OPTIMIZED RESOURCE ALLOCATION FOR A TWO SIDED MARKET

Leena Venugopal, Subramaniam Ganesh Kumar & Subramanian Vellaiyan

STEVENS INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY
Business Intelligence & Analytics

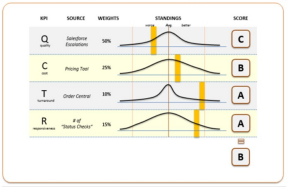
Introduction

- Client is a multi-national global information services organization specializing in corporate legal services.
- Client currently has a customer-vendor management system that allocates the jobs requested by the customers to the vendors, which involves a lot of paperwork.
- The client wants to automate this customer-vendor management system. A platform was created, which can automate the job allocation process by optimizing quality, turnaround, responsiveness and cost of the job.
- This platform can highly reduce the hassle of paperwork involved, the wait time for customers, and the money spent.

Model Design

- Linear programming model was used to allocate jobs optimally based on the scores.
- A basic static linear programming model was formed in excel and using Gurobi optimizer it was extended for dynamic data.

Optimizer Methodology



Allocation of 20 jobs to a pool of 6 vendors based on vendor scores and job allocation time

Assumptions

- Forced Ranking method is used for job allocation - Top 4 Vendors.
- When 4 vendors are not available the jobs are allocated to the available vendors.
- Vendors read from the input file are checked-in and ready for job allocation.
- When the maximum score is less than 85 the model waits for 20 minutes for the input to be read.

Inputs				
Vendors	V1	V2	V3	V4
Quality	50	45	45	40
Cost	25	20	20	20
TAT	15	15	10	10
Responsiv	10	10	10	10
Score	100	90	85	80

Number of vendors available for allocation	% of the jobs allocated to			
	1st ranked vendor	2nd ranked vendor	3rd ranked vendor	4th ranked vendor
4	60	20	15	5
3	70	20	10	
2	80	20		
1	100			

Methodology

- Unique score creation (Vendor score * job * job priority _score) for maximizing the allocation.
- Jobs are prioritized based on the in time : First in First out approach.
- Different scenarios created to stress test the model.

Constraints

- Vendors can have multiple jobs but one job can have only one vendor
- Allocation constraints were implemented based on forced ranking scenarios

m = Number of Vendors
n = Number of Jobs

J1…Jn = Jobs
V1…Vn = Vendors

Allocation constraint for each vendor

V1(J1+J2+....+Jm) <= m*0.6 if n=4
V2(J1+J2+....+Jm) <= m*0.2 if n=4
V3(J1+J2+....+Jm) <= m*0.15 if n=4
V4(J1+J2+....+Jm) <= m*0.05 if n=4
V1(J1+J2+....+Jm) <= m*0.7 if n=3
V1(J1+J2+....+Jm) <= m*0.2 if n=3
V1(J1+J2+....+Jm) <= m*0.1 if n=3
V1(J1+J2+....+Jm) <= m*0.8 if n=2
V1(J1+J2+....+Jm) <= m*0.2 if n=2
V1(J1+J2+....+Jm) <= m*1 if n=1

Sum of vendors per job should be equal to one.

J1(V1+V2+...+Vn) = 1
J2(V1+V2+...+Vn) = 1
...
...
...
Jm(V1+V2+...+Vn) = 1

Results

Data

The model needs two input files:

- Job Id and their arrival time
- Scores for individual vendors and respective vendor ID.

Output

The ouput is stored in a text file has the following details Vendor ID, allocated number of jobs for the respective vendor and average vendor score

Job_Id	Arrival_Time
3106	1148
2024	1411
4276	1104
5628	1303
1149	1442
3173	1434
2675	1043
3773	1010
7996	1125
6541	1148
7931	1224
3479	1242
7564	1205
7431	1214
7098	1340
7234	1227
9135	1419
6168	1248
8472	1335
6968	1335

Job_Id	Arrival_Time
3106	1210
2024	906
4276	1600
5628	1500
1149	913
3173	1125
2675	1023
3773	930
7996	1200
6541	933
7931	934
3479	940
7564	1320
7431	1450
7098	955
7234	1010
9135	935
6168	1020
8472	1034
6968	1123

Scenario 1

Number of Vendors: 6
Number of Jobs: 20
Allocation by the model

Vendor_ID	Count
738	3.0
109	12.0
1065	1.0
185	4.0

Average vendor score: 93.783

738	0.928571429
11797	0.886363636
1000	0.905172414
109	0.942307692
1065	0.927197802
185	0.93404908

Scenario 2

Number of Vendors: 3
Number of Jobs: 20
Allocation by the model

Vendor_ID	Count
738	2.0
11797	4.0
1000	14.0

Average vendor score: 84.149

738	0.78571429
11797	0.786363636
1000	0.865172414

Scenario 3

Number of Vendors: 2
Number of Jobs: 20
Allocation by the model

Vendor_ID	Count
738	16.0
11797	4.0

Average vendor score: 92.184

738	0.95571429
11797	0.786363636

Scenario 4

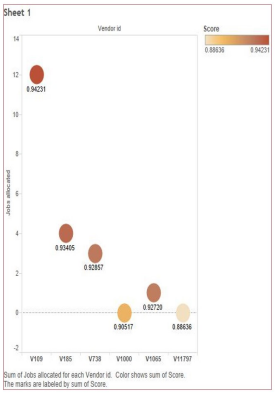
Number ofVendors: 1
Number of Jobs : 20
Allocation by the model

Vendor_ID	Count
738	20.0

Average vendor score: 95.57

738	0.95571429

Jobs-Vendors	Arrival Time	Job Priority Score	V1	V2	V3	V4	Job Allocated	Maximum Job Allocatio
J1	930	770	1	0	0	0	1	1
J2	935	765	1	0	0	0	1	1
J3	940	760	1	0	0	0	1	1
J4	945	755	1	0	0	0	1	1
J5	950	750	1	0	0	0	1	1
J6	955	745	1	0	0	0	1	1
J7	1000	700	1	0	0	0	1	1
J8	1005	695	1	0	0	0	1	1
J9	1010	690	1	0	0	0	1	1
J10	1015	685	1	0	0	0	1	1
J11	1020	680	1	0	0	0	1	1
J12	1025	675	1	0	0	0	1	1
J13	1030	670	0	1	0	0	1	1
J14	1035	665	0	1	0	0	1	1
J15	1040	660	0	1	0	0	1	1
J16	1045	655	0	1	0	0	1	1
J17	1050	650	0	0	1	0	1	1
J18	1055	645	0	0	1	0	1	1
J19	1100	600	0	0	1	0	1	1
J20	1105	595	0	0	0	0	1	1
Jobs per Vendor			12	4	3	1		
Jobs per Vendor Maximum Allocation			12	4	3	1		
Job Priority Score* Binary (J,V) * Vendor			867000	238500	169875	47600	1314175	



Future Enhancements

- Jobs are currently allocated on the basis of the arrival time(FIFO). Each job can be prioritized for allocation by assigning weights for the jobs.
- The score of each vendor can be dynamically calculated after the allocation and updated back to the vendor input data .
- The allocation is currently at the overall score level. Allocation can be prioritized for Quality, cost , turnaround and responsiveness at the individual level incase of tie in the scores.