Decision Making under Uncertainty

Yasemin Merzifonluoglu

Assignment I

Iwan van Es	2128762
Rory Kolster	2070539
Quirijn Schevenhoven	1274478
Maxim de Vries	2079287

GROUP 20

Contents

1 Introduction and Problem Data

In this section we will give a short description of the mathematical model of the Selective Newsvendor Problem (SNP), as defined by Fleuren et al. (2024). The SNP extends the classic newsvendor problem (where the procurement quantity Q is determined) by having the firm decide whether or not to serve each customer. First we present the involved parameters. Secondly, we discuss the decision variables. Finally, we present the objective function and the constraints of the model.

Sets and indices

I: The set of customers, i.e., $I = \{1, ..., I\}$

i, j or k: The customer indices, i.e., $i, j, k \in I$

Decision Variables

Q: Procurement quantity for base product

$$x_i: \begin{cases} 1, & \text{if customer } i \text{ is selected,} \\ 0, & \text{otherwise} \end{cases}$$

Parameters

c: Unit ordering cost for base product

p: Random spot market price for base product

$F^P(\cdot)$ $(f^P(\cdot))$: CDF (PDF) of spot market price

μ^P: Expected spot market price

σ^P: Standard deviation of spot market price

h: Unit inventory holding cost for residual base product

D_i: Random demand from customer i

$F_i(\cdot)$ $(f_i(\cdot))$: CDF (PDF) of demand from customer i

μ_i: Expected demand from customer i

σ_i: Standard deviation of demand from customer i

Λ_i: Fixed cost for selecting customer i

c_i: Customer-specific customization cost for customer i

r'_i: Unit price for customer i

$r_i = r'_i - c_i$: Unit revenue after customization for customer i

s_i: Loss of goodwill cost for not satisfying customer i

$\tilde{s}_i = r_i + s_i$: Stockout cost due to sales lost for the customer i

$g_i = \tilde{s}_i - \tilde{s}_{i+1}$,
$g_N = \tilde{s}_N + h$: Incremental (echelon) shortage cost for customer i relative to $i + 1$

Objective function

Let $D_x(j) := \sum_{i=1}^{j} D_i x_i$ with mean $\mu_x(j) = \sum_{i=1}^{j} \mu_i x_i$ and variance $\sigma_x^2(j) = \sum_{i=1}^{j} \sigma_i^2 x_i$, where $x \in \{0,1\}^N$. According to Fleuren et al. (2024) by eq(8) in section 3.2, the expected profit objective function can be defined as

$$G(x, Q) = \sum_{j=1}^{N} ((r_j + h)\mu_j - \Lambda_j)x_j - \sum_{j=1}^{N} g_j \ell_j(x, Q) - (c + h)Q, \tag{1}$$

where $\ell_j(x, Q) := E[(D_x(j) - Q)^+]$ denotes the loss function.

The objective function can be dissected into three parts. We describe each part in the order they appear in the objective function. The first term can be understood as the net revenue for serving the customers. The second term is the total expected loss (based on the stock-out cost due to lost sales) for all selected customers. The final term is the total cost of the ordering and holding the procured base product. The objective is to maximize $G(x, Q)$, that is, maximize the total expected profit. As firms need to be profitable to stay competitive, we arrive to the maximization problem as shown in equation (1).

Constraints

The constraints to which the decision variables have to adhere are intuitive. Constraint (2) ensures that a customer x_i can be selected or not. That is, x_i is a binary variable. Equation (3) guarantees that the procurement quantity non-negative.

$$x_i \in \{0,1\}^N \tag{2}$$

$$Q \geq 0 \tag{3}$$

In the following section, we estimate the expected profit for the SNP (in a scenario where customer demands are independent and there is no spot market) using Monte Carlo simulation. Section 3 discusses solving this problem using the Sample Average Approximation (SAA). Finally, Section 4 solves the SNP considering risk aversity.

We now give the problem instance with which we will be performing simulations on, for the rest of the project. We set the number of customers $N = 10$ and generated the following parameters prescribed in Table 2 of Fleuren et al. (2024). By setting the seed in our code, we are able to reuse the same parameters throughout. The following table contains the instance parameters.

Parameter	Values									
c	5.428									
h	1.474									
μ_i	26.03	21.64	11.88	18.66	19.58	13.19	24.69	12.27	17.82	20.33
σ_i	6.699	6.022	3.546	6.112	4.658	3.780	7.230	2.934	3.568	6.706
Λ_i	460.6	329.9	127.6	260.8	208.1	127.4	187.0	42.27	85.73	51.30
r_i	23.34	22.21	20.09	20.00	17.68	16.27	14.51	11.84	10.90	10
s_i	26.72	25.54	22.99	22.68	20.04	17.77	16.48	14.02	11.97	10

Table 1: Instance Parameters

2 Monte Carlo Simulation

The objective of this section is to estimate the expected profit for the SNP under a scenario where customer demands are independent and there is no spot market. It consists of two main parts. Part one considers a specific solution with a pre-defined customer selection. Part two takes this customer selection as an initial solution and improves upon it by utilizing the DERU-based heuristic (from Section 3.2.1 Fleuren et al. (2024)). The initial and improved solutions are then compared. Finally, the estimate of the *improved solution's* expected profit is corrected using control variates.

2.1 Monte Carlo Simulation of the Initial Solution

Consider a specific solution where $x_1 = x_2 = \cdots = x_5 = 1$ and $x_6 = \cdots = x_{10} = 0$. This means that only the first five customers are selected. The total order quantity Q is defined as $Q = \sum_{i=1}^{5} \mu_i$, where μ_i represents the mean demand for each of these customers. This scenario is referred to as the *initial solution*, and this section estimates the expected profit under these conditions. In our case we ran 100000 iterations in our Monte Carlo simulation to approximate a large enough sample size, such that the Law of Large Numbers can be approximated to obtain meaningful conclusions. For each run, the next steps are followed:

1. Demands and the spot market price are generated using the means of the problem instance found in Table 1. Any negative values are adjusted to 0.
2. The revenue, cost, and total profit for the given solution are calculated.

The random numbers that are generated here are done using the Scipy.stats package within Python. We could have generated random numbers via the methods learnt in class with a $Unif(0,1)$ distribution, but as we perceived this not to be the aim of the assignment, we used the built-in distributions from Scipy.

To determine the profit, we use expression 7 from section 3.2 in the article. First let $D_x(j) := \sum_{i=1}^{j} D_i x_i$ with mean $\mu_x(j) = \sum_{i=1}^{j} \mu_i x_i$ and variance $\sigma_x^2(j) = \sum_{i=1}^{j} \sigma_i^2 x_i$, where $x_i \in \{0,1\}^n$. Then for given demand realizations, the expression for profit is as follows.

$$\sum_{j=1}^{N}(r_j D_j - \Lambda_j)x_j - \sum_{j=1}^{N} g_j(D_x(j) - Q)^+ - (c+h)Q + hD_x(N), \tag{4}$$

where r_j is the unit revenue after customization, Λ_j is the fixed cost for selecting customer j, g_j is the incremental (echelon) shortage cost, c is the unit ordering cost, and h is the unit inventory holding cost.
The profits, calculated using the expression above, is determined for each iteration of the Monte Carlo simulation. The sample mean of these profits is then computed, yielding an estimated expected profit of -66.4 with a confidence interval of $(-67.36, -65.44)$ for a two-tailed test at a significance level of $\alpha = 0.05$. Below we present the histogram representing the profits in each iteration of the Monte Carlo simulation.

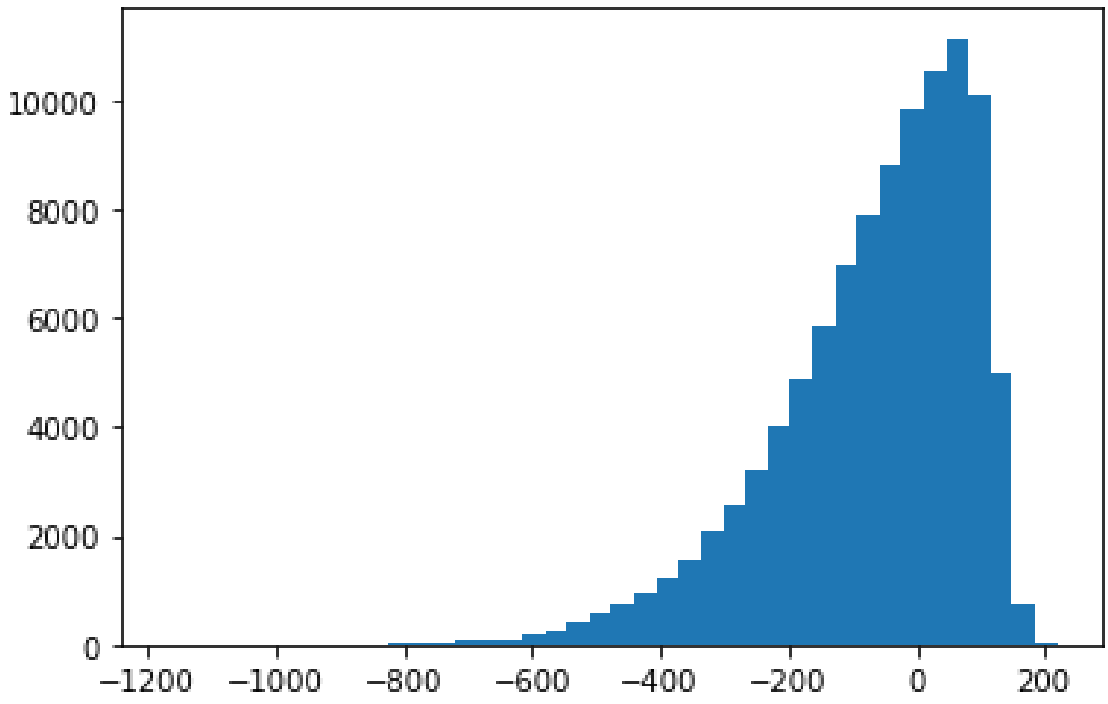

Figure 1: Profit Histogram for initial solution

We now compare our Monte Carlo result to the expected profit obtained by using equation 10 from the article. Define

$$z_Q^j := \frac{Q - \sum_{i=1}^{j} \mu_i x_i}{\sqrt{\sum_{i=1}^{j} \sigma_i^2 x_i}}, \tag{5}$$

for $j = 1, \ldots, N$. We have the loss function $l_j(x, Q) := \mathbb{E}\left[(D_x(j) - Q)^+\right]$, which then becomes $L\left(z_Q^j\right)\sigma_x(j)$, where $L(z)$ is the standard normal loss function. Define, also, $\rho_j := (r_j - c)\mu_j - \Lambda_j$. This allows us to write the expected profit, equation 10 from the article, as

$$G(x, Q) = \sum_{j=1}^{N} \rho_j x_j - \sum_{j=1}^{N} g_j L\left(z_Q^J\right) \sqrt{\sum_{i=1}^{j} \sigma_i^2 x_i} - (c+h) z_Q^N \sqrt{\sum_{j=1}^{N} \sigma_j^2 x_j}. \tag{6}$$

Using the above formula, we obtained -65.8 as expected revenue for this initial solution, so comparing this to the -66.4 revenue we obtained, it seems verifiable.

2.2 Monte Carlo Simulation of the Improved Solution

In this section, the objective is to estimate the expected profit for an *improved solution.* First, the DERU-based heuristic (Fleuren et al. (2024)) is implemented to suggest an improved solution. Next, the simulation steps as outlined in Section 2.1 are followed and a similar analysis is provided.

We first explain the DERU-based heuristic, as given in the article. When the incremental shortage cost g_j for customers 1 to $N-1$ is zero, the problem simplifies to the SNP problem defined in Taaffe, Geunes, and Romeijn (2008), where it is shown that ordering the customers based on the Decreasing Expected Revenue to Uncertainty (DERU) ratio, i.e. $\frac{\rho_j}{\sigma_j^2}$, yields an optimal solution. However, for at least one customer with $g_j \neq 0$, the DERU statistic is no longer guaranteed to find an optimal solution. Ranking customers based on the DERU statistic may not be preferable due to customer-specific stockout costs may be significant.
Fleuren et al. (2024) propose using this DERU-heuristic as follows.

- They order customers $j = 2, \ldots, N-1$ according to the DERU ratio, and reindex them. Then they consider all candidate solutions: the first k customers for $0 \leq k \leq N-2$, of which there are $N-1$ possibilities. They then consider each possible solution for the first and last customers, which were not ranked. These 4 scenarios are combined with the $N-1$ choices, creating a customer selection vector of length $4(N-1)$.

- Then for each candidate x^p, they find the optimal order quantity, Q. They do this by performing a bisection search on the following variant of equation 4 from Fleuren et al. (2024):

$$\sum_{j=1}^{N} g_j(1 - F^j(Q)) - c - h = 0, \tag{7}$$

  where F^j is the cdf of $\sim \mathcal{N}\left(\sum_{i=1}^{j} \mu_i x_i, \sum_{i=1}^{j} \sigma_i^2 x_i\right)$.

- Then for each candidate x^p and Q^p use the Equation 6 (equation 10 from the article) to find the expected profit.

- Finally select the candidate x^p and Q^p which yield the highest expected profit.

Then with this *improved solution*, we apply the same Monte Carlo simulation as done above for the initial solution.
It turns out that for our instance, the chosen *improved solution* was selecting all customers except the first, i.e. $x_1 = 0, x_2 = 1, x_3 = 1 \ldots, x_10 = 1$. The associated order quantity found was $Q = 167.5$. Then doing an Monte Carlo simulation yielded an estimated expected profit of 145.4 and a $\alpha = 0.05$ two-tailed confidence interval of $(144.2, 146.7)$. The following figure shows the histogram plot of the simulations.

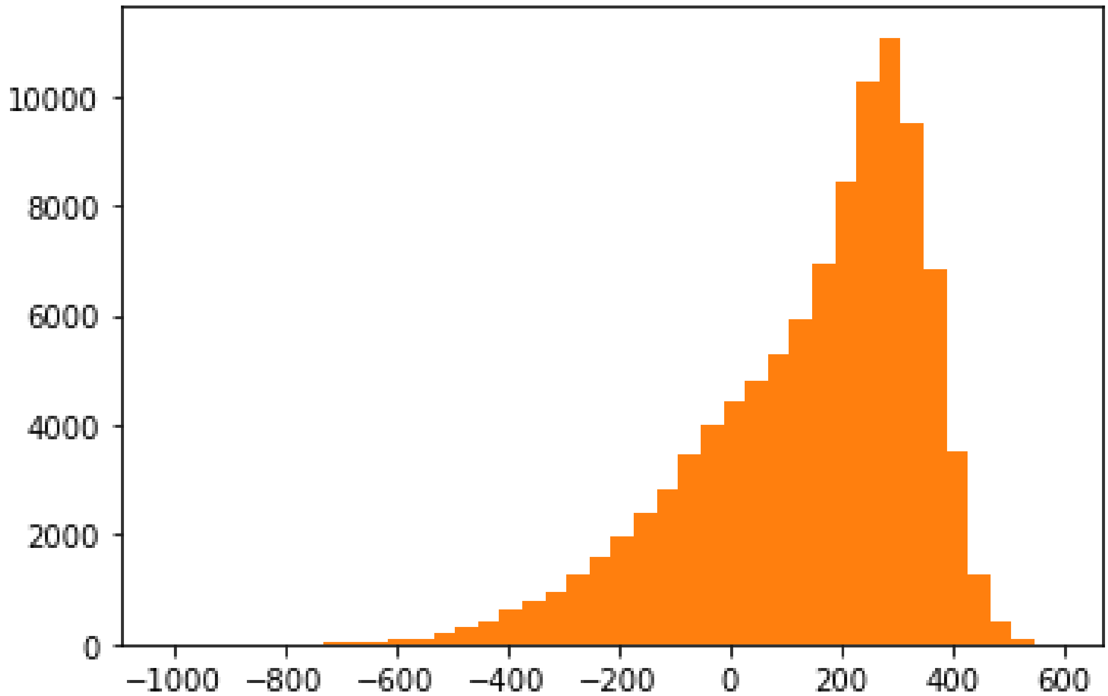

Figure 2: Profit Histogram for improved solution

2.3 Comparison of the Solutions

We now compare the two solutions obtained in the above section. The initial solution only chooses the first 5 customers, while the improved solution for our instance choose all but the first customer. The order quantity for the initial solution was 97.8, and 167.5 for the improved. Hence, the improved solution chooses more customers to supply and also chooses a higher order quantity to satisfy this. We programmed it such that the same demands in the simulations were used in both solutions. The following plot shows both histograms for each of the solutions.

As seen, the improved solution is similar to the initial solution in shape, but shifted to the right. We calculated the difference confidence interval to be $(210.2, 213.4)$, with mean 211.8.

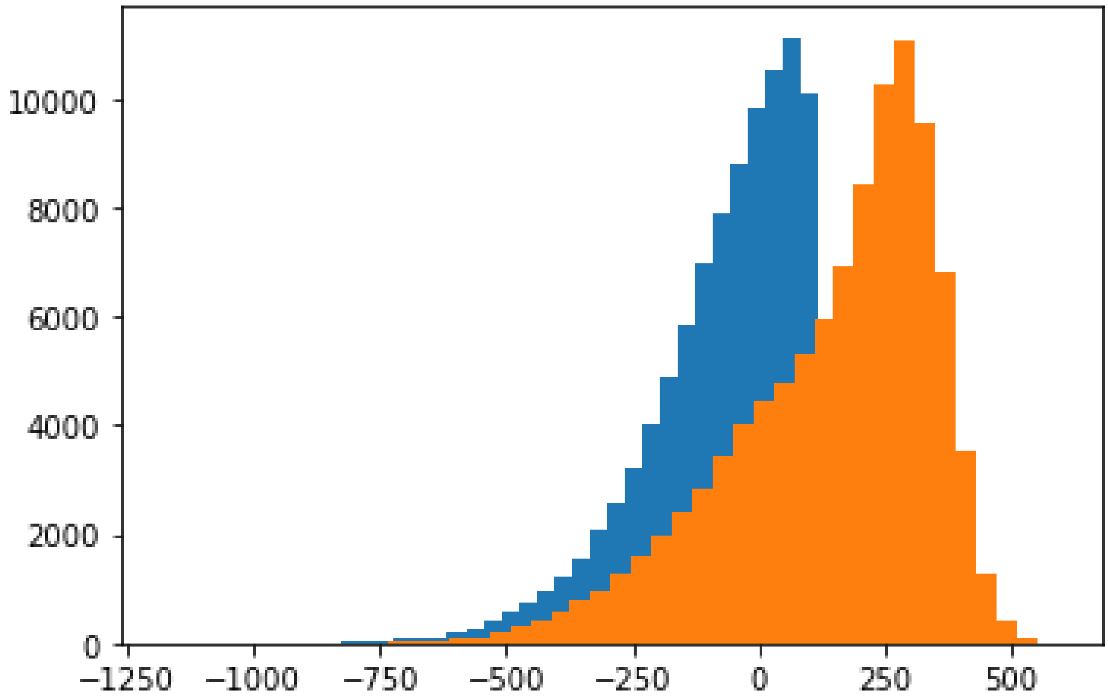

Figure 3: Profit Histogram comparing solutions

2.4 Variance Reduction

To reduce variance, we employ control variates. Our chosen control variate is the sum of demands of the customers in each specific run of the simulation, as we have the known mean being the sum of the expected demands. We then follow the formula given in the lecture slides:

$$X_c = X + c(Y - \mu_y)$$
$$c^* = -\frac{Cov(X,Y}{Var(Y)}$$

where X is the revenue, Y is the control variable, and c^* is the optimal scale factor that minimizes the variance. So c^* for us is the covariance of the revenue from the Monte Carlo simulation and the cumulative demand, divided by the variance of the cumulative demand.

Performing the Monte Carlo simulation again and applying the control variates yielded a new mean of 146.0 and 95% confidence interval of $(145.2, 146.8)$. So this is a slightly lower profit, but with a tighter confidence interval. The effectiveness of the variance reduction can also be seen in the variance between profits in each iteration of the Monte Carlo simulation. With variance reduction, the variance decreases from $40,807.8$ to $17,574.9$, resulting in approximately a 57% reduction in variance. We present the two histograms of the improved solution with and without the control variates in the figure below.

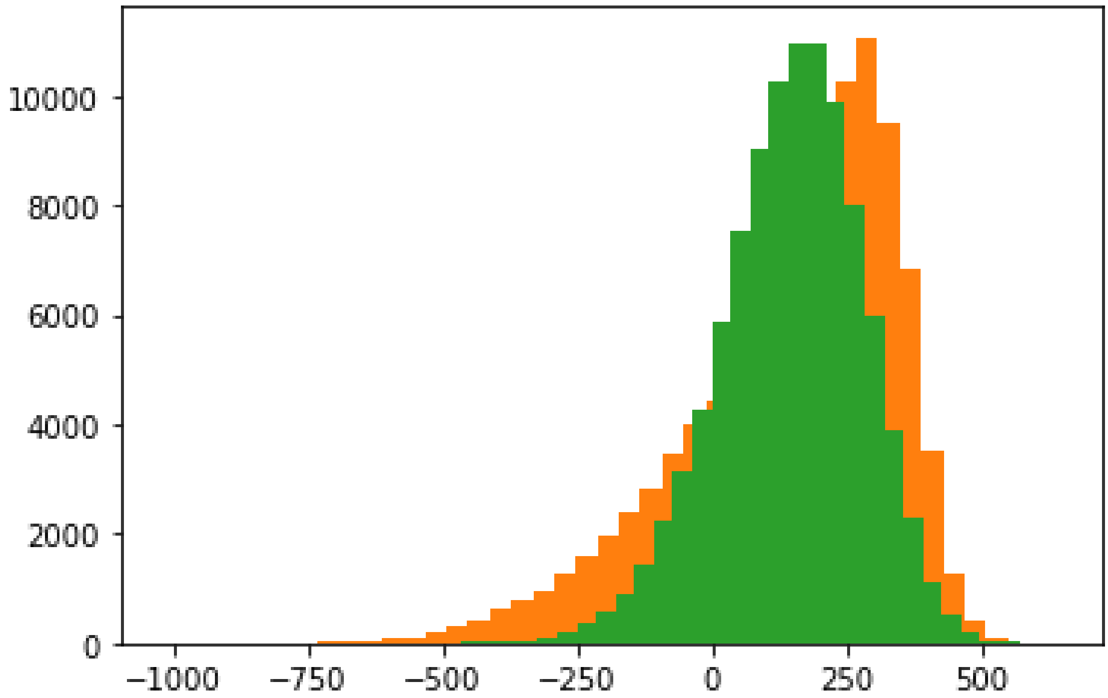

Figure 4: Profit Histogram comparing control variates

3 Stochasting Programming

3.1 Problem Formulation

We provide the two-stage problem formulation for stochastic programming, as given in Fleuren et al. (2024). We define the following decision variables. Let $x_i \in \{0,1\}$ be the customer selection for customer i. Let I_m represent the leftover inventory for scenario m. Let q_{im} be the allocated quantity from the procured stock Q that is allocated to customer demand i. Define l_{im} as the excess demand that is lost. For section 3.4 we define e_{im} as the demand that is satisfied from the spot market for order i in scenario m, while otherwise it is 0. We now have the optimization problem as follows:

$$
\begin{aligned}
\max \quad & \frac{1}{M}\sum_{m=1}^{M}\left[\sum_{i=1}^{N}(r_i D_{im} - \Lambda_i)x_i - cQ - hI_m - \sum_{i=1}^{N}(\tilde{s}_i l_{im} + p_m e_{im})\right] \\
\text{s.t.} \quad & q_{im} + l_{im} + e_{im} = D_{im}x_i && i = 1,\dots,N,\ m = 1,\dots,M, \\
& \sum_{i=1}^{N} q_{im} + I_m = Q && m = 1,\dots,M, \\
& I_m \geq 0,\ q_{im} \geq 0,\ l_{im} \geq 0, e_{im} \geq 0 && i = 1,\dots,N,\ m = 1,\dots,M, \\
& x_i \in \{0,1\} && i = 1,\dots,N, \\
& Q \geq 0.
\end{aligned}
$$

In the above, x and Q are both first-stage decision variables. The rest of the decision variables are second-stage.

3.2 SAA Algorithm

The idea of Sample Average Approximation (SAA) is to convert a stochastic optimization problem into a deterministic one by approximating the value using averages. We follow the following algorithm:

1. We generate M independent, identically distributed (IID) random samples of the N stochastic variables: $\xi^1, \ldots, \xi^N$. We do this by using the scipy.stats package in python, generating random variables from the multi-variate normal distribution. It is sufficient to take M relatively small.

2. We then compute the optimal selection vector x_i, procurement quantity Q, and objective value G(x,Q) for each iteration SAA problem.

3. Then from this vector of size M, we choose the SAA iteration with the maximum objective value (since we are maximising), saving the first-stage decision variables. We denote it as $\hat{\Omega}$.

4. We then determine the optimality gap by $gap(\hat{\Omega}) = g^*(\Omega) - g(\hat{\Omega})$, where $g(\hat{\Omega})$ is the objective value evaluated at the chosen customer selection vector, and $g^*(\Omega)$ is the optimal value of the true problem. Note that since we are with a maximization problem we have multiplied the usual definition for minimization by -1, and hence the upper bounds and lower bounds are switched as well. However, since $g^*(\Omega)$ is not known and $g(\hat{\Omega})$ is still random, we estimate an upper bound of $g^*(\Omega)$, and we also estimate $g(\hat{\Omega})$ via an lower bound.

   (i) Generate again a new IID random sample $\zeta^1, \ldots, \zeta^{N'}$ for a large sample N' ($N' > N$). The aforementioned method is used to generate the samples as in step one. This solves the second-stage problem, given the first-stage decision variables $\hat{\Omega}$ and the sample ζ^k. We compute the sample mean, $\hat{g}_{N'}(\hat{\Omega})$ and the sample standard deviation $\hat{\sigma}_{N'}(\hat{\Omega})$. This provides the lower bound for $g(\hat{\Omega})$ as $L_{N'} = \hat{g}_{N'}(\hat{\Omega}) - z_\alpha \hat{\sigma}_{N'}(\hat{\Omega})$, where z_α is the critical value for the standard normal distribution at significance level α.

   (ii) We reuse the sample from step 1, compute the sample mean $\overline{g_N^*}(\Omega)$ and sample standard deviation $\overline{\sigma_N}(\Omega)$. It can be shown that (for maximization) $g^*(\Omega) \leq \mathbb{E}[g_N(\Omega)]$, where $g_N(\Omega)$ is the results of one SAA iteration. Hence, the upper bound of $g^*(\Omega)$ is $U_{N,M} = \overline{g_N^*}(\Omega) + t_{\alpha,M-1}\overline{\sigma_N}(\Omega)$, where t_α is the critical value for the Student's t-distribution at significance level α, with $M-1$ degrees of freedom.

   (iii) Then together with the lower bound of $g(\hat{\Omega})$ and the upper bound of $g^*(\Omega)$, we the estimated optimality gap as $g\hat{a}p(\hat{\Omega}) = U_{N,M} - L_{N'}$.

3.3 SAA Analysis and Results

This is the analysis of the our SAA numerical experiments and shown below in Table 2 are the reported results. We applied SAA for three settings. Table 2 not only presents the solutions but also the lower bounds, upper bounds, estimated optimality gap, and SAA variances. The bounds here are done with a significance level of $\alpha = 0.025$. For lower bound we take $N' = 10N$, since this gave reasonable runtime to find lower bound a bunch. A larger N' could be taken to decrease variance of lower bound but we were happy with the current variance.

Setting	$N = 1000, M = 5$	$N = 5,000, M = 5$	$N = 1000, M = 10$
Solutions (x)	$2, 3, 4, 5, 6, 7, 8, 9, 10$	$2, 3, 4, 5, 6, 7, 8, 9, 10$	$2, 3, 4, 5, 6, 7, 8, 9, 10$
Solutions (Q)	167.58	167.62	167.58
Lower bound	141.01	144.88	141.27
Upper bound	159.51	150.08	153.17
Optimality gap	18.5	5.21	11.89
Lower bound variance	4.065	0.814	4.053
Upper bound variance	6.636	1.435	4.185
Running time (s)	25.49	178.3	34.80

Table 2: SAA results

It follows from Table 2 that increasing the number of iteration that Monte Carlo simulation uses (N), the optimal selection and procurement quantity remain similar. Critically, is that the variance and optimality gap does decrease. Additionally, the influence of increasing M and N is affecting the running time, N logically more than M. Increasing the number of SAA runs (M) also has no effect on the selection and procurement quantity, although it does decrease the variance of the upper bound and thus the gap. Given that all of the solutions are very similar, we define our *SAA solution* as selecting customers $2, \dots, 10$ (not customer 1) with a procurement quantity of $Q = 167.6$.
Comparing this solution to the *improved solution*, we see that it is the same except for the marginally higher procurement quantity.

3.4 Spot Market Purchases and Correlated Demand

This section tackles the SNP with the additional complexity of spot market purchases and correlated demand. We work the approximate setting of one SAA run and 1000 scenarios ($M = 1, N = 1000$).

For the case of spot market purchases, we have the following conditions. The mean spot market price, μ^p, is set equal to the median stockout costs, $r_i + s_i$, for the customers, with the standard deviation chosen such that the coefficient of variation, $\frac{\sigma_p}{\mu^p}$, is 0.1.
For the case of correlated demand, we use the correlated demand matrix in Section

6.2 from Fleuren et al. (2024) to solve the problem:

$$\begin{bmatrix} 1 & 0.8 & 0.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.8 & 1 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7 & 0.6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -0.9 & 0.8 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.9 & 1 & -0.6 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & -0.6 & 1 & -0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0.3 & -0.6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -0.6 & -0.3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.6 & 1 & -0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.3 & -0.5 & 1 \end{bmatrix} \tag{8}$$

Table 3 presents the same outcome variables as found in Table 2, but spot market purchases and correlated demand are taken into account. Note that no value is given for the upper bound variance, since we are working with $M = 1$, so there is no variance in the upper bound estimate.

Setting	Correlated Demand	Spot Market Purchases	Both
Solutions (x)	$2, 3, 4, 5, 6, 7, 8, 9, 10$	$2, 3, 4, 5, 6, 7, 8, 9, 10$	$2, 3, 4, 5, 6, 7, 8, 9, 10$
Solutions (Q)	167.35	168.37	165.35
Lower bound	175.15	143.26	175.11
Upper bound	176.20	143.97	176.20
Optimality gap	1.05	0.71	1.09
Lower bound variance	2.865	4.153	2.871
Upper bound variance	NaN	NaN	NaN
Running time (s)	18.30	21.54	21.06

Table 3: Correlated demands and Spot Market results

Once again, the selection vectors for each of the solutions are the same, while the procurement quantity changes slightly. The optimality gaps are relatively similar, as well as the lower bound variances. Since the spot market prices were set to be, in expectation, equal to the shortage cost, which means we would expect a similar order quantity for the same selection vector, with and without spot market purchases. But we see that the Q for spot market purchases is higher, which indicates that we are more willing to procure more, and have excess than have to by the shortage in the market. But for both correlated demands and a spot market, the order quantity drops slightly. This might be due to shortage costs being negatively skewed.

4 Risk Averse Decision Making

4.1 Problem Formulation

We provide the two-stage stochastic programming problem for Risk Adverse Decision Making, based on the CVAR objective. VaR is Value at Risk and CVaR is the conditional Value at Risk. Again the first stage decision variables are x and Q. Then the second-stage decisions are I_m, q_{im}, l_{im}, e_{im}, and x_i, as well as $profit_m$, η_m, VaR and $CVaR$.

$$
\begin{aligned}
\max \quad & (1-\beta)\frac{1}{M}\sum_{m=1}^{M} profit_m + \beta \cdot CVaR \\
\text{s.t.} \quad & CVaR = VaR - \frac{1}{\alpha}\frac{1}{M}\sum_{m=1}^{M}\eta_m \\
& q_{im} + l_{im} + e_{im} = D_{im}x_i && i = 1,\dots,N,\ m = 1,\dots,M, \\
& \sum_{i=1}^{N} q_{im} + I_m = Q && m = 1,\dots,M, \\
& VaR - profit_m \leq \eta_m && m = 1,\dots,M, \\
& profit_m = \sum_{i=1}^{N}(r_iD_{im} - \Lambda_i)x_i - cQ - hI_m - \sum_{i=1}^{N}(\tilde{s}_i l_{im} + p_m e_{im}) && m = 1,\dots,M, \\
& I_m \geq 0,\ q_{im} \geq 0,\ l_{im} \geq 0, e_{im} \geq 0 && i = 1,\dots,N,\ m = 1,\dots,M, \\
& x_i \in \{0,1\} && i = 1,\dots,N, \\
& Q \geq 0.
\end{aligned}
$$

4.2 Analysis and Results for the Base Setting

Here we solved the problem for varying levels of α, and setting $\beta = 1$, and with uncorrelated demand and no spot market. In the following table we present the resulting demand selection vector, order quantity, CVaR and expected profit. For the selection vector x, we write the indices of the chosen customers, where the rest are not chosen and would be 0 in x. Note, also, that since $\beta = 1$, the optimal objective value will be the same as the $CVaR$.

α can be viewed as the risk tolerance level, where $\alpha = 0.9$ is risk-averse, while an $\alpha = 0.99$ is more risk-seeking. This is because the conditional value at risk determines the expected loss given that we are in the $1-\alpha$ worst cases. Hence as α increases, the CVAR increases. As Q and $profit$, these also increase with α since in these scenarios, it is more optimal to procure more, for more customers, and thus the expected profit is higher. However, this comes at a cost, with far higher losses beyond a point.

Q does not change much between $\alpha = 0.8$ and $\alpha = 0.9$, expected profit increases only a little, while the CVAR increases a fair bit. It would be, thus, advisable to set Q at around 146, and selecting customers $2, 3, 5, 6, 7, 8, 9, 10$, to mitigate risk.

α	x	Q	$CVAR$	$profit$
0.01	-	0	0	0
0.1	-	0	0	0
0.2	-	0	0	0
0.4	$3, 5, 6, 7, 8, 9, 10$	120.1	9.0	94.5
0.6	$2, 3, 5, 6, 7, 8, 9, 10$	142.6	53.1	120.3
0.8	$2, 3, 5, 6, 7, 8, 9, 10$	145.2	98.1	138.7
0.9	$2, 3, 5, 6, 7, 8, 9, 10$	147.3	132.1	153.4
1	$2, 3, 4, 5, 6, 7, 8, 9, 10$	168.0	138.8	138.8

Table 4: RADM results

4.3 Analysis and Results for Spot Market and Correlated Demands

Again we solved the problem for varying α and $\beta = 1$, but now with correlated demand and a spot market. For how we obtained the correlated demand, see Section 3.4. Below is the table with the resulting values.

α	x	Q	$CVAR$	$profit$
0.01	-	0	0	0
0.1	$3, 4, 5, 6, 7, 8, 9, 10$	134.2	13.9	100.9
0.2	$3, 4, 5, 6, 7, 8, 9, 10$	135.8	50.3	126.4
0.4	$3, 4, 5, 6, 7, 8, 9, 10$	137.5	86.1	145.4
0.6	$3, 4, 5, 6, 7, 8, 9, 10$	139.2	124.9	164.4
0.8	$3, 4, 5, 6, 7, 8, 9, 10$	140.4	142.9	166.6
0.9	$2, 3, 4, 5, 6, 7, 8, 9, 10$	164.0	153.3	174.5
1	$2, 3, 4, 5, 6, 7, 8, 9, 10$	164.5	173.2	173.2

Table 5: RADM results with correlated demands and spot market

Again we have similar increasing properties as α increases, for the same reasons. Although, for correlated demands and a spot market, the change in values as α increases is more gradual. Again, we would advise to select customers 2 through 10, with an order quantity of 164. This is because although the Condition Value at Risk increases, this value is only concerned with the 10% worst cases, while the profit still increases reasonably.

5 Conclusion

First we have shown in Section 1 the general problem at hand. The mathematical framework is based on the SNP and is further investigated by Fleuren et al. 2024. This provided us with the objective functions to maximize under certain constraints.

In Section 2, Monte Carlo simulations were used to estimate the profit for an initial solution, where the first five customers were selected, yielding an expected profit of -66.4. Calculating the expected profit (-65.8) directly using the loss-function barely improved the solution. The DERU-based heuristic massively improved this by selecting all but the first customer, increasing the expected profit to 145.4. Control variates were then applied, reducing the variance and providing a more precise profit estimate of 146.0, with a tighter confidence interval. Overall, this process demonstrated how both the heuristic and variance reduction improved profitability and accuracy.

Following from Section 3, we implemented a two-stage stochastic program using the SAA algorithm to estimate the optimality gap, lower and upper bounds and variance. For the instance of $N = 5000$ and $M = 5$, the optimality gap was 5.21, with lower and upper bounds of 144.88 and 150.08 , respectively. Increasing sample sizes reduced the variance as seen from Table 2. While selection and procurement decisions remained stable. From Table 3 we saw how incorporating real-world scenarios like spot market prices and correlated demand changed the procurement quantity. This showed how there is a preference for procuring more stock to avoid shortages.

Lastly, in Section 4 we took risk measures such as VaR and CVaR to account for Risk Adverse Decision Making in the stochastic program When the risk tolerance level α increased from 0.8 to 0.9, both CVaR and expected profit increased for a procurement quantity of around 146 to satisfy customers 2, 3, 5, 6, 7, 8, 9 and 10. In situations where correlated demand and spot prices were accounted for, the selected customers remained similar.

References

Fleuren, Tijn et al. (2024). "Integrated customer portfolio selection and procurement quantity planning for a supplier". In: *Omega* 128, p. 103126. ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2024.103126. URL: https://www.sciencedirect.com/science/article/pii/S0305048324000926.

Taaffe, Kevin, Joseph Geunes, and H. Edwin Romeijn (2008). "Target market selection and marketing effort under uncertainty: The selective newsvendor". In: *European Journal of Operational Research* 189.3, pp. 987–1003. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2006.11.049. URL: https://www.sciencedirect.com/science/article/pii/S0377221707006716.