Starbucks Deployment Tool to Optimally Assign Employees

IND E 495 **By: Carter Herron, Eric Kammers, Dimitri Sastropranoto, Mohammed Said, Yuchao Fan, Weihuan Fu** **Class of 2017**

Project Introduction

Problem Statement

Increase Starbucks store manager's capability by creating a staffing tool that level-loads tasks across a given amount of employees and minimizes customer wait time.

Our project will aim to develop an improved and more dynamic version of the current staffing tool by forming a new backend algorithm that can provide store specific, and time specific outputs.

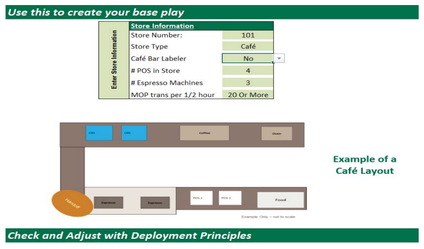
Example Question:

Manager: What tasks should my 10 Starbucks employees be assigned to during the 9:00am shift at the University Village location so customer waiting time is minimized?

Current State vs. Opportunity

Current State	Opportunity
Keeping inputs constant, it produces the same output for every store	Treat each store uniquely and make allocations based on store specific historical data
Allocates workers using an excel based V Lookup Table	Allocate workers by comparing the average customer time-in-system between allocations and level-loading utilization
Inputs •Ask for an estimate of the MOP transactions per ½ hour •Does not ask for number of employees available at that time	Inputs •Ask for a time range to allocate based off of historical data •Ask for an estimate of how many employees to allocate
Output •"What's possible" - # of transactions per ½ hour •Primary and secondary responsibilities •Service Standards	Output •Expected time in system for customer and partner utilization •Visual aids •Minimum # of partners so as to not be understaffed

Current Playbook Input Display



Resource Allocation Algorithm

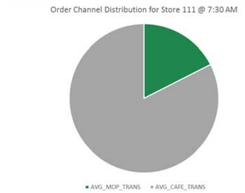
Objective

Use inter-arrival rates, average service times, and product distribution to optimally assign partners to roles in order to reduce time in system for a customer.

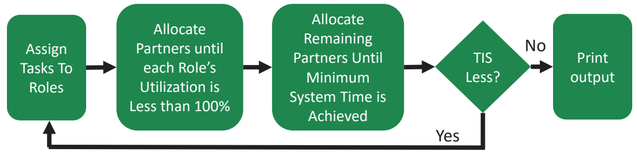
Process

Pull Data from Starbucks Database

- Store number
- Time of Day (1/2 hour increments)
- Product mix (Blended, Brew, Espresso, Warm Food, Ambient Food, Tea)
- Order Channel



Algorithm Flow Chart



Notes about algorithm

- Service times for stations weighted off of inter-arrival rates to that station

$$S_{Support} = \left(\frac{\lambda_{Tea}}{\lambda_{Tea}+\lambda_{Brew}}\right)\cdot(S_{Tea}) + \left(\frac{\lambda_{Brew}}{\lambda_{Tea}+\lambda_{Brew}}\right)\cdot(S_{Brew})$$

- Time in system is calculated from the initialized assignment vector with 1 partner on each role
- Time in system has to be at least two seconds lower in order to exit loop
- Roles: POS, BAR, SUPP1, SUPP2
- Tasks: Process Order, Tea, Brew, Espresso, Blended, Ambient Food, Warm Food

Example of Output (8 person play @ 10:30 on a Sunday)

Role: POS
Tasks Assigned: Process Order, Tea
Partners Assigned: 2

Role: BAR
Tasks Assigned: Espresso
Partners Assigned: 3

Role: SUPP1
Tasks Assigned: Warmed Food, Blended
Partners Assigned: 2

Role: SUPP2
Tasks Assigned: Brew, Ambient Food
Partners Assigned: 1

Average Time in System: 5 Minutes 29 Seconds
Average Partner Utilization: 82%

Validation Platform

Objective

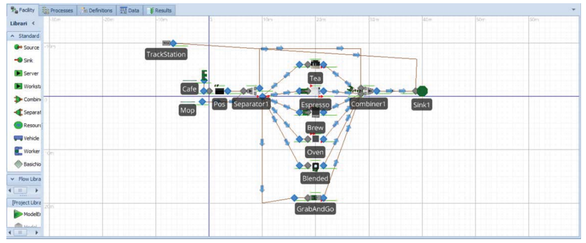
Create a model of a Starbucks store to test different assignment plays and verify/validate the output of the assignment algorithm.

*Simulation model is constructed based on actual store data including number of arrivals per half an hour, items ordered per half an hour and each station's average working time.

Simio Model

Objects, Stations, and Workers

- 7 working stations including 1 POS and 6 order prep stations (Espresso, Brew, Oven, Tea, Blended, and GrabAndGo)
- 4 types of workers: POS, BAR, FOOD, CUST SUPP



Assumptions

- Travel distances and speeds are assumed as 1 meter between each station and 1.4 meters per second.
- No defective products and rework process.
- The workers at oven stations are allowed to help with other stations while the food is warming.
- The processing time for Grab-And-Go station is zero.
- Exclude outliers such as extreme huge or small quantity orders.

Data

- Same data as assignment algorithm

Model Logic

- OptQuest Plugin
- Experiments are conducted based on any possible combination for number of workers at each station within certain range.
- After running all scenarios, the best can be chosen which has the lowest value of time in the system.
- The utilization of each type of worker is calculated by its busy time divided by the total time running the model.

Example of Output

Controls				Responses				Constraints
NumPosWorkers	NumBarWorkers	NumFoodWorkers	NumSupWorkers	POSUtilization	BARUtilization	FOODUtilization	EXPOUtilization	TimeInSyst...	TotalWorkers
2	3	2	2	33.4179	84.0309	72.2386	84.1785	1.21077	9
2	2	2	3	44.629	93.6569	83.9648	70.9482	1.75254	9
2	2	3	2	41.0589	93.2297	53.8362	84.0683	1.96671	9
2	3	3	1	43.377	86.734	59.8375	94.7097	2.04227	9
2	1	3	3	43.2544	95.9243	56.8974	67.0813	2.04244	9
3	3	2	1	30.5049	88.5119	82.0478	93.7799	2.12176	9
3	1	2	3	28.7971	95.8838	83.0479	73.2427	2.26295	9
3	2	2	2	29.446	94.0074	78.2741	84.3518	2.26923	9
2	3	1	3	43.183	90.4244	94.3326	70.1116	2.62315	9
3	1	3	2	28.5064	95.9644	58.0511	87.8964	2.75223	9
3	2	3	1	30.3958	94.0971	57.3686	94.9057	2.95514	9
1	3	2	3	77.5284	85.494	75.9479	62.0223	2.99285	9
1	2	3	3	79.0596	91.3943	52.2051	63.6051	3.08225	9
1	3	3	2	82.1736	83.829	51.2135	81.6272	3.20961	9
1	3	1	3	28.9033	88.3141	94.6567	86.9166	3.33269	9

Model Verification

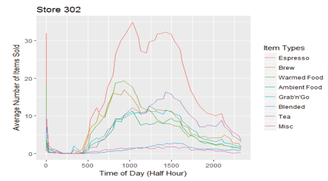
Comparing Output of Algorithm and Simio Model

Six different scenarios were run and tested for both the algorithm and the Simio model

- Number of partners: 5-7
- Store number: 302
- Play format: {POS, BAR, FOOD, SUPP}
- Play format = {POS, BAR, FOOD, SUPP}
- TIS = customer time-in-systm

Comparison Table

Time	Simio Play	Algorithm Play	Simio TIS	Algorithm TIS
8:30 am	{ 2, 2, 1, 2}	{3, 2, 0, 2 }	2 min 49 sec	2 min 28 sec
10:30 am	{1, 2, 1, 3}	{3, 2, 0, 2}	3 min 1 sec	2 min 37 sec
12:30 pm	{1, 2, 1, 2}	{2, 2, 0, 2}	2 min 56 sec	2 min 22 sec
2:30 pm	{1, 2, 1, 2}	{3, 2, 0, 1}	2 min 58 sec	2 min 8 sec
4:30 pm	{1, 2, 1, 1}	{2, 2, 0, 1}	3 min 00 sec	2 min 14 sec
5:30 pm	{1, 2, 1, 1}	{2, 2, 0, 1}	3 min 15 sec	2 min 28 sec



In-store Verification

Went in and observed store 302 (U village) from 10:30am - 11:00am on a Thursday

- Observed customer TIS: 5 min 24 sec; {1, 1, 1, 2}
- Simio model output: 4 min 40 sec; {1, 1, 1, 2}
- Algorithm output: 4 min 1 sec; {1, 2, 0, 2}

Results

Findings

- Changing the number of workers has a significant affect on TIS
  - From 9 to 7 partners the customer TIS will increase by 110%
- Current tool output vs. new tool output
  - Example: Current TIS = 4.12 min; New TIS = 3.05 min
  - Customer TIS reduced by 26%

Optimal Allocation & Average Waiting Time

ROLE	9	8	7	6
POS	2	2	1	1
BAR	2	2	1	1
FOOD	2	2	2	2
SUP	3	2	3	2
Waiting Time	1.75	2.32	3.68	4.82

Comparison of Utilizations by Total Number of Partners

ROLE	9	8	7	6
POS	44.6	45.9	77.4	80.7
BAR	93.7	93.6	94.1	94.1
FOOD	84	78.4	72.9	72.6
SUP	71.2	88.3	66.5	90.2

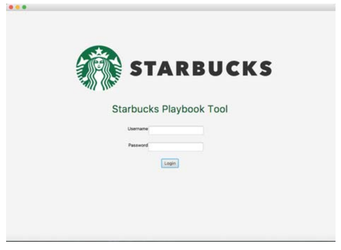
- Changing the number of workers has significant affect on POS role
  - From 2 to 1 POS partner, utilization increases 74%
- Target utilization value ranges from 70% to 80%
  - BAR role typically experiences high utilization due to high product flow

Impact on Starbucks

- Developed the backend of a staffing tool that level-loads tasks across partners and minimizes average time-in-system
  - Dynamic: treats every store uniquely
  - Effective: Outputs time-in-system and utilization values to help with manager's decision making process
- Economic Impact of Current Tool vs. New Tool (6 employee play)
  - Current tool TIS = 4.12 min; New Tool TIS = 3.05 min
  - Assuming avg. customer transaction = $5.50
  - Inter-arrival rate: 49 customers per ½ hour
  - Potential to increase sales by approx. $440 per day

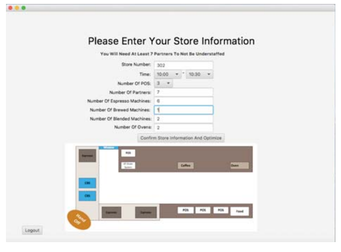
Concept Design of Deployment Tool

1



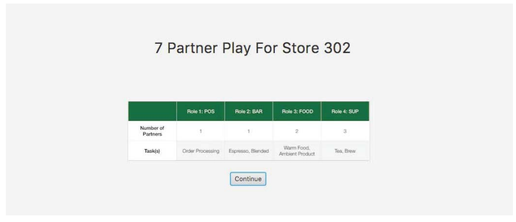
Login

2



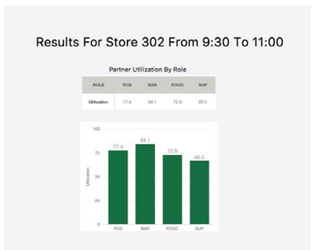
Input Data

3



Allocation

4



Graphs