BAN403

Simulation of Business Processes

Assignment 2
21.04.2023

Candidate Numbers:

1072, 1104, 1120

Contents

1.0 Introduction

The success of the newly established Burrito-business is dependent on choosing the right placement and service level of the operating locations. Optimization is a tool well suited for creating a placement which optimized location based on a set of predesignated constraints. To address the uncertainty involved in choosing the operating location, the solution is implemented as a discrete event simulation, which allows for the statistical analysis of the key performance indicators of the operation. The following statistics provide valuable insights for selecting optimal locations and assessing their economic feasibility. These insights will inform a service level analysis, which will evaluate the profitability of scaling up the operation based on the parameters of the discrete event simulation. Assumptions will be explained continuously throughout the report.

This report aims to address the following key objectives:

- *Consideration of profit estimated by optimization model against profit estimated with a simulation model.*
- *Analysis of whether it is economically viable increase the service level of the business.*

The report is structured as a pipeline, whereas the input data is analyzed before it is applied to an optimization and simulation model. The outputs of these models will be presented and used as a framework for addressing the key objectives of the report.

2.0 Input Analysis

For input data, the location data is regarded as a grid. The building locations are denoted as the set *I*, and possible truck locations as the set *J*. For example, *I2* is the second building location i.e. *(6,8)*, and *J1* is the first possible truck location i.e. *(0,7).* Manhattan distances is used to calculate distance between each building and all the possible truck locations, giving us parameter *c*. The fixed cost, revenue per unit and ingredient cost per unit are defined as the parameters *f, r, k*.

The following input data analysis is done using Jupyter Notebook. Using *Service1_sample.txt* and *Service2_sample.txt* the distribution of the data is tested using the *fitter* package. Outliers are not removed due to these having a natural explanation for service times from unexpected events. From both table 1 and table 2 it is observed that the beta, erlang, and gamma distributions are grouped quite closely together. Even though norm and exponential have lower SSE (sum square error), the p-value together with the visual input from the fitted graph in *service-speed-input.ipynb* indicates that the Beta distribution is a better fit for both service samples. The goodness-of-fit value for *service 2* confirms this by beta having the highest value of 0.8328. However, for *service 1* Gamma distribution has the highest goodness-of-fit value with 0.9551.

Table 1: Fitter summary for Service1_sample.txt (red indicates best fit)

Distribution	SSE	AIC	BIC	KS statistic	KS p-value
Beta	0.0387	769.3056	-10131.1262	0.0179	0.8998
Gamma	0.0388	765.2576	-10137.1756	0.0189	0.8594
Normal	0.1723	1040.0719	-8652.2889	0.0998	0.0
Exponential	0.238	697.067	-8329.5063	0.1312	0.0

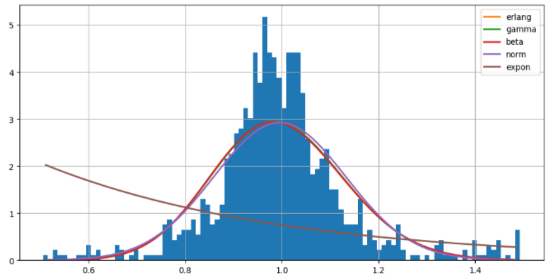
Table 2: Fitter summary for Service2_sample.txt (red indicates best fit)

Distribution	SSE	AIC	BIC	KS statistic	KS p-value
Beta	0.0559	777.8204	-9764.6942	0.0196	0.8309
Gamma	0.0555	759.9266	-9778.253	0.0199	0.8167
Normal	0.2049	1034.6043	-8479.1861	0.0943	0.0
Exponential	0.2765	688.7569	-8179.5828	0.1348	0.0

The data used is a limited sample from the real distribution. The low variations between beta, gamma, and erlang make the beta distribution hypothesis uncertain. Moreover, it seems implausible that one truck has a different distribution than the other, based on the best goodness-to-fit values. A closer look at the nature of beta and gamma distributions reveals that beta is a probability distribution with an area under the curve of 1, while gamma is a distribution for modeling waiting times and duration. A gamma distribution hypothesis seems more probable based on the time aspect of the observations. The script produces two files per service, serviceX-dist.txt with table 1 and table 2, and serviceX-param.txt with gamma parameters, mean, gamma mean, st.dev, min, max, and goodness-of-fit values.

To collect data from a similar business the local café “Merino kaffebar”. The number of observations were 58 which gives a very small sample size. A stopwatch was used as time measurement, this together with inaccuracy of when the serving time begins, gives high volatility for each observation. When compared to the burrito trucks, the café had a much lower proportion of meals served compared to drinks, giving a much lower mean compared to the service data. This makes it highly improbable that the new server’s times are drawn from the same distribution as the others, due to the much higher efficiency. After importing the data, *nk-observations.csv*, to *service-speed-input.ipynb*, the data gathered went through the same process using fitter. With the lowest SSM (6.479892), highest KS p-value (0.032996), and highest goodness-of-fit value (0.0341) a beta distribution looks like the most convincing distribution hypothesis. However, following the same reasoning as service 1 and service 2, an assumption for gamma distribution is taken, despite only having a goodness of fit value of 0.0006. The remaining output data can be found in *NK-dist.txt* and *NK-param.txt*.

Figure 1: Fitted distribution plots over histogram of walking 1.



In the script walking-speed-input.ipynb the data configurations for the walking speeds can be found. To identify the number of blocks walked per observation a classification model was first used, but this provided poor results. Looking at the histogram of the data in the script explains this result, a visual representation does not identify any clear classification trends using the human eye. As the assignment states that customers have an average walking speed of 1 block per minute, a reasonable assumption is that all observations within ± 0.5 of each integer number contain those who walked that integer number of blocks. This assumption excludes outliers that walk exceptionally fast or slow, especially over multiple blocks. The script imports the data through a *for*-loop which reads Walkingfrom_building{i}.txt. To follow the assumption explained, a new column is made that contains the data rounded to the nearest integer value, this represents distance. With total time and distance available, the average speed for each customer is calculated. The mean for all five datasets is within 0.993-1.004, which gives credibility to our assumption and indicates that the outliers lost do not affect the result with too much significance. Next a function is created based on the fitter code form service-speed-input.ipynb . This takes the data frames produced by the for-loop as input and provides the fitter results in IX-dist.txt. From figure 1 no distribution fits the dataset well, this explaining why all KS p-values and goodness-of-fit values are too low to use as acceptable criteria for distribution evaluation. When using lowest SSM as criteria, the strongest hypothesis is that I1 has erlang, I2 has beta, I3 has beta, I4 has erlang, and I5 has a beta distribution.

The dataset may not reflect the reality of walking speed from different buildings, unless it accounts for other factors like transport modes. The small sample sizes and the beta distribution assumption are also questionable. Normal-, gamma- and erlang distributions are more plausible based on empirical and theoretical grounds. Among them, normal distribution is the most likely, as it fits most of the data better, follows the central limit theorem, and matches real-world data patterns. Assuming normal distribution, the standard deviation of all the data sets is exported through IX-param.txt. A truncated normal distribution could be considered, this distribution would exclude natural outliers that were classified by ± 0.5 of the mean.

3.0 Method

3.1 Optimization Model

The optimization task is solved using the Gurobi solver in the framework of the AMPL IDE. The output of the models is contained in Appendix 3. The optimization model used as a basis for this report is contained in Appendix 1.

3.1.1 Base Optimization Model

The model used is a modified version of Gurobi's Burrito IP formulation (Guriobi Optimization, 2023). The first modification is done to allow placement of two trucks giving the constraint $\sum_{j \in J} x_j \leq 2$. The second modification is implementing the demand multiplier $a_{ij} = e^{-0.1c_{ij}}$. The placement recommended by the model is placing trucks in location *J3 (5, 12)* and *J5 (12, 0)*, which gives a profit of NOK 3514.11.

The base model for calculating the demand from each building has a limitation. The demand multiplier constraint is linear, which results in non-integer values for the demand from each building. For example, the demand from building I1 to the truck in J5 is 21.48 people. This is not realistic for a real-world scenario. To address this issue, the demand is converted to integer values by rounding down any decimal numbers. The decimal numbers are interpreted as individuals who almost decided to buy a burrito. The conversion is done by introducing the modifications and additions from Appendix 2 through the new integer variable z_{ij}. The new optimal solution gives a profit of NOK 3340. The truck locations remain the same as in the previous model but the demand from each building has changed.

Building	*I1 (9, 0)*	*I2 (6, 8)*	*I3 (1, 12)*	*I4 (12, 4)*	*I5 (2, 14)*
Demand	21 (21.48)	16 (16.98)	24 (24.13)	14 (14.75)	14 (14.56)
Truck location	*J5*	*J3*	*J3*	*J5*	*J3*

Table 3:Optimal truck placement and demand

3.1.2 Modifications to optimization model

To evaluate the service level of the food truck placement scheme, some adjustments are needed to allow for the option of adding more trucks to the city. However, based on this model and data, the optimal solution remains the same regardless of the number of trucks. In order to simulate this option, even though it is not optimal, the number of trucks is fixed at 3 with the new truck located at *J6* and serving a demand of 25 from building *I2*. This yields a profit of NOK 2880. The rest of the buildings route to the same trucks as before.

3.2 Discrete Event Simulation Model

This discrete model simulation is based on a loop, as the demand is finite and as the model uses internal logic to determine if a given customer is unsatisfied or has eaten in a given day. The base model is shows in Appendix 2 Figure 2.

3.2.1 Base Simulation Model

The base simulations consist of five identical buildings that distribute potential customers to the nearest food truck. The layout is based on the optimal solution of the optimization model. The buildings generate a customer pool that matches the demand forecasted by the optimization model for each building. As the demand is dependent on a global weather parameter, the number of possible customers is 20% higher than that which is predicted in the optimization model. This demand is rounded down when a non-integer demand is determined. When generated, each customer from a given building is given a *building number, number of blocks* from the building to the food truck, the customers *walking speed* pulled from a normal distribution and a *uniform probability* which is used to determine meal routing later in the model.

Each building has a queue where customers wait for the next break after being assigned values. At each break, a *WorkBreak*-threshold lets the *BreakStart* servers move customers from the queue. The demand depends on the weather condition (drawn from a discrete distribution every two hours) and is calculated by the *WeatherCond* EntityGate. The demand defined as the maximum value of zero and the weather multiplier times the base demand minus the customers rejected twice and the *NonEligeble* customers for that day. This reflects the actual demand for each meal and weather condition, not the theoretical maximum. Customers blocked by the gate go to the breakroom and get lunch elsewhere.

Customers passing the gate are routed to the food truck or the breakroom, depending on the demand distribution of 20% for breakfast, 60% for lunch and 20% for afternoon break. The *UniProb* attribute determines the routing probability for each customer. Customers going to the FoodTruck are delayed by their walkspeed and distance to the food truck queue. They are assigned a service time from an exponential distribution with the mean from the input analysis. This service time is an approximation based on the results from the input analysis. Customers can leave the queue and go back to their building if they think they will not get their burrito before the break ends. This is done by setting the *RegedeTime* of the queue to the breaktime minus the travel time to and from the queue and the service time. Customers leave if they are not served within this time limit.

If the customer is processed by the food truck, it is assigned an *eaten* value of 1, which represents that they have eaten a burrito this working day. If they leave the line before being served, a 1 is added to their *reject* parameter, which counts the number of times a given customer is rejected. Before walking back, each customer is checked whether or not they have been rejected twice. If the customer has been rejected 2 times, the customer is discarded from the system, as they are no longer a viable customer because of their dissatisfaction with the food truck. The number of rejected customers is kept count of by a counter below the *EntitySink*. Customers which have been rejected less than two times start walking back and end up at the *Breakroom* of the building. When the break is over, the people who have eaten this day reset their *eaten* variable and are placed in the *NotEligible* queue until the next working day. The rest of the customers are sent back into the *Building*. Customers who have eaten a burrito get assigned a new *UniProb* probability the next day, to represent the fact that people may vary their eating habit for each day.

3.2.2 Modifications to Simulation Model

This part will present the changes made to the base model in order to accommodate analysis of the service capacity in the food trucks. Each scenario will be stored in their own .cfg file in the Appendix, and the output will be analyzed in the output analysis.

Expanding the number of workers per truck

Determining the number of workers operating each truck has an impact on the efficiency and brand image of the food trucks. Since the optimization model assumes that every customer who arrives at the truck will be served, increasing the efficiency of serving is not practical to measure within this framework. The best way to analyze the feasibility of increasing the staff size on each truck is therefore applying the new scenarios to the simulation model. The main change to the model lies in the capacity and available workers, which makes it possible for the food truck to handle 2 customers at a time without bottlenecks.

Expanding the number of trucks in Burritoville

In addition to analyzing the possibility of increasing staff at each truck, this report will consider the consequences of establishing another food truck in Burritoville. As mentioned in section
3.1.2 Modifications to optimization model, adding a new truck changes the optimal food truck placement and the estimated demand of Building 2. The proposed solution projects lower profits for Gurobi, but the potential impact of uncertainty on the increased capacity can be evaluated using a simulation model. The solution involves adding a new truck that serves only Building 2, which requires a dedicated server. This report will compare two scenarios with the new truck configuration, one with one worker per truck and another with two workers per truck.

4.0 Output Analysis

The simulation is non-terminating, as customers only leave after two rejections or when the simulation stops. This could last indefinitely. The throughput rate of each truck reveals the system's stability and capacity to meet customer demand. This helps determine the stabilization point and the optimal service level without harming the brand image. The system reaches a steady state if $\rho = \lambda/\mu < 1$. To estimate ρ and evaluate the service performance, the average of 1000 replications of 30-day simulation periods for each scenario is used. The inputs are identical, but the seed for uncertainty varies in each iteration. The scenarios are independent and identically distributed, so the central limit theorem can be used to approximate the sample mean distributions by normal distributions. The total break time for these 30-day runs are 3600 minutes, which is the basis of the throughput of the model.

In order to compare the service levels against each other, four distinct scenarios are created in JaamSim, according to the specifications presented in 3.2.2 Modifications to Simulation Model. Scenario 1 is the base model, where two trucks are located according to the optimization output, with one worker in each. Scenario 2 is similar, yet the number of workers has changed to two per truck. Scenario 3 adds in yet another worker, which makes the total number of workers per truck 3. Scenario 4 adds in a third truck, which is located according to the modified optimization model, with one worker in each truck. Lastly, Scenario 4 has three trucks, but with two workers in each.

	Truck (placement)	Workers (per truck)	Service Time (minutes)	Service Queue (minutes)	Throughput (per minute)	Steady State (ρ)	Rejected Customers (n)	Cycle Time (minutes)	Total Profit (NOK)
Scenario 1	3	1	2.8753	14.7280	0.1880	0.5400	36.4	16.7573	34719.74
	5	1	2.8911	13.8216	0.1835	0.5299	16.7	27.5282	36228.32
Scenario 2	3	2	2.9168	11.8183	0.3442	1.0037	17.8	14.7492	70614.26
	5	2	2.9301	8.7749	0.2723	0.7979	1.8	17.5723	57326.00
Scenario 3	3	3	2.9484	8.5991	0.4135	1.2193	4.4	11.5700	87292.22
	5	3	2.9416	5.1730	0.2776	0.8168	0.0	10.3485	58936.16
Scenario 4	3	1	2.8999	13.6279	0.1769	0.5125	20.5	16.2980	34337.66
	5	1	2.8903	13.8487	0.1840	0.5313	16.6	27.5869	36344.72
	6	1	0.7278	3.7930	0.2065	0.1503	0	17.5313	43605.38
Scenario 5	3	2	2.9451	9.2873	0.2842	0.8371	4.6	12.2618	59440.10
	5	2	2.9331	8.7940	0.2723	0.7988	1.8	17.6126	57328.46
	6	2	0.7261	1.4072	0.2065	0.1499	0	10.2257	43605.62

Table 4 – Aggregated output of JaamSim based on each truck server system

If each food truck is considered its own system, most of the trucks will eventually reach a steady state, as the ρ is lower than 1. Truck 5 in *Scenario 2* and *3* has a higher ratio between its throughput and service rate than 1, which indicates that the system will not reach steady state. For these two observations it is not possible to determine that the system will never reach this state, because of the limited run time of 30 days. If the number of customers were to reduce by people getting rejected, a steady state might be reached.

Regarding the performance of various simulation models, it is useful to examine independent systems individually and aggregate data for each scenario as a whole. Scenarios where service levels have increased perform better overall on measurements of average queue waiting time, number of rejected customers, and cycle time. As the number of customers has a maximum limit, an increase in throughput and processing rate can be observed as service levels increase and fewer people are rejected, which increases demand in later stages of the simulation. The service times of Truck 3 and Truck 5 are consistent across the scenarios and comparable to each other. Since they reflect the time spent per unit, they are expected to remain stable under different conditions. Truck 6 has a markedly lower service time than the other two trucks, which reflects the different data sources which estimated these distributions.

(aggregated)	Total Profit (NOK)	Profit per Day (NOK)	Optimization Profit Total (NOK)	Mean Service Time (minutes)	Mean Queue Times (minutes)	Sum Rejected (n)
Scenario 1	70 948.06	2 364.94	5273.33	2.883239	14.275	53.192
Scenario 2	127 940.26	4 264.68	5273.33	2.923491	10.297	19.632
Scenario 3	146 228.38	4 874.28	5273.33	2.945081	6.886	4.511
Scenario 4	114 287.76	3 809.59	5780	2.172719	10.423	37.240
Scenario 5	160 374.18	5 345.81	5780	2.201498	6.496	6.495

Table 5 – Aggregated output of JaamSim based on each scenario

One of the main findings is that increasing the number of workers and trucks leads to higher profits in both models. However, the profit from the optimization model is always higher than the profit from the simulation model when extrapolated to 30 days. This is because the optimization model does not account for uncertainty and brand image, which are important factors in a real-world scenario. The simulation model captures these aspects and allows us to compare how a given system performs against its optimal counterpart. Optimization models and simulation models are different tools for solving complex problems. Optimization models find the best solution given a set of constraints and a goal, while simulation models show the variability and randomness of a system and how different scenarios affect a solution, which combined form a powerful framework for placement analysis.

5.0 Service Level Analysis

Assessing the service level can be done by considering each scenario's impact on the most important KPIs of the business. Increasing the number of workers increases the profit by serving more customers. By adding another employee to the trucks in *Scenario* 2, the cycle time of the Truck 3 improves from 16.76 to 14.75, and Truck 5 from 27.5 to 17.57. The revenue impact of increasing the number of workers from 1 to 2 for each truck is measured as increase of 80.32%. As long as the relevant costs of hiring an extra two workers do not exceed 1 900 NOK per day, this increase is profitable for the business. Increasing the number of workers from two to three per truck has a slightly lower impact on the financials of the trucks, as it only increases revenue by 14.3%. The costs associated with hiring two of these workers must therefore be lower than 609.6 NOK per day to make this service level profitable. Based on the financials of this potential increase, the optimal level seems to be the one presented in *Scenario 2* with two workers in each truck.

In addition to the financial perspective, one can also make use of the relevant cost framework in order to account for hidden costs or benefits of increasing the number of workers in each truck. $Relevant\ Costs = Real\ Costs + Oppurtunity\ Costs \pm Externalities$. Until this point only the actual cost and alternative costs have been considered. Metrics such as the queue time and number of customers rejected are important when deciding which service level to operate the business at. The simulation internalizes some externalities in the form of bad customer experiences. If a customer can't buy a burrito twice, they will never consider buying from that specific food truck again. If the queue is too long, the loss in demand is reflected in lower revenue. The decrease in queue times for each extra worker added to the truck is 27.9% for one extra worker and 33.1% for two extra workers. Other externalities include the cost of losing goodwill, which might spread around the city if customers repeatedly have a bad experience of long queues of service times with the business. The business should aim toward keeping the queue level as low as possible, which is in *Scenario 3*.

The option of adding another food truck to the placement scheme has been explored in the simulation model as *Scenario 4* and *5*. Even though the new truck is much faster in processing customers, it has a small fallout in the city, which makes the option of expanding without increasing the number of workers worse in every major KPI except service time. The prospect of increasing both the number of trucks and workers seems like a scenario which encapsulates the best from both models. Although *Scenario 5* has the highest total profit, whilst also scoring high marks on the KPIs, it is important to understand the scope of this expansion. Overextension is one of the most common reasons for failure when expanding or starting a business to a new market. Gradually increasing the service level after establishing the business may prove vital in its success, as investing in three food trucks and hiring 6 workers is a significant investment. In order to recommend this strategy, one must complete thorough market research in order to be very confident that the parameters used to model the scenario are representative of the real food trucks.

6.0 Conclusion and Recommendation

The finding of this report shows that the initial service level of one worker per truck is inadequate for serving the entirety of Burritoville. The number of people rejected is very high, even though another truck is added in *Scenario 4*. From a purely profit centric view, *Scenario 6* is the option which delivers the highest profits over time, though this requires a large upfront investment which might not be feasible for a food truck business entering a new market. Compared to the output of the optimization model, the added uncertainty reduced the profitability on the whole. Although the salary of the workers is not specified, the increase in 2 to 6 workers might be outside the range of possibilities for the business at this point, even though scaling is a possible route of expansion later in the lifespan of the business. In order to capture the demand and reduce the risk of tarnishing the brand image by having long lines, having two workers for each truck is a therefore a must have criteria for establishing this business. This report will therefore recommend that *Scenario 2*, with two workers per truck, with two trucks in *J3* and *J5* is chosen as the given service level of the business.

Bibliography

Guriobi Optimization. (2023, April). *Burrito Optimization Game*. Retrieved from Gurobi: https://www.gurobi.com/burrito-optimization-game-guide/

Laguna, M., & Marklund, J. (2013). *BUSINESS PROCESS MODELING, SIMULATION AND DESIGN 2. edition.* Boca Raton: CRC Press.

Appendix 1

1.1 Burrito Trucks IP Model

Sets and Indexes

$i \in I$: Set of customers
$j \in J$: Set of trucks

Parameters

d_i: Demand for customer $i \in I$
c_{ij}: Travel distance from customer $i \in I$ to truck $j \in J$
f: Fixed cost for Gurobi for placing a truck at a potential location.
r: Revenue per burrito sold.
k: Ingredient cost per burrito sold.

Decision Variables

x_j: Binary. 1 if a truck is located at location *j*, and 0 otherwise.
y_{ij}: Binary. 1 if the closest truck to customer $i \in I$ is at location $j \in J$, and 0 otherwise.
a_{ij}: Demand multiplier for customer $i \in I$ to truck $j \in J$

Objective Function

$$Max \sum_{i \in I} \sum_{j \in J} (r-k) a_{ij} d_i y_{ij} - \sum_{j \in J} f x_j$$

Constraints

A customer can only be closest to one food truck location in the finished model.

$$\sum_{j \in J} y_{ij} \leq 1 \quad \forall i \in I \tag{1}$$

A customer *i* can only be served by a truck at location *j* if there is a truck located there.

$$y_{ij} \leq x_j \quad \forall i \in I, \forall j \in J \tag{2}$$

Both of the decision variables, *x* and *y*, are binary variables.

$$x_j y_{ij} \in \{1,0\} \quad \forall i \in I, \forall j \in J \tag{3}$$

1.1 Integer demand model

First of all, the objective function is changed in order to accommodate the fact that demand has to be an integer. This is done by adding a z_{ij} decision variable which is defined as a rounded down value of $a_{ij} d_i y_{ij}$. The max number of trucks is also defined. The changes to the model av presented below.

Decision Variables
z_{ij}: Integer. Rounded down value of $a_{ij} \cdot d_i \cdot y_{ij}$.

Objective Function

$$Max \sum_{i \in I} \sum_{j \in J} (r-k) z_{ij} - \sum_{j \in J} f x_j$$

Constraints
Allow for up to two trucks to be placed at one time.

$$\sum_{j \in J} x_j \leq 2 \tag{4}$$

The *z* value must be higher than or equal to the product of the *y*, *d*, and *a* values.

$$z_{ij} \leq \ y_{ij} \cdot \ d_i \cdot a_{ij} \quad \forall i \in I, \forall j \in J \tag{5}$$

The *z* value must be lower than or equal to the product of the *y, d,* and *a* values.

$$z_{ij} \geq \ y_{ij} \cdot \ d_i \cdot a_{ij} - 0.99 \quad \forall i \in I, \forall j \in J \tag{6}$$

The *z* value is defined as integers for all values of *i* and *j*.

$$z_{ij} \in \mathbb{Z} \quad \forall i \in I, \forall j \in J \tag{7}$$

1.2 Three Truck Model
In order to analyze the effects of adding another truck to the system, constraint number (4) is changed to allow for this. The change to the model is presented below.

The number of trucks to be placed is defined as 3.

$$\sum_{j \in J} x_j = 3 \tag{4}$$

Appendix 2

Figure 2 – Base Model