0	0
APD2	0	0
APD3	0	0
APD4	0	0
APD5	0	0
APD6	0	0
APD7	0	0
APD8	0	0
APD9	0	0
AVD1	719	0
AVD2	0	0
AVD3	361	0
AVD4	0	0
AVD5	0	0

;

7.6 Conceptual Model

1 Conceptual Design Report

1.1 Background – Problem Description

Kemito Pipfruit are a logistics company. Their operations are the transhipment of produce (avocados and apples) from suppliers to packhouses and packhouses to markets. Our company delivered a packing machine investment plan to minimise the acquisition and installation machine cost whilst able to meet historical demand.

Kemito Pipfruit want to build a model to simulate the transhipment of both their avocado and apple supply chains. The purpose of the simulation is to investigate the role uncertainty plays in their operations and the effect on their machine investment plan. In particular, both the uncertainty in transhipment processes and supply chain interactions are of interest.

The company's transhipment operations have temporal, capacity and loading constraints. Trucks, with a capacity of 100 units of produce, arrive at the suppliers at 7am to begin loading fruit. After loading is complete, the fruit is transported to the relevant packhouses for unloading before packing can begin. After packing, the fruit is loaded into another truck, shipped to the relevant market for unloading. Kemito Pipfruit aim to have all fruit delivered to the relevant market by 5pm. Loading bays at each destination (supplier, packhouse and market) have the capacity to load or unload one truck at a time.

Kemito Pipfruit wish to investigate the submitted packing machine investment plan. The company seeks an assessment on how suitable the plan is. The assessment is in terms of the plan's cost and the ability to deliver fruit on time under supply chain uncertainties. The existing plan was built on the following considerations; transportation and machine costs, averaging processing rates and the historical demands per period.

1.2 Objectives of the study;

The Objective of the study is to validate the packing machine investment plan. Kemito Pipfruit are interested in the quantity and size of packing machines at each location. The setup is to ensure all produce travels from the suppliers to the markets via the packhouses for the week, to meet 100% of demand 95% of the time. The setup is to ensure **95%** of trucks wait no more than **10 minutes** in the supplier, market and packhouse loading bays for loading/unloading, **95% of produce** wait no more than **30 minutes** to be packed, and **95% of produce** waits no more than **30 minutes** to be loaded. Due to loading bay constraints, only one truck may be loaded/unloaded each time. Each truck can transport up to 100 units at a time. Ideally, no produce is to be unloaded at a market past 5pm or loaded at a supplier before 7am and after 5pm. The number and size of packing machines at each location are fixed to our investment plan first but are not constrained, therefore will change in subsequent iterations. Produce is shipped daily. It is required to meet weekly demand.

1.3 Expected benefits;

The expected benefits are a virtual environment for evaluating the subsequent factors:

- Supplier, packhouse and market truck loading/unloading times.
- Produce packing and distribution waiting times (avocados/apples).
- The total time trucks spend transporting produce from supplier to packhouse (loading at supplier, transportation time, unloading at packhouse, loading bay waiting times).
- The total time trucks spend transporting produce from packhouse to market (loading at packhouse, transportation time, unloading at market, loading bay waiting times).
- Total time produce spends at the packhouse(s).
- The total time produce (avocados and apples) are in the system (supplier to packhouse to market).
- The aggregate produce reaching the market.
- The aggregate produce packed.

- The aggregate number of trucks waiting for loading/unloading in each of the supplier, packhouse and market loading bay.
- The cost of transportation and the investment plan.

Kemito Pipfruit will be able to make informed decisions about how to best invest in packing machinery.

The environment maybe used to experiment with the following features:

- The number and type of machines at each packhouse.
- The variability of (un)loading, packing times, transportation times and demand.

1.4 The CM: inputs, outputs, content, assumptions, simplifications;

1.4.1 Inputs and Outputs

1.4.1.1 Experimental Factors (Inputs)

- Packing Machine Investment Plan (The number and size of each machine to install at each packhouse), varied, integer values above 0, comes in three sizes (small, medium or large).

1.4.1.2 Responses (to determine achievements of objectives) (Outputs)

- Percentage of trucks waiting no more than maximum number minutes at the supplier, market and packhouse loading/unloading bays.
- Percentage of produce waiting no more than maximum number minutes at the packing/loading zones.
- Discrepancy in cost between the existing investment plan and the simulation.
- Cumulative percentage of demand met overall and at each market.

1.4.1.3 Responses (to determine reasons for failure to meet objectives) (Outputs)

- Frequency diagrams of waiting time for each truck at the supplier, packhouse and market loading zones accompanied with the mean, standard deviation, minimum and maximum.
- Frequency diagrams of waiting time for each produce in the packing and distribution waiting zones accompanied with the mean, standard deviation, minimum and maximum.
- Time-series of mean queue size per hour for all queues.
- Machine Utilisation for each size of the machine in each packhouse (cumulative percentage).
- Loading Bay utilisation for each loading bay (cumulative percentage).
- Cumulative percentage of discarded produce, packed and unpacked.
- Cumulative percentage of trucks delivering produce after 5pm.
- Cumulative percentage of market and aggregate demand not met.
- Cumulative percentage of trucks which are turned away from loading/unloading produce.

1.4.2 Component Lists

The components for this conceptual model are:

- Produce with type (Avocados/Apples)
- Machines with given distributions of packing times and size.
- Trucks with given variable distribution of transportation times, shipment type and capacity (Supply trucks and market trucks).
- Suppliers with produce supply (thresholds) and fixed loading times.
- Markets with produce demand (thresholds) and loading times.
- Packhouses with given fixed loading times and storage capacity.
- Produce queues with produce type (Avocados /Apples) and storage capacity.
- Loading queues with queuing capacity.

For a detailed component list, see in the appendix.

1.4.3 Process Flow Diagrams

Both apple and avocado trucks/produce will have the same process flow diagrams.

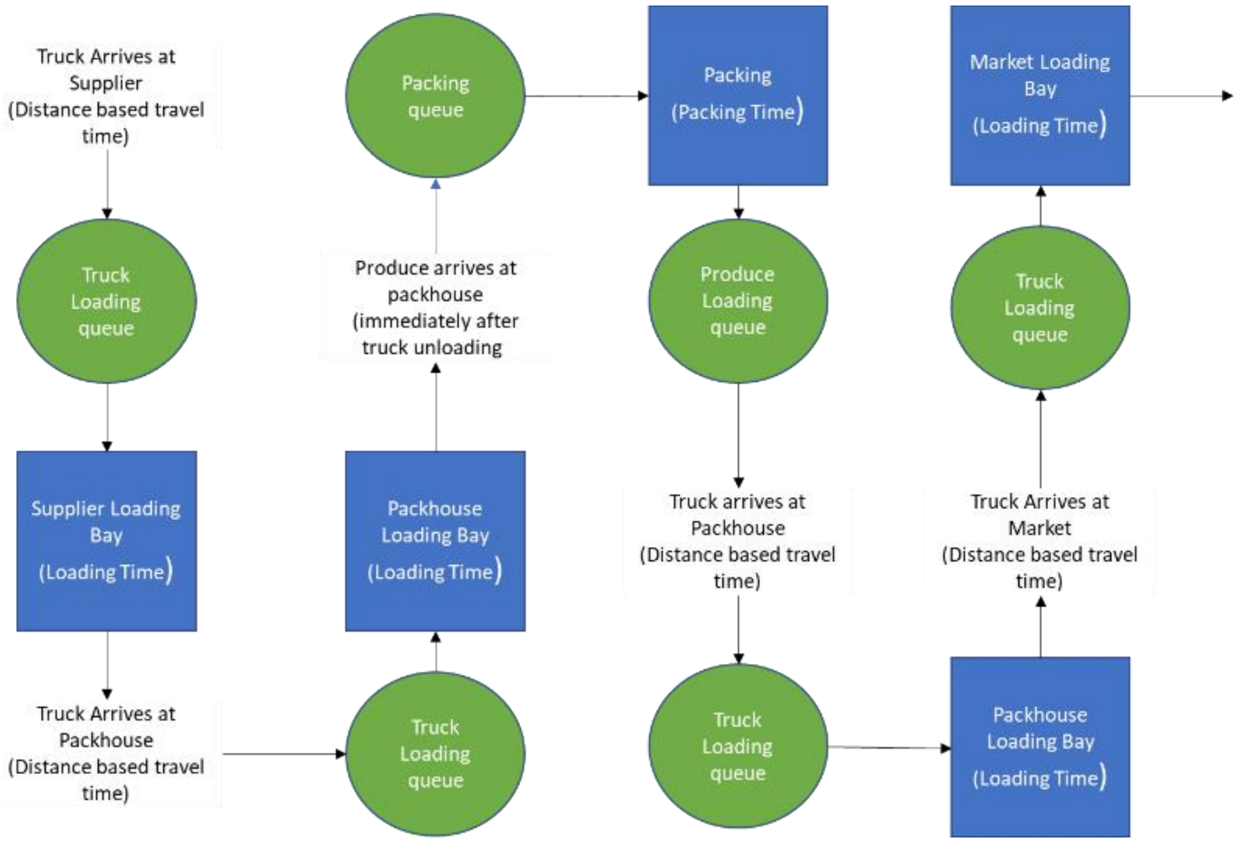


Figure 1: Process Flow Diagram for both Avocados and Apples

1.4.4 Logic Flow Diagrams

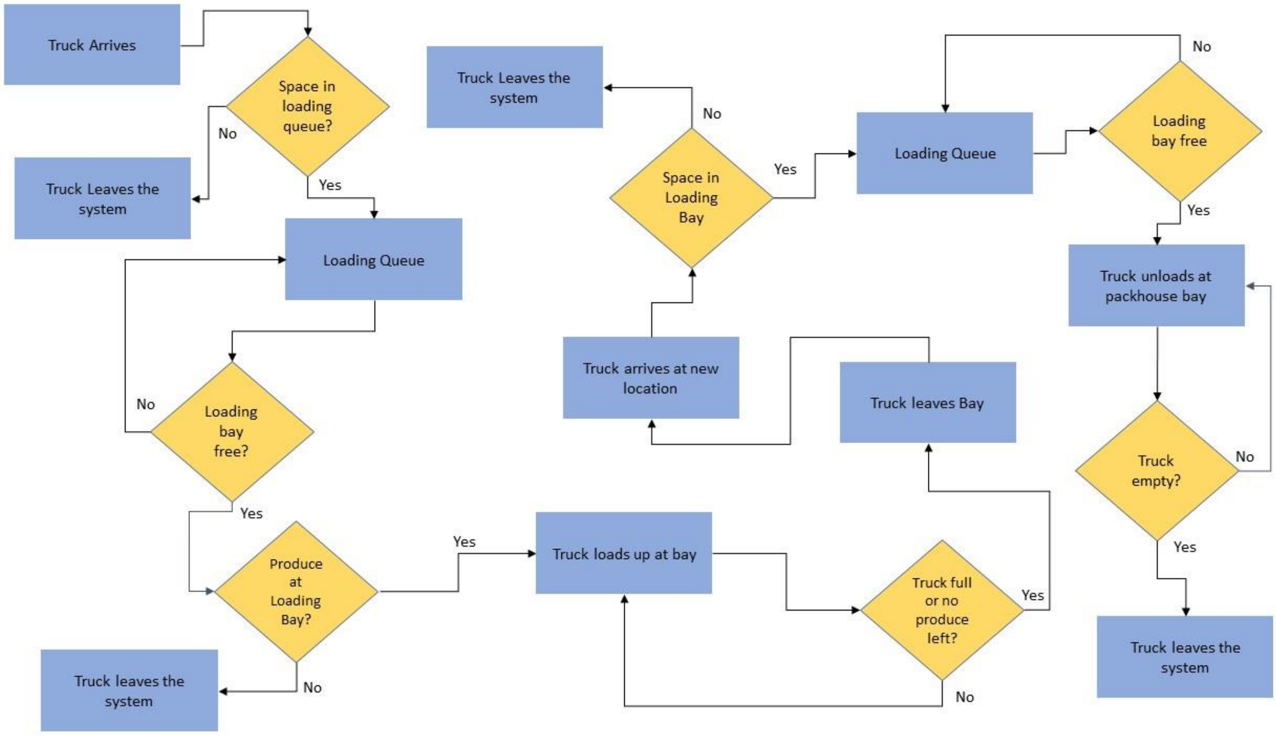


Figure 2: Truck Logic Diagram for both supplier to packhouse and packhouse to market produce delivery

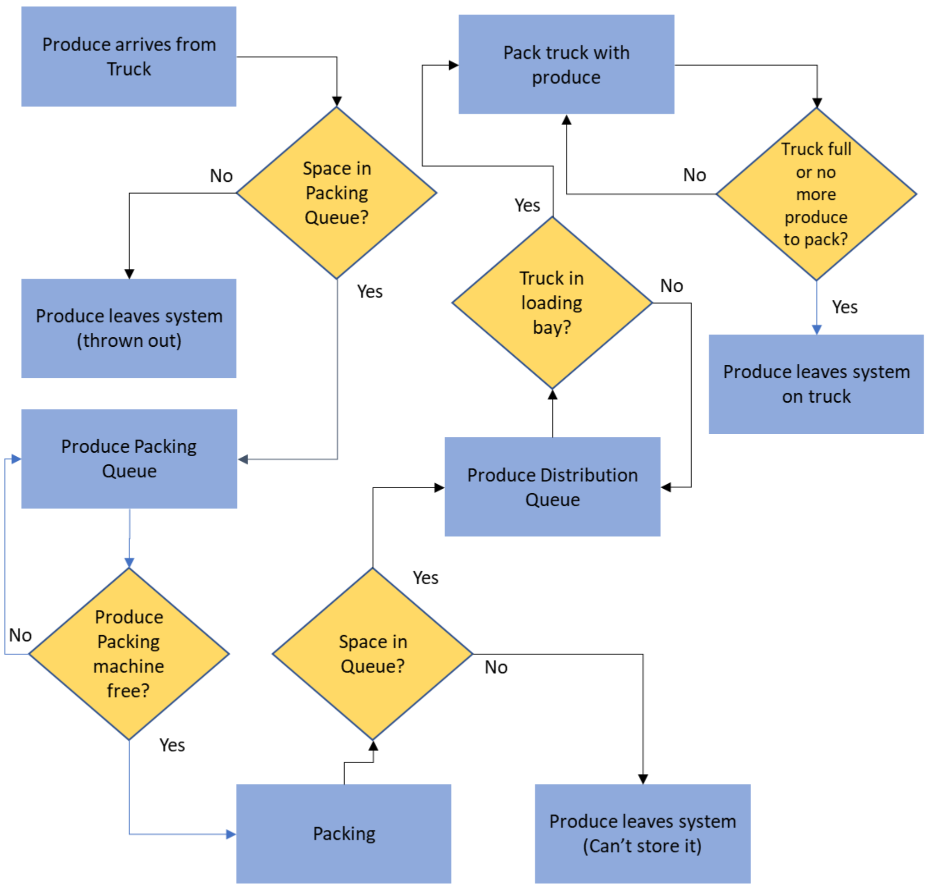


Figure 3: Produce Logic Diagram for either Avocados or Apples

1.4.5 Activity Cycle Diagram

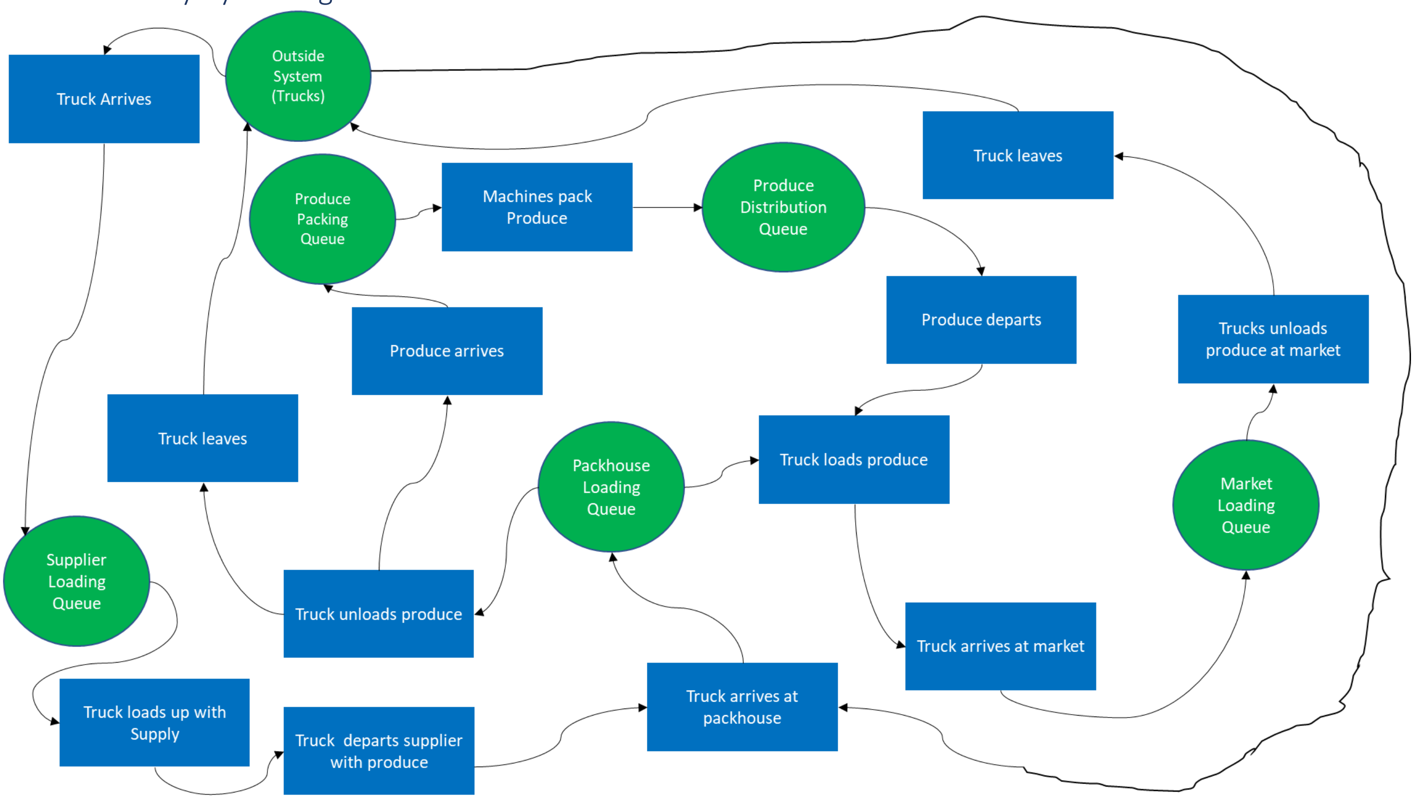


Figure 4: Avocado and Apple Activity Cycle Diagram

1.5 Assumptions

We have made the following assumptions:

- Apples and Avocados are to be shipped in different trucks along different routes and packed by different machines. However, produce is loaded/unloaded in the same bay.
- Administrative, parking, ordering and re-fuelling items are excluded activities to simplify the model, focusing on loading and packing.
- There will always be space in the loading/unloading bay queues for trucks (no bulking, jockeying or reneging).
- We are not concerned with the number of trucks used or where the trucks go after they exit the system. We assume they are under contract.
- Packhouses are open 7am – 5pm seven days a week and no new trucks will be added to the system after 5pm. Operations will continue until the existing entities in the system no longer flow.
- Trucks picking up the produce from suppliers will all arrive at the markets at 7am.
- Both produce types can be stored in the same queues, stored in a storage facility with finite capacity.
- Supply and demand levels are tracked by through additions and subtraction when entities flow to/from nodes.

1.6 Simplifications

We made the following simplifications:

- Model is decomposed into three separate stages; the loading, transportation and unloading of produce between suppliers and packhouses, the packing of produce, and the loading, transportation and unloading of produce between packhouses and markets.
- Trucks flow through the system with produce. The produce is the entity that flows through packing whist trucks flow through transportation.
- We are not concerned with what trucks do outside the system.
- Trucks transport grouped produce entities, assigned by type.
- Transporting produce with trucks required no queues and no rare events are included.
- There are two sets of trucks: Suppliers to Packhouses and Packhouses to Markets. Within each set is a subset: trucks which transport avocados and trucks which transport apples.
- Produce will always enter the packing system. Truck for suppliers will not return to the packhouse with no supply. Trucks will not drive empty to the markets.
- The distributions of the packing and transportation times will be decided upon analysing the data.

1.7 Experiments to run;

The following experiments need to be run:

- Simulate the model (transportation per day) for seven days for each of the ten historically weekly periods for the market.
- Run the simulation with different investment plans. Start with our original investment plan then adjust.
- Switch loading/unloading prioritisation. Prioritise trucks loading produce at the packhouses first. Run a separate set of simulations prioritising unloading next.
- Switch transportation prioritisation. Start with trucks shipping the quantities of produce specified in the optimisation model flows. After, experiment with trucks heading to locations based on lowest/highest number of produce received (Markets) and the amounts already delivered to packhouses (Suppliers).
- Run the prioritisation of loading and unloading produce in different simulations. Prioritise apples first then avocados.
- Switch the produce packing and distributing prioritisation. Prioritise apples first then avocados.
- Switch queue capacities. Start with no capacity. Add changing capacities in subsequent iterations.
- Switch the order markets are prioritised to be delivered to first and which suppliers are prioritised to be have their produce picked up from first.

7.7 Data

The distribution for the packing machine times for each size are derived through exploring the packingTimes.csv. This file can be found by following the link below.
"https://canvas.auckland.ac.nz/courses/32650/files/folder/Project"

7.8 JaamSim

Download JaamSim by following the link below. "https://jaamsim.com/"

7.9 Text Files

There are four capacity assignment files: Supplier to Packhouse and Packhouse to Markets for both Apples and Avocados. This is a subset of the used files to show an example.

Capacity	Start	Des
69	2	1
10	1	2
100	2	3
91	2	3
100	3	3
100	3	3
100	3	3
100	3	3
100	3	3
100	3	3
50	3	3
100	4	4
95	4	4
100	4	5
100	4	5
100	4	5
100	4	5
100	4	5
100	4	5

13	6.8	8.8	7.6
0.6	7	15.8	7
13.6	2	0.6	6.4
16	18	16	0.4
14.6	19.6	7.2	1.8
16	11.2	9.4	9.6
4	12.6	14.4	13.4
17.4	9.4	14.4	4
4.8	13.6	16.6	0.2
6.4	4	19.2	7.2
18.6	10.2	19.8	8.2
13.2	18.4	14.2	9.2
8.4	18	2	10.6
3.8	11.4	12.8	5.8
11.6	3	0.4	11.8

Figure 19: Example of Standard Deviation for Distribution Flows for Apples between Suppliers and PHS

21.66666667	11.33333333	14.66666667	12.66666667
1	11.66666667	26.33333333	11.66666667
22.66666667	3.333333333	1	10.66666667
26.66666667	30	26.66666667	0.666666667
24.33333333	32.66666667	12	3
26.66666667	18.66666667	15.66666667	16
6.666666667	21	24	22.33333333
29	15.66666667	24	6.666666667
8	22.66666667	27.66666667	0.333333333
10.66666667	6.666666667	32	12
31	17	33	13.66666667
22	30.66666667	23.66666667	15.33333333
14	30	3.333333333	17.66666667
6.333333333	19	21.33333333	9.666666667
19.33333333	5	0.666666667	19.66666667

Figure 20: Example of Means for Travel Costs for Apples between Suppliers and PHS

7.10 Distribution Plots

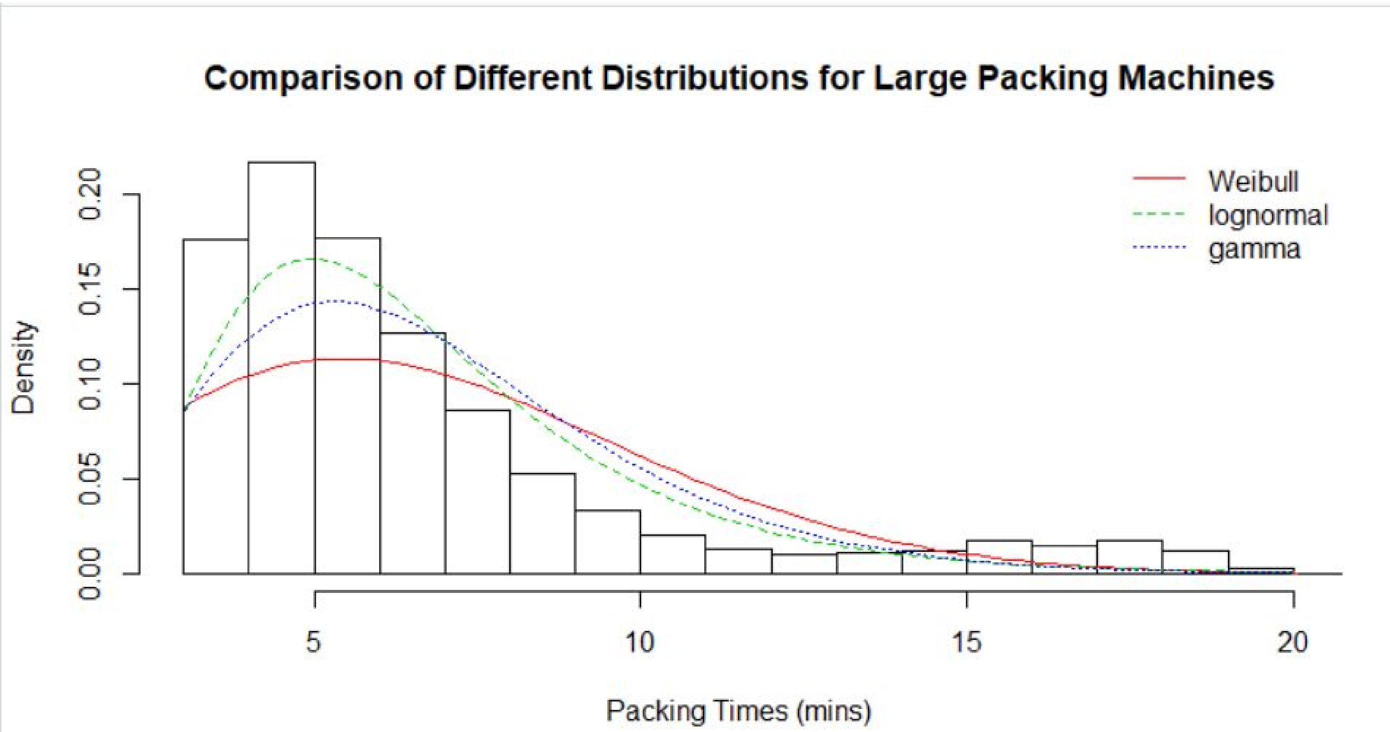


Figure 21: Distributions Fitted Large Machines

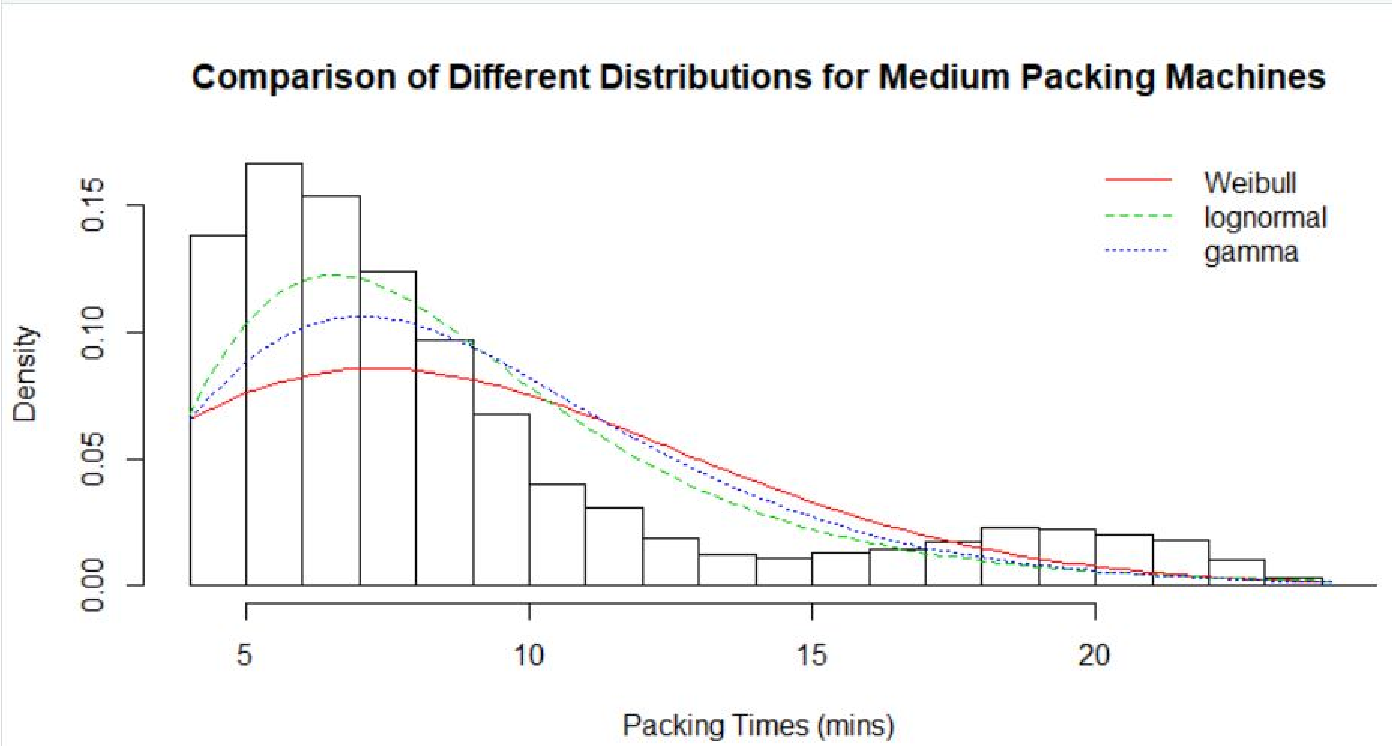


Figure 22: Distributions Fitted Medium Machines

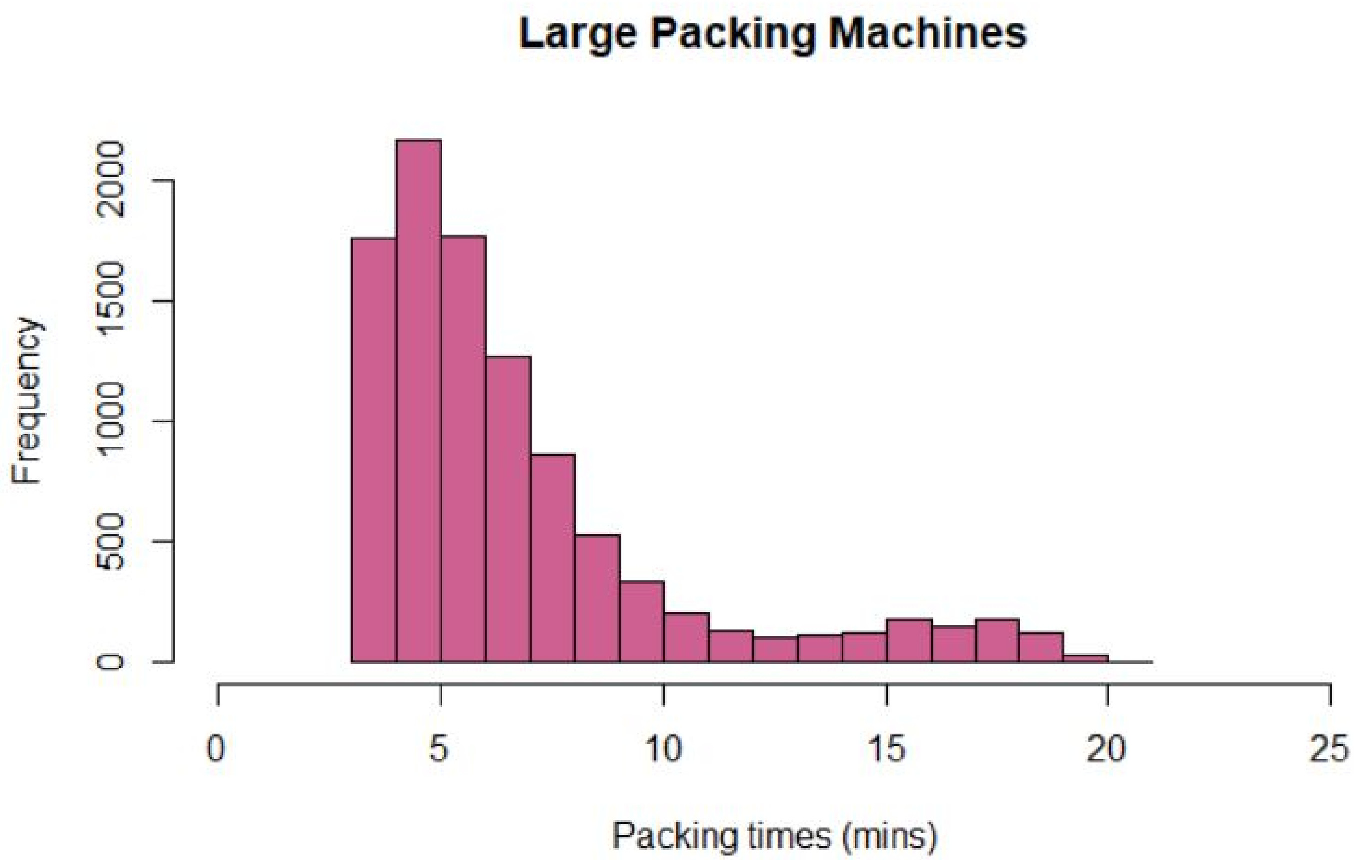


Figure 23: Large Packing Machine Times

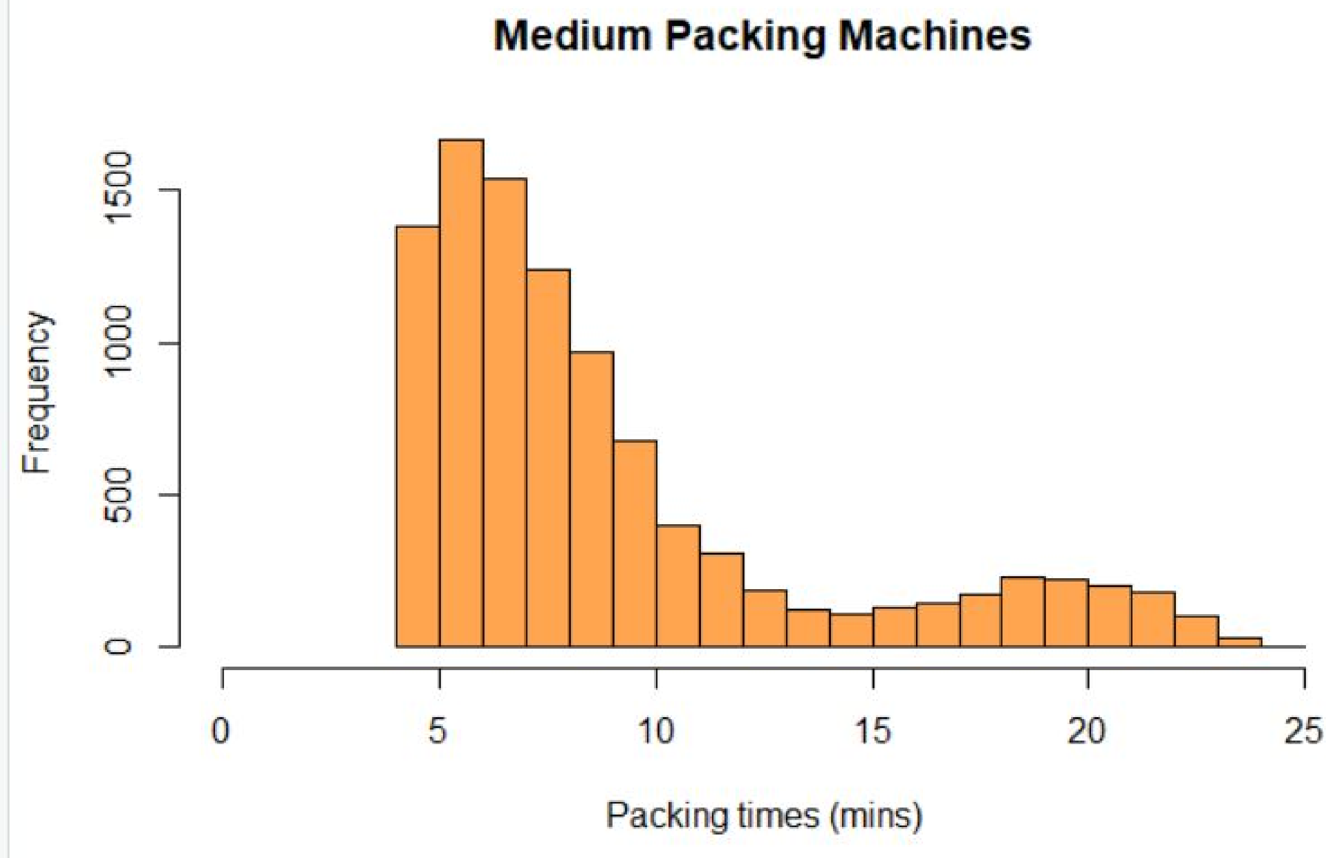
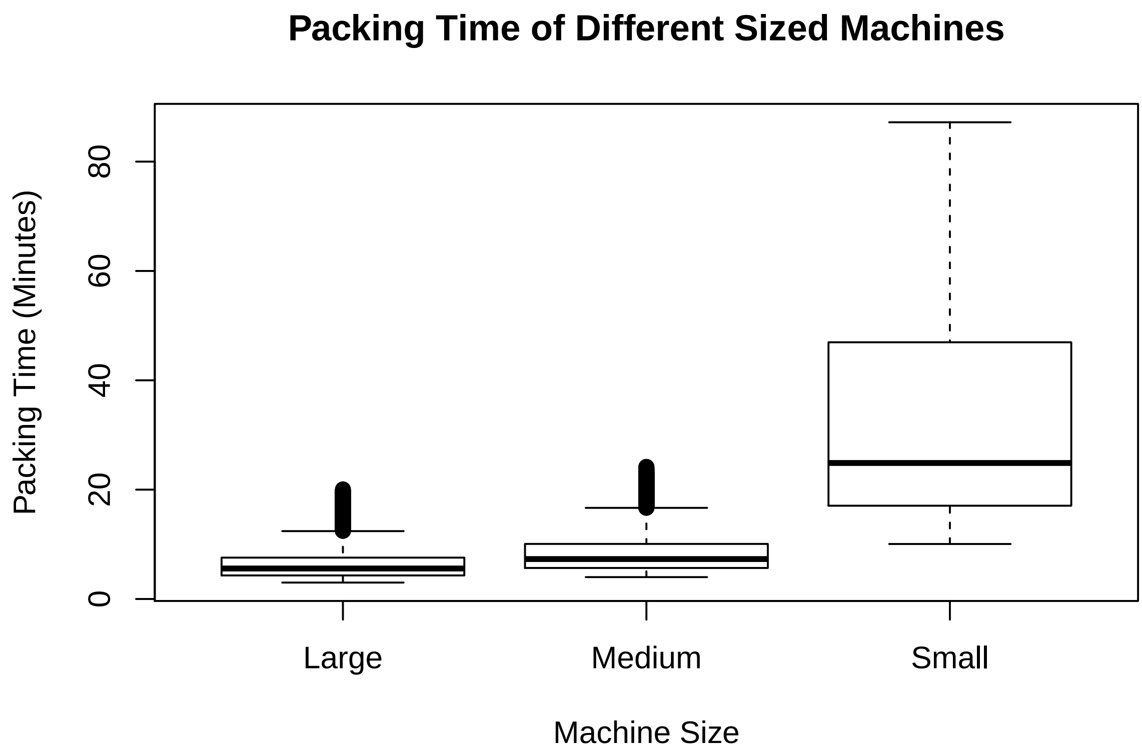


Figure 24: Medium Packing Machine Times

7.11 R Script for Data Analysis

```r
# setwd("D:/CM and Simulation/Project") for working in home drive

setwd("D:/CM and Simulation/Project") # for usb
PT.df <- read.csv("packingTimes.csv", header = TRUE)

boxplot(PackingTimeMins~MachineType,data=PT.df, main="Packing Time of Different Sized Machines",
        xlab="Machine Size", ylab="Packing Time (Minutes)")
```




```r
summary(PT.df)
```

```
##   MachineType    PackingTimeMins
##   Large :10000   Min.   : 3.006
##   Medium:10000   1st Qu.: 5.622
##   Small :10000   Median : 8.899
##                  Mean   :16.425
##                  3rd Qu.:19.109
##                  Max.   :87.195
```

```r
head(PT.df, 6)
```

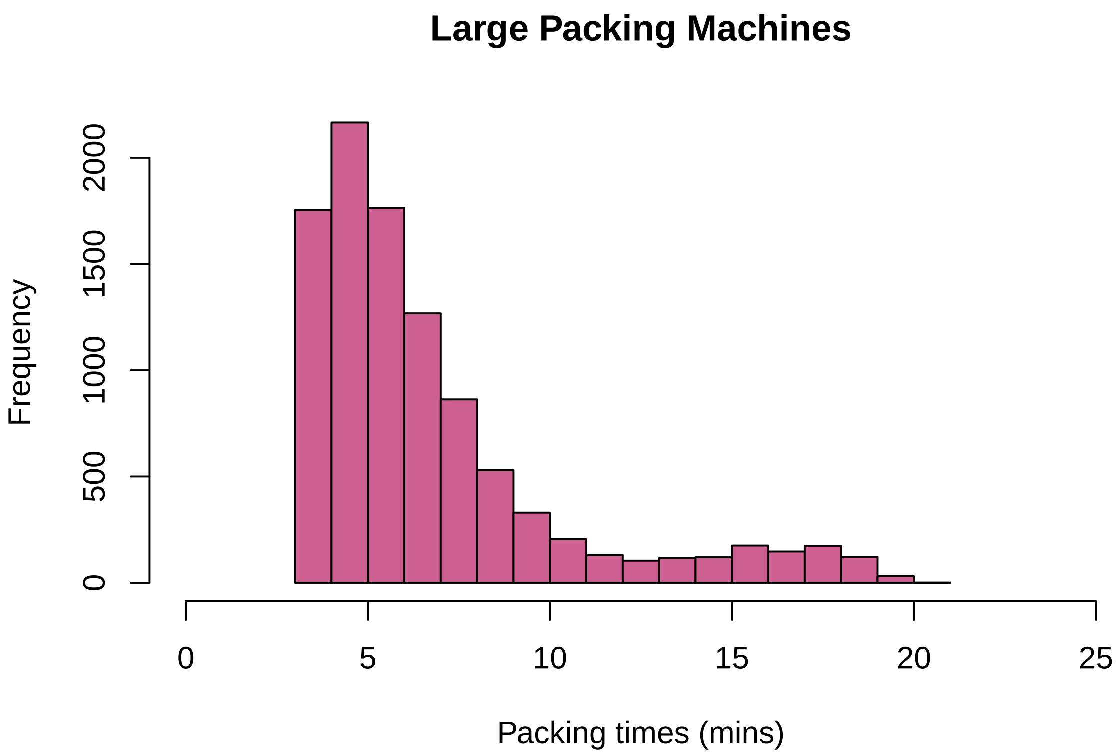
```
##    MachineType PackingTimeMins
## 1        Small       20.968162
## 2        Small       33.440080
## 3        Large        4.642006
## 4       Medium       16.993729
## 5        Small       11.625051
## 6        Large       16.745468
```

```r
small <- filter(PT.df, MachineType=="Small")
```

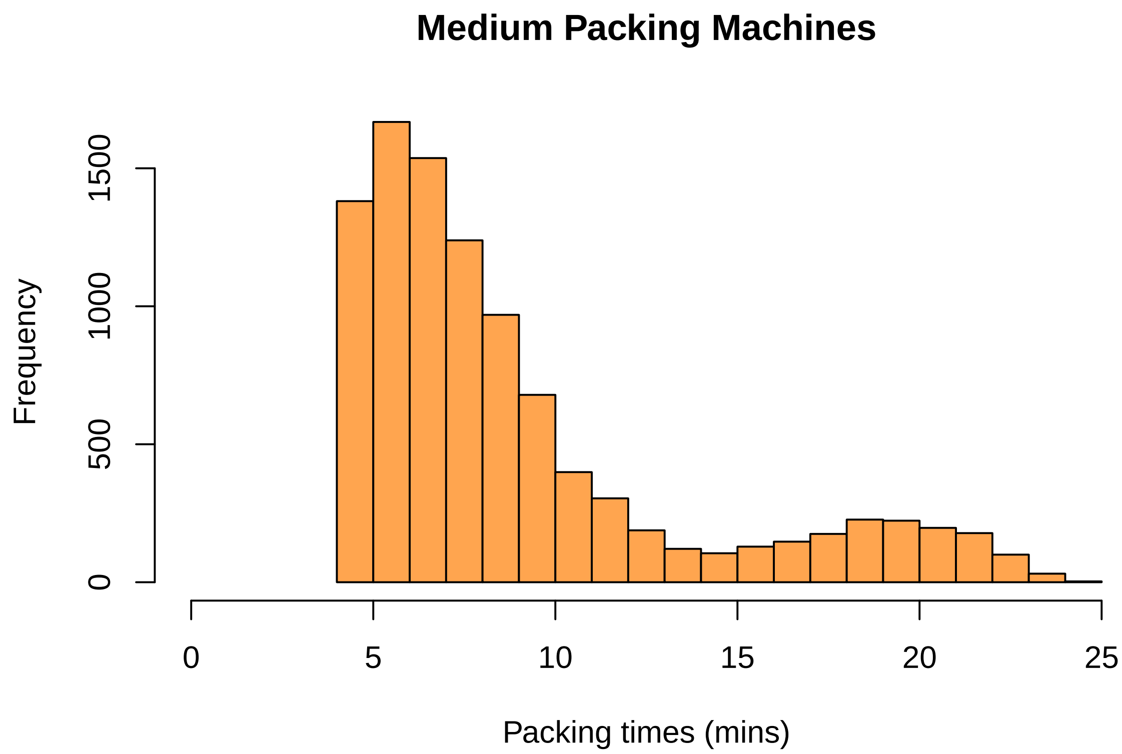
```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```r
medium <- filter(PT.df, MachineType=="Medium")
large <- filter(PT.df, MachineType=="Large")
```

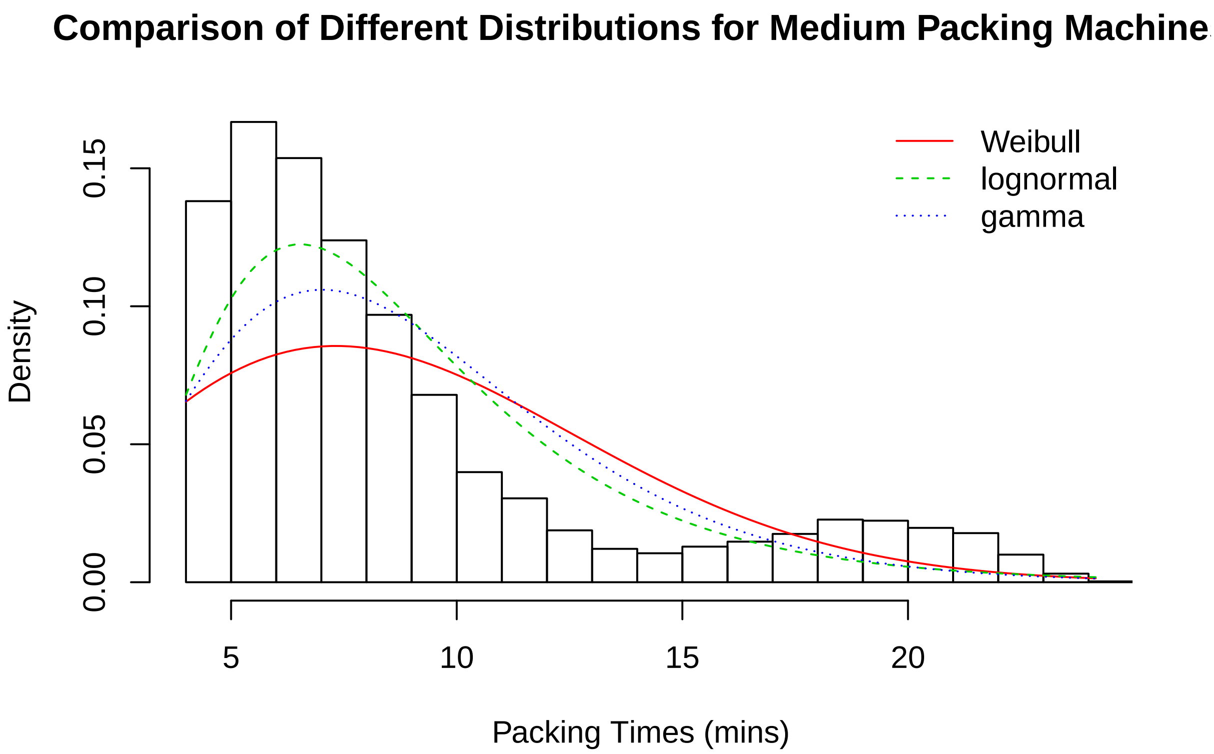
```r
# Large Packing Machines
hist(large$PackingTimeMins, col = "hotpink3", xlim = c(0, 25), main  = "Large Packing Machines", xlab =
```




```r
# Medium Packing Machines
hist(medium$PackingTimeMins, col= "tan1", xlim = c(0, 25), main  = "Medium Packing Machines", xlab = "Pa
```



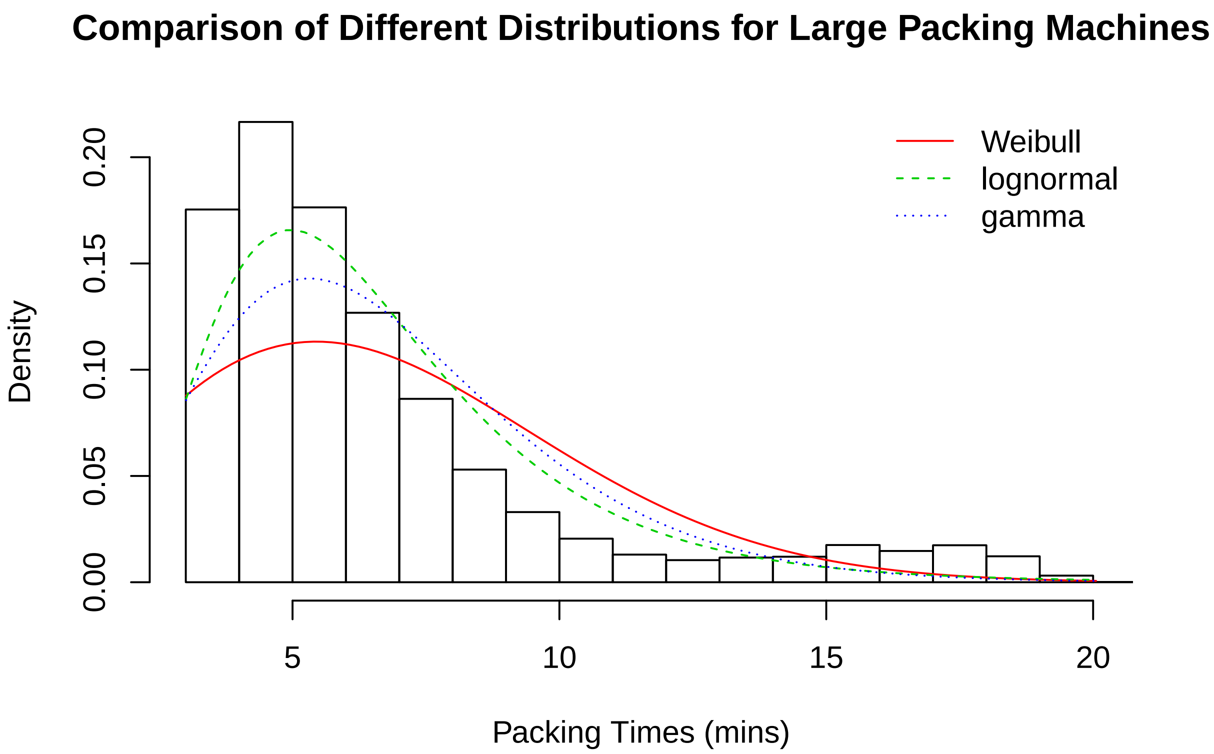

```
# Medium Packing
Mfw <- fitdist(medium$PackingTimeMins, "weibull")
Mfg <- fitdist(medium$PackingTimeMins, "gamma")
Mfln <- fitdist(medium$PackingTimeMins, "lnorm")
plot.legend <- c("Weibull", "lognormal", "gamma")
denscomp(list(Mfw, Mfln, Mfg), legendtext = plot.legend, main = "Comparison of Different Distributions f
```




```
summary(Mfln)
```

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters :
##          estimate  Std. Error
## meanlog 2.079229 0.004508270
## sdlog   0.450827 0.003187758
## Loglikelihood:  -27014.96   AIC:  54033.92   BIC:  54048.34
## Correlation matrix:
##          meanlog sdlog
## meanlog       1      0
## sdlog         0      1
```

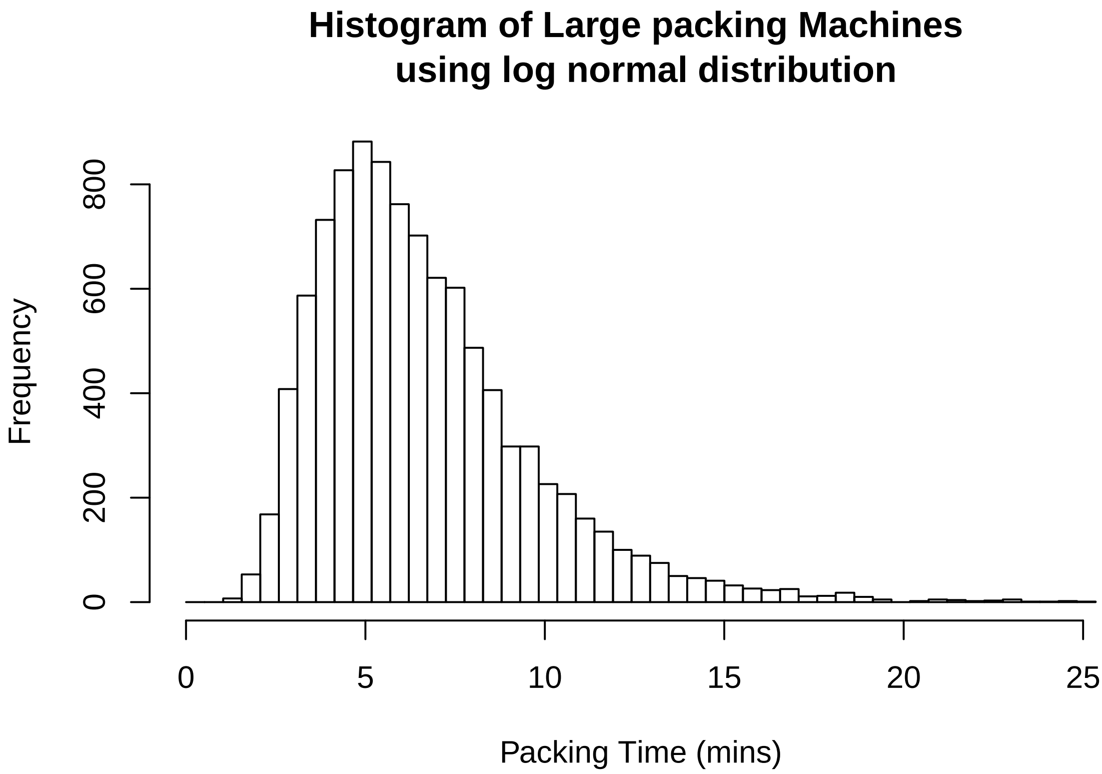
```
# Large Packing
Lfw <- fitdist(large$PackingTimeMins, "weibull")
Lfg <- fitdist(large$PackingTimeMins, "gamma")
Lfln <- fitdist(large$PackingTimeMins, "lnorm")
plot.legend <- c("Weibull", "lognormal", "gamma")
denscomp(list(Lfw, Lfln, Lfg), legendtext = plot.legend, main = "Comparison of Different Distributions f
```



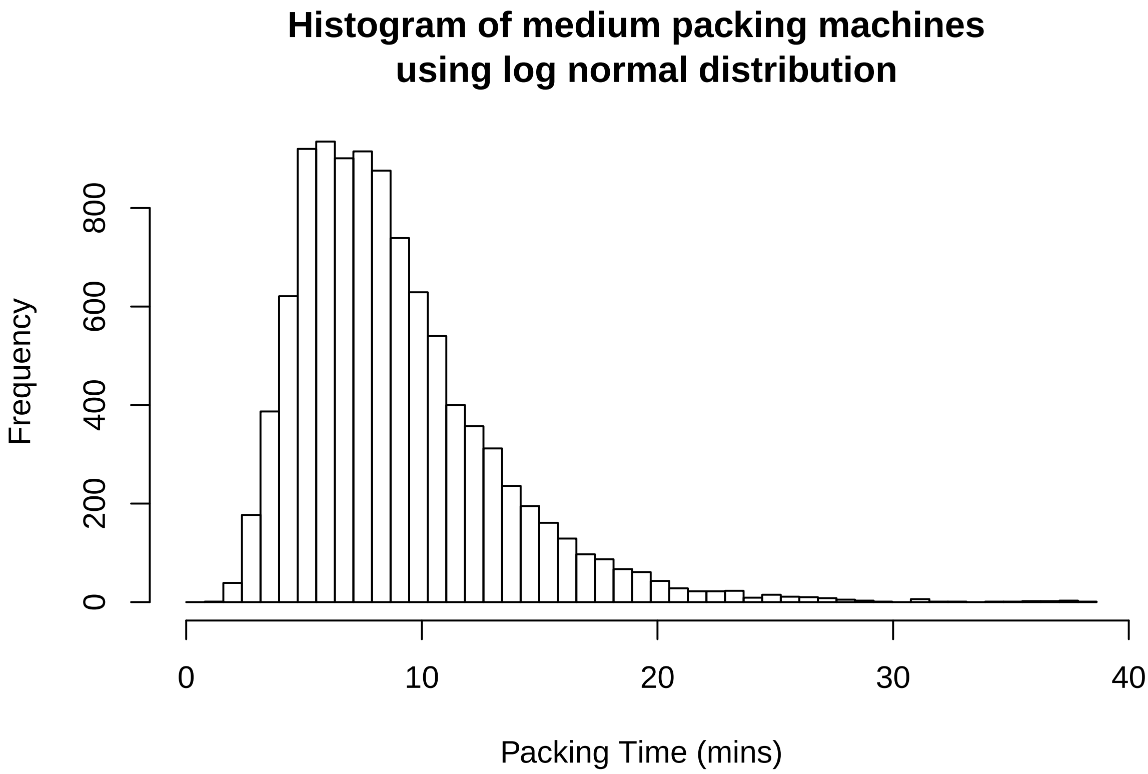

```r
summary(Lfln)
```

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters :
##          estimate  Std. Error
## meanlog 1.7963584 0.004397938
## sdlog   0.4397938 0.003109739
## Loglikelihood:  -23938.48   AIC:  47880.95   BIC:  47895.37
## Correlation matrix:
##         meanlog sdlog
## meanlog       1     0
## sdlog         0     1
```

```r
n <- 10000
dat <- rlnorm(n, meanlog = 1.7964, sdlog = 0.43979)

# create a vector of histogram breaks
x <- seq(0,max(dat),length=50)
hst <- hist(dat, breaks=x, main = "Histogram of Large packing Machines \n using log normal distribution
```




```r
m <- 10000
dat <- rlnorm(m, meanlog = 2.079, sdlog = 0.4508)

# create a vector of histogram breaks
y <- seq(0,max(dat),length=50)

# histogram the data
hst <- hist(dat, breaks=y, main = "Histogram of medium packing machines \n using log normal distributio
```



Histogram of medium packing machines
using log normal distribution
Frequency
0
200
400
600
800
0
10
20
30
40
Packing Time (mins)