WonderMarket
Section A – Internal Report

Kenton Lam

MATH3202 Assignment 3
Due 27/05/2019 1:00 pm

Abstract

Our long-time client WonderMarket has approached regarding their foray into the refridgerator space. Competition in the area is fierce and they need us to optimise their fridge logistics. Given information about their costs and requirements, we created and optimmised a dynamic programming model using Python. This report describes the model and its solution.

Model Definition

WonderMarket wishes to start selling fridges. For the time being, they are focusing on 3 bespoke fridge options—named "Alaska", "Elsa" and "Lumi". Over a 4 week trial period, they need to decide how many fridges of each type to purchase each week.

After communication 9, we developed a detailed model to mathematically represent their objectives. This included considering each possible demand scenario to maximise expected profit.

Sets

Let the 3 types of fridges be numbered as 0 is Alaska, 1 is Elsa and 2 is Lumi. Denote this set as F.

Data

The following data has been provided to us by WonderMarket.

Profit_f	profit made by selling one of fridge f.
$\text{Expected}_{f,n}$	expected number of fridge f sold if n are displayed (comm 1). ($0 \leq n \leq 4$).
$\text{Demands}_{f,n}$	probability that n units of fridge f will be sold (comm 2). ($1 \leq n \leq 6$).
StoreCost	cost of storing one fridge for one week (comm 2).
FridgesPerTruck	maximum fridges transported by one truck (comm 3).
TruckCost	cost of one truck (comm 3).
MaxTrucks	maximum number of trucks per week (comm 3).
MaxStore	maximum number of each fridge type handled per week (comm 3).

Stages

Our stage will be the number of weeks elapsed since the trial began. That is,

$$t = 0, 1, 2, 3, 4.$$

Because this is only a 4 week trial, we stop once 4 weeks have elapsed.

State

This is the state carried from one week to the next. Specifically, this is number of each type of fridge they have in storage. Denote this as a 3-dimensional vector $\mathbf{s}$, where each component is

$$s_f = \text{number of fridge } f \text{ stored.}$$

Actions

At each time step, WonderMarket can order up to 14 fridges, distributed in any way among the 3 types. These are the actions they can take. Thus, the action space is given by the set

$$\text{Actions} = \{(i, j, k) \mid i, j, k \in \mathbb{Z}_{\geq 0}, i + j + k \leq \text{FridgesPerTruck} \times \text{MaxTrucks}\}.$$

Let $\mathbf{a} \in \text{Actions}$ denote one specific action.

More Notation

For a vector $\mathbf{v}$, let v_i be the i-th component of the vector. We write the element-wise minimum of two n-dimensional vectors as

$$\text{emin}(\mathbf{v}, \mathbf{u}) = (\min\{v_1, u_1\}, \min\{v_2, u_2\}, \ldots, \min\{v_n, u_n\})$$

Note that Profit is a 3-dimensional vector.

In our value function, we need to consider all possible permutations of demands of each fridge type. We write the set of demand permutations as

$$\text{DemandPerms} = \{1, 2, \ldots, 6\}^3$$

and denote the probability of each permutation $\mathbf{d} \in \text{DemandPerms}$ as

$$\text{Prob}_{\mathbf{d}} = \prod_{f \in F} \text{Demands}_{f, d_f}.$$

Value Function

The value function represents the maximum expected profit given a starting time and state. It can be expressed as $V_4(\mathbf{s}) = 0$, then for $t < 4$,

$$\begin{aligned}
V_t(\mathbf{s}) = \max_{\substack{\mathbf{a} \in \text{Actions} \\ \max(\mathbf{s}+\mathbf{a}) \leq \text{MaxStore}}} \Bigg\{ & - \text{StoreCost} \times \sum_{f \in F} (s_f + a_f) \\
& - \text{TruckCost} \times \left\lceil \frac{\sum_{f \in F} a_f}{\text{FridgesPerTruck}} \right\rceil \\
& + \sum_{\mathbf{d} \in \text{DemandPerms}} \text{Prob}_{\mathbf{d}} \times \Big(\text{Profit} \cdot \text{emin}(\mathbf{d}, \mathbf{s}+\mathbf{a}) \\
& \qquad + V_{t+1}(\mathbf{s}+\mathbf{a} - \text{emin}(\mathbf{d}, \mathbf{s}+\mathbf{a})) \Big) \Bigg\}
\end{aligned}$$

Derived Data

To simplify later calculations, we derived some data from the data given above.

$\text{SurgeMultiplier}_{u,s}$	surge demand at store s during surge u, divided by regular demand at store s. $\text{SurgeMultiplier}_{u,s} = \text{SurgeDemand}_{u,s}/\text{Demand}_s$
NormalWeeks	number of weeks in the year which have no surge scenario. $\text{NormalWeeks} = 52 - \sum_{u \in U} \text{SurgeWeeks}_u$

Variables

The following variables were used in the Gurobi model.

B_d	binary variables for whether DC d is active. (1 means if d is new then d is built; if d already exists then d is not closed).
P_d	integer number of part-time teams employed at DC d year-round.
F_d	integer number of full-time teams employed at DC d year-round.
$C_{u,d}$	integer number of casual employeed employed at DC d during surge u. (each casual employee is only employed for the duration of the surge).
$A_{d,s}$	binary variable for whether DC d delivers to store s.
$X_{d,s}$	integer truckloads to be sent from DC d to store s during normal demand.
$Y_{d,s,u}$	integer truckloads to be sent from DC d to store s during surge scenario u. (X and Y will be 0 or exactly match demand, so will be integers.)

Objective

This calculates the total yearly cost, considering transport and labour costs. The objective is to *minimise* the following function.

$$\begin{aligned}
&\text{(normal transport cost)} && \sum_{s\in S}\sum_{d\in D} \text{NormalWeeks}\cdot \text{Cost}_{d,s}X_{d,s} \\
&\text{(surge transport costs)} && +\sum_{u\in U}\sum_{s\in S}\sum_{d\in D} \text{SurgeWeeks}_u \text{Cost}_{d,s}Y_{d,s,u} \\
&\text{(full/part-time labour costs)} && +52\sum_{d\in D}\text{PTCost}\,P_d + 52\sum_{d\in D}\text{FTCost}\,F_d \\
&\text{(casual labour costs)} && +\sum_{u\in U}\sum_{d\in D}\text{CasualCost}\,\text{SurgeWeeks}_u C_{u,d}
\end{aligned}$$

Constraints

First, we have the constraint that all variables are non-negative and certain variables are integers or binary. For all $d \in D,\ s \in S,\ u \in U$, we have

$$\begin{aligned}
B_d, P_d, F_d, C_{u,d}, A_{d,s}, X_{d,s}, Y_{d,s,u} &\geq 0 \\
B_d, A_{d,s} &\in \{0,1\} \\
P_d, F_d, C_{u,d} &\in \mathbb{Z}
\end{aligned}$$

We link X and Y variables via A and the known demand at each store. This ensures each store receives all its supplies from one DC.

$$X_{d,s} = \text{Demand}_s A_{d,s} \qquad \forall\ d\in D,\ s\in S$$

We link X and Y via the surge multipliers data. This ensures for each store, it receives deliveries from the same DCs in each scenario as during normal demand.

$$Y_{d,s,u} = \text{SurgeMultiplier}_{u,s}X_{d,s} \qquad \forall\ d\in D,\ s\in S,\ u\in U$$

We ensure the solution is valid for normal demand considering truckloads and capacities. Note that the northside capacity limit has been removed.

$$\begin{aligned}
\sum_{d\in D} X_{d,s} &\geq \text{Demand}_s && \forall\ s\in S \\
\sum_{s\in S} X_{d,s} &\leq \text{Capacity}_d && \forall\ d\in D
\end{aligned}$$

Ensuring at most 2 new DCs are built and there are 4 DCs in total (3 or 2 old).

$$\begin{aligned}
\sum_{d\in \text{NewDCs}} B_d &\leq 2 \\
\sum_{d\in D} B_d &= 4
\end{aligned}$$

Ensuring enough teams are employed to meet normal demand at each DC.

$$\sum_{s\in S} X_{d,s} \leq \text{FTCapacity}\,F_d + \text{PTCapacity}\,P_d \qquad \forall\ d\in D$$

We ensure our assignments can scale up to each surge scenario while remaining feasible with the given constraints. For each $u \in U$,

$$\sum_{d \in D} Y_{d,s,u} \geq \text{SurgeDemand}_{u,s} \qquad \forall\ s \in S$$

$$\sum_{s \in S} Y_{d,s,u} \leq \text{Capacity}_{d} \qquad \forall\ d \in D$$

$$\sum_{s \in S} Y_{d,s,u} \leq \text{FTCapacity}\, F_d + \text{PTCapacity}\, P_d + C_{u,d} \qquad \forall\ d \in D$$

Solution

Solving this MILP model in Gurobi returns the following solution. This would cost WonderMarket $12576018.00 each year ($241846.50 per week).

Store Assignments

Store	DC0	DC1	DC2	DC3	DC4	DC5	DC6
S0				100.0%			
S1				100.0%			
S2				100.0%			
S3						100.0%	
S4				100.0%			
S5		100.0%					
S6		100.0%					
S7			100.0%				
S8			100.0%				
S9				100.0%			

Distribution Centres

WonderMarket should close DC0 and build DC3 and DC5. DC1 and DC2 should remain open.

Labour

WonderMarket should hire the following teams for the whole year.

DC	Part-time	Full-time
DC1	0	2
DC2	0	2
DC3	0	8
DC5	0	2

Additionally, they should hire the following casual staff for certain surges.

Surge	DC	Casual
Surge 0	DC5	11
Surge 1	DC2	20
Surge 3	DC2	20
Surge 4	DC1	42
Surge 4	DC3	2