The Transshipment Problem

University of Southern California
Spring 2022
ISE-533 31533

Group 7
Yiwen Chen (1678961165)
Jiachen Tu (2260088714)
Jiaqi Ma (3028009001)

1. Introduction

In this project, we solved the transshipment problem using the All-In-One Stochastic Linear Program (SLP) and the Stochastic Quasi-Gradient (SQG) methods. The transshipment problem is to minimize the long-run average costs between retailers and a supplier. The specific problem description and the steps of algorithms are illustrated in the following report.

2. The Transshipment Problem

2.1 Problem Description & Assumptions

The transshipment problem includes one supplier and *N* retailers at distinct locations satisfying orders from customers. It allows transshipment between any pair of retailers to meet their demands. The goal of the problem is to find an optimal order-up-to stock level quantity minimizing the long-run average costs including replenishment, transshipment, holding (for inventory) and penalty (for backorders) costs. We divide the long-run business into infinite periods which can be treated as identical. The whole process within one period is described as follows.

S: Order-up-to inventory (the same in every period)
B: Beginning inventory after replenishments and backorders met)
M: Demand at each retailer
R: Replenishment before next period
E: Ending inventory including units on order from supplier

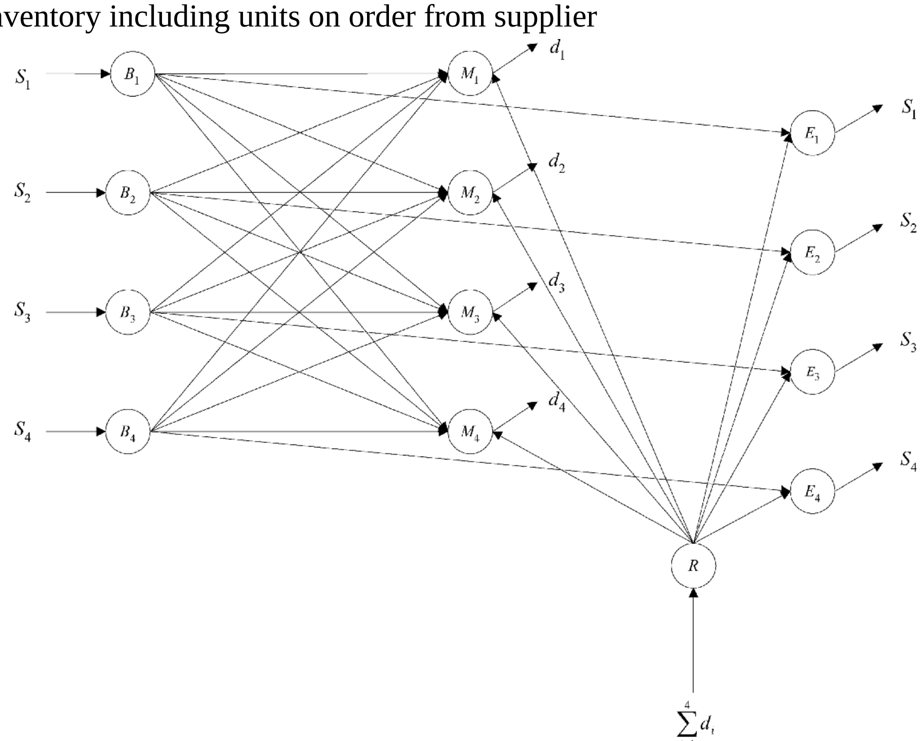

Figure 1: Network flow representation (Herer, 2006)

At the beginning of a period, every retailer has its beginning inventory according to order-up-to stock level. Then, each of the retailers has distinct demands from customers. Every retailer has three sources to satisfy the demands: (i) its own beginning; (ii) transshipment from any other retailers; (iii) replenishment from supplier to satisfy backorders. After all the demands are met in this period, all the retailers will replenish their inventory to order-up-to stock level again and start next period.

In our model, we assume that retailers are non-identical and each retailer faces uncertain customer demand which distribution is stationary. Also, we assume transshipment policies are stationary and no-buildup, replenishment policies are non-shortage inducing. To be specific, stationary means the order placed by each retailer only depends on pre-transshipment inventory and the observed demand. No-buildup means transshipments are only made to satisfy actual observed demand. Non-shortage inducing means the beginning inventory (after replenishments and backorders met) cannot be negative.

2.2 Notations

The following notations are used to set up the transshipment problem.

N = number of retailers
S_i = order-up-to quantities at retailer i
D_i = Demand realization at retailer i
e_i = Ending inventory held at retailer i
f_i = Stock used to satisfy demand at retailer i
q_i = Inventory increased through replenishment at retailer i
r_i = Amount of shortage met after replenishment at retailer i
t_{ij} = Stock at retailer i used to meet demand at retailer j by transshipment
h_i = Unit cost of holding inventory at retailer i
c_{ij} = Unit cost of transshipment from retailer i to retailer j
p_i = Penalty cost for shortage at retailer i

2.3 Primal Program

The goal of this transshipment problem is to find an order-up-to S policy that minimizes the cost of the system while satisfying the relationships in the network flow:

$$h(S,D) = \min \sum_i h_i e_i + \sum_{i \neq j} c_{ij} t_{ij} + \sum_i p_i r_i \qquad (1)$$

$$f_i + \sum_{j \neq i} t_{ij} + e_i = s_i, \quad \forall i \qquad (1a)$$

$$f_i + \sum_{j \neq i} t_{ji} + r_i = d_i, \quad \forall i \tag{1b}$$

$$\sum_i r_i + \sum_i q_i = \sum_i d_i \tag{1c}$$

$$e_i + q_i = s_i, \quad \forall i \tag{1d}$$

$$e_i, f_i, q_i, r_i, s_i, t_{ij} \geq 0, \quad \forall i, j.$$

2.4 Data

The following are the data to be used in solving the program.

Number of Retailers, $N = 7$

The transshipment system is assumed to contain 7 different retailers.

Holding Cost, $h = 1$

The unit cost of holding inventory is assumed to be the same across all retailers at $1.

Transshipment Cost, $c = 0.1$

The unit cost of transshipment across all retailers is uniform at $0.1.

Backorder Cost, $p = 4$

The unit cost of backorder when shortage exists at retailers is uniform at $4.

Demand, D_i

The demand of each retailer is assumed to be normally distributed random variables with mean and standard deviation shown in Table 1.

Table 1: Parameters of random variable D_i

Retailer	1	2	3	4	5	6	7
Mean	100	200	150	170	180	170	170
SD	20	50	30	50	40	30	50

Using quantiles of distributions to represent the demand scenarios is a simple way. The 25th, 50th, and 75th percentile are then generated to simulate the low, medium and high level of demand. This would give a total of $M = 3^7 = 2187$ different combinations of demand scenarios.

Table 2: Levels of demand

Retailer	1	2	3	4	5	6	7
Low	86.51	166.28	129.77	136.28	153.02	149.77	136.28
Medium	100.00	200.00	150.00	170.00	180.00	170.00	170.00
High	113.49	233.72	170.23	203.72	206.98	190.23	203.74

Another way of generating scenarios is to sample from normal distributions with parameters in Table 1. We use truncated normal distributions in Table 3 to generate sample points which are within limit of $\mu \pm 3\sigma$.

Table 3: Truncated normal distributions

Retailer	1	2	3	4	5	6	7
Lower Bound	40	50	60	20	60	80	20
Mean, SD	100, 20	200, 50	150, 30	170,50	180,40	170,30	170,50
Upper Bound	160	350	240	320	300	260	320

3. Models

3.1 All-in-One Stochastic Linear Program (SLP)

With 2187 different demand scenarios simulated, we can determine the order-up-to policy S for each scenario using the basic program (1). Assuming that all the demand scenarios provide a well simulation of the real-world long run demand situations, we can modify the model to determine one order-up-to policy S that takes all 2187 scenarios into consideration and minimizes the average long-run cost of the transshipment system.

The order-up-to quantities that minimize the average long-run costs can be obtained by solving the following objective

$$\min E\left[h\left(S, \widetilde{D}\right)\right] = \min \frac{1}{M} \sum_{k} \left(\sum_{i} he_{ki} + \sum_{i \neq j} ct_{kij} + \sum_{i} pr_{ki} \right) \qquad (2)$$

subject to

$$f_{ki} + \sum_{j \neq i} t_{kij} + e_{ki} = s_{ki}, \quad \forall i, k \qquad (2a)$$

$$f_{ki} + \sum_{j \neq i} t_{kji} + r_{ki} = d_{ki}, \quad \forall i, k \qquad (2b)$$

$$\sum_i r_{ki} + \sum_i q_{ki} = \sum_i d_{ki}, \quad \forall k \qquad (2c)$$

$$e_{ki} + q_{ki} = s_{ki}, \quad \forall i, k \qquad (2d)$$

$$e_{ki}, f_{ki}, q_{ki}, r_{ki}, s_{ki}, t_{kij} \geq 0, \quad \forall i, j, k.$$

where k denotes the different demand scenarios.

3.2 Stochastic Quasi-Gradient (SQG)

The SQG method estimates the subgradient of $E[h(S, \widetilde{D})]$ by solving the dual formulation.

The dual program of (1) is given by

$$\max \sum_i s_i B_i + \sum_i d_i M_i + \sum_i d_i R + \sum_i s_i E_i \qquad (3)$$

subject to

$$B_i + E_i \leq h_i, \quad \forall i \qquad (3a)$$

$$B_i + M_i \leq 0, \quad \forall i \qquad (3b)$$

$$B_i + M_j \leq c_{ij}, \quad \forall i, j, i \neq j \qquad (3c)$$

$$M_i + R \leq p_i, \quad \forall i \qquad (3d)$$

$$R + E_i \leq 0, \quad \forall i \qquad (3e)$$

where B_i , M_i , R_i and E_i denotes the dual multipliers of (1a) - (1d).

Steps of SQG method:

1. Initial S
2. M i.i.d. scenarios are selected, denoted as $D^1, \ldots, D^M$
3. Solve the dual problem and get dual multipliers $B^{k,m}$ and $E^{k,m}$
4. Calculate the unbiased estimate of subgradient: $\hat{\xi}^k = \frac{1}{M}\sum_{m=1}^{M}(B^{k,m} + E^{k,m})$
5. Update S: $S^{k+1} = P_+(S^k - \alpha_k \hat{\xi}^k)$

6. Repeat Steps 2 - 5 until stop condition (predetermined maximum number of iterations N) is met

Note that $P_+(\cdot)$ denotes the positive part of the argument, and α_k are nonnegative scalars that satisfy $\alpha_k \rightarrow 0$, $\sum_k \alpha_k \rightarrow \infty$, and $\sum_k \alpha_k^2 < \infty$. Learning rate $\alpha_k = \frac{N/a}{i}$ where a is a preset coefficient and i is the iteration number. Therefore, the learning rate is decreasing along with the iterations. As default, the initial order-up-to level S is set as the mean of the normal distribution, $M = 100$, $N = 100$, and $a = 10$. Figure 2 shows the values of learning rate α_k along with iteration i in our model.

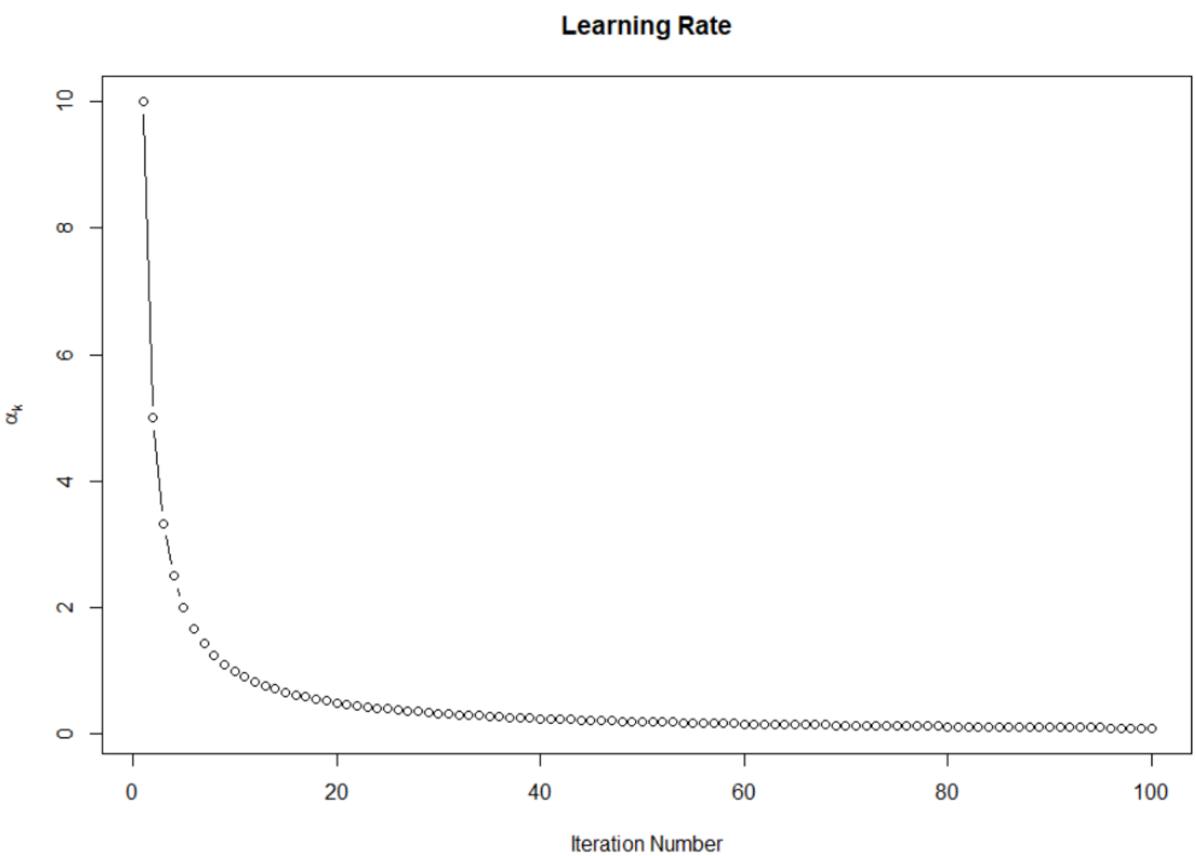

Figure 2: Learning Rate vs. Iteration Number

4. Model Outputs & Analysis

4.1 Result using quantile samples

Table 3: Outputs of SLP and SQG model

	SLP model	SQG model	95% CI (SQG model)
S (Retailer 1)	104.50	107.22	[107.18, 107.27]
S (Retailer 2)	208.99	207.41	[207.36, 207.46]
S (Retailer 3)	156.74	157.45	[157.40, 157.50]
S (Retailer 4)	178.99	177.37	[177.32, 177.42]
S (Retailer 5)	188.99	187.35	[187.30, 187.40]
S (Retailer 6)	176.74	177.44	[177.40, 177.49]
S (Retailer 7)	178.99	177.52	[177.47, 177.57]
Total Cost	$85.99	$86.01	NA

4.2 Result using truncated normal distribution samples

Table 4: Output of SLP and SQG models with truncated normal distribution samples

	SLP model	SQG model	95% CI (SQG model)
S (Retailer 1)	105.85	111.76	[111.72, 111.80]
S (Retailer 2)	215.59	212.35	[212.31, 212.39]
S (Retailer 3)	160.19	162.10	[162.06, 162.14]
S (Retailer 4)	187.30	182.41	[182.37, 182.45]
S (Retailer 5)	191.79	192.23	[192.19, 192.26]
S (Retailer 6)	179.33	182.05	[182.01, 182.09]
S (Retailer 7)	184.77	182.32	[182.28, 182.35]
Total Cost	$149.96	$150.02	NA

4.3 Convergence of order-up-to level

The convergence of S is shown in Figure 3. All the variables converge very quickly.

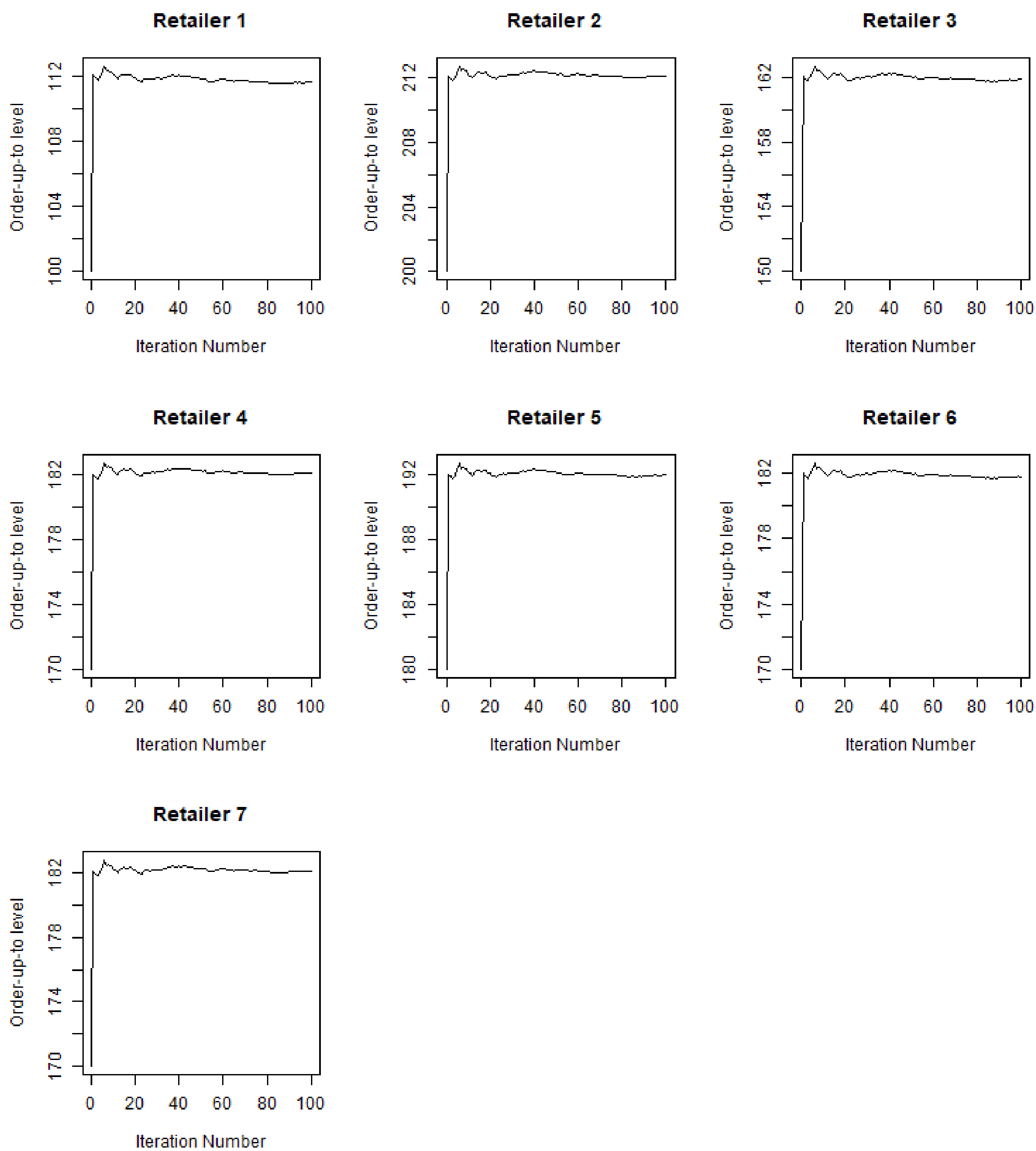

Figure 3: Convergence of order-up-to level in SQG model

5. Sensitivity Analysis on Convergence in SQG

When using the SQG method, slight changes in parameters can sometimes lead to very different results. This is common in gradient descent methods. To have a brief understanding of effects of different parameters on convergence, we conduct some trails by tuning them. All the trials are run using SQG method for 100 replications with the same random seed making the results comparable.

Here, we try different values of initial stock level and learning rate. All these changes are compared with SQG method results with default values and the same random seed.

5.1 Initial value of stock level

The original initial values of stock level are set as the mean of the normal distributions. But in real world situations, we may not be able to estimate the distributions of demands exactly. We could start with zeros or other arbitrary numbers which may be larger. Here we try zero and three hundred as initial values of stock levels while keeping all other parameters not changed. The results are shown in Table 5 and 6.

Table 5: Output of SQG model with initial stock level equal to zero comparing with former result

	SQG model (Initial stock level = mean)	SQG model (Initial stock level = 0)
Retailer 1	111.76 95%CI: [111.72, 111.80]	168.64 95%CI: [168.60, 168.67]
Retailer 2	212.35 95%CI: [212.31, 212.39]	170.83 95%CI: [170.79, 170.86]
Retailer 3	162.10 95%CI: [162.06, 162.14]	170.04 95%CI: [170.00, 170.07]
Retailer 4	182.41 95%CI: [182.37, 182.45]	170.37 95%CI: [170.33, 170.40]
Retailer 5	192.23 95%CI: [192.19, 192.26]	170.63 95%CI: [170.60, 170.67]
Retailer 6	182.05 95%CI: [182.01, 182.09]	170.56 95%CI: [170.52, 170.59]
Retailer 7	182.32 95%CI: [182.28, 182.35]	170.36 95%CI: [170.32, 170.39]
Total Cost	$150.02	$161.71

Table 6: Output of SQG model with initial stock level equal to three hundred comparing with former result

	SQG model (Initial stock level = mean)	SQG model (Initial stock level = zero)

Retailer 1	111.76 95%CI: [111.72, 111.80]	248.13 95%CI: [248.13, 248.13]
Retailer 2	212.35 95%CI: [212.31, 212.39]	248.47 95%CI: [248.47, 248.48]
Retailer 3	162.10 95%CI: [162.06, 162.14]	248.13 95%CI: [248.13, 248.13]
Retailer 4	182.41 95%CI: [182.37, 182.45]	248.24 95%CI: [248.24, 248.24]
Retailer 5	192.23 95%CI: [192.19, 192.26]	248.19 95%CI: [248.19, 248.19]
Retailer 6	182.05 95%CI: [182.01, 182.09]	248.13 95%CI: [248.13, 248.13]
Retailer 7	182.32 95%CI: [182.28, 182.35]	248.22 95%CI: [248.21, 248.22]
Total Cost	$150.02	$600.95

The result of SQG method with initial stock level equal to zero is close to optimal since its error rate of objective value is less than 10%. However, the result of SQG method with initial stock level as three hundred does not perform well. It seems not converged. We will discuss this situation later.

5.2 Learning rate

Learning rate is also a very important parameter in SQG method. It usually cooperates with the initial value set in the model. In the process of gradient descent (quasi-gradient descent in SQG), the step sizes in each iteration are usually set as the product of learning rate and gradient. In our SQG model, learning rate is

$$\alpha_k = \frac{N/a}{i} = \frac{N}{a \cdot i}$$

Recall that N is the maximum number of iterations, a is a preset coefficient and i is the iteration number. When N is fixed, we could control the learning rate by tuning coefficient a. As default, we set $a = 10$ in our SQG model. It determines that the step size is equal to $\frac{1}{a}$ in the last iteration. A smaller coefficient a makes the step size bigger in all iterations. But with the change of initial stock level, the learning rate should also be tuned and may lead to a better result. Here, following the two trails above in Section 5.1, we further tune the models by setting $a = 2, 5$.

The result of models with initial stock level equal to zero shown in Table 7 does not change much while tuning the learning rate. It performs not bad with default value of coefficient a. When we reduce the coefficient a to 5, the objective value gets closer to the optimal. But when we continue to reduce the coefficient a to 2, making the learning rate bigger, the objective value only gets a little bit smaller. It may tell us that in this case, the learning rate is appropriate. Keeping on increasing the learning rate is not a good choice now since a learning rate that is too large can cause fluctuations in the gradient descent process, making the target higher.

To prove our conjecture, the plot showing the convergence of variable S in models with different values of coefficient a is shown in Figure 4. In the models with initial stock level equal to zero, the models with a equal to 2 and 5 converge quickly. While the model with $a = 10$ may need more steps to get close enough to optimal.

Table 7: Output of SQG model with initial stock level equal to zero with different coefficient a

	Stock level = 0 a = 10	Stock level = 0 a = 5	Stock level = 0 a = 2
Retailer 1	168.64 [168.60, 168.67]	171.62 [171.57, 171.66]	167.21 [167.14, 167.27]
Retailer 2	170.83 [170.79, 170.86]	177.09 [177.04, 177.13]	180.86 [180.80, 180.93]
Retailer 3	170.04 [170.00, 170.07]	173.87 [173.83, 173.92]	171.33 [171.26, 171.40]
Retailer 4	170.37 [170.33, 170.40]	175.50 [175.46, 175.54]	176.36 [176.29, 176.43]
Retailer 5	170.63 [170.60, 170.67]	176.15 [176.11, 176.20]	177.81 [177.74, 177.88]
Retailer 6	170.56 [170.52, 170.59]	175.46 [175.42, 175.50]	175.39 [175.32, 175.46]
Retailer 7	170.36 [170.32, 170.39]	175.47 [175.43, 175.51]	176.21 [176.15, 176.28]
Total Cost	$161.71	$152.86	$152.50

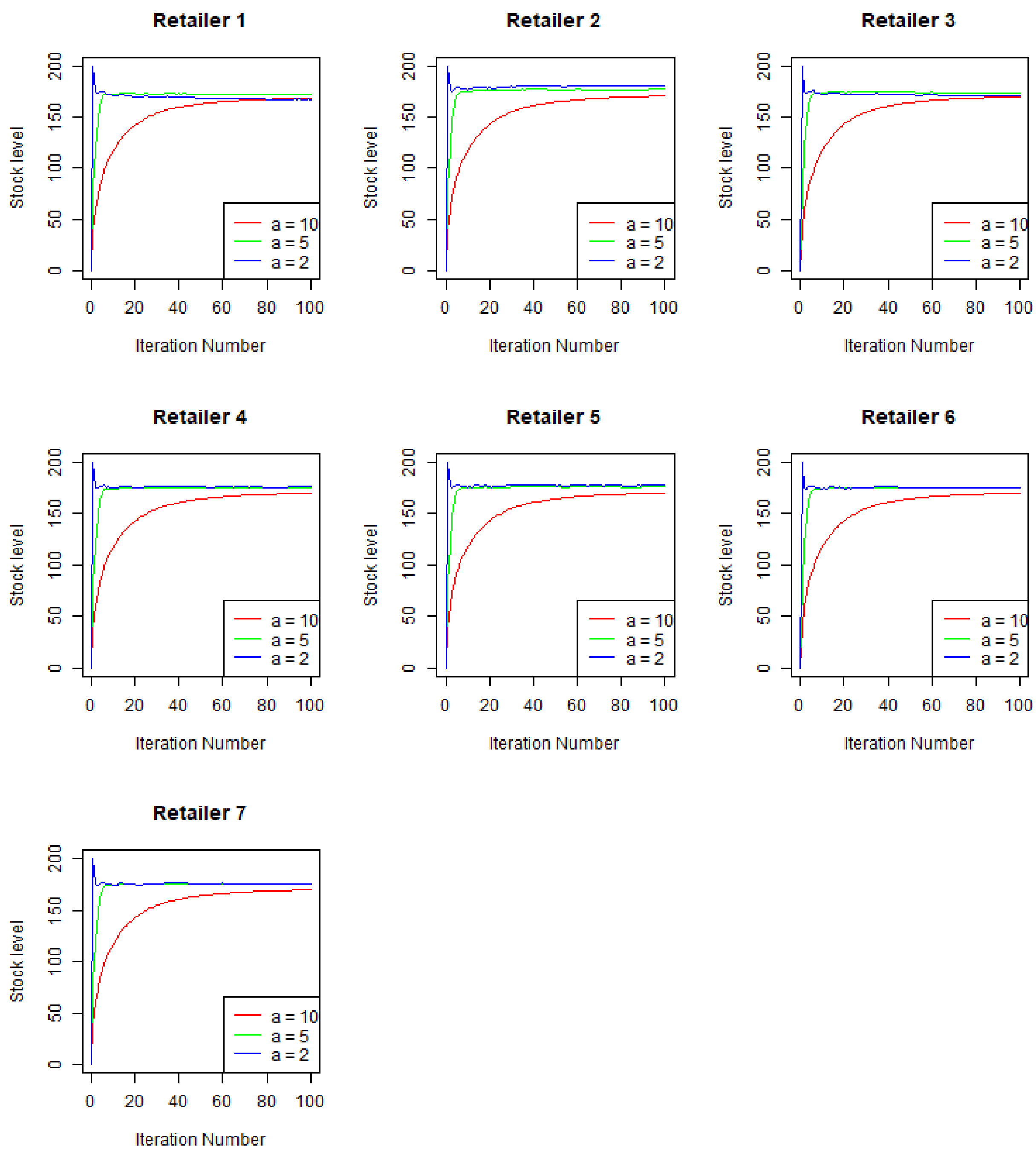

Figure 4. Convergence Process of Stock level S in SQG model (first replication) with initial stock level equal to zero

Table 8. Output of SQG model with initial stock level equal to three hundred with different coefficient a

	Stock level = 300 a = 10	Stock level = 300 a = 5	Stock level = 300 a = 2
Retailer 1	248.13 [248.13, 248.13]	196.48 [196.47, 196.48]	168.66 [168.60, 168.72]
Retailer 2	248.47 [248.47, 248.48]	198.43 [198.42, 198.44]	180.32 [180.24, 180.39]
Retailer 3	248.13 [248.13, 248.13]	196.55 [196.54, 196.56]	171.77 [171.70, 171.83]
Retailer 4	248.24 [248.24, 248.24]	197.38 [197.37, 197.39]	176.15 [176.08, 176.21]
Retailer 5	248.19 [248.19, 248.19]	197.32 [197.31, 197.33]	177.22 [177.16, 177.29]
Retailer 6	248.13 [248.13, 248.13]	196.76 [196.75, 196.77]	175.05 [174.98, 175.11]
Retailer 7	248.22 [248.21, 248.22]	197.35 [197.34, 197.36]	176.03 [175.96, 176.09]
Total Cost	$600.95	$249.52	$152.59

However, from the result of SQG models with initial stock level equal to three hundred shown in Table 8, the change of learning rate performs well. At the beginning, the model with coefficient $a = 10$ does not converge. When we reduce the coefficient a to 5, though it still does not converge, its objective value gets much smaller than before. When we continue to reduce a to 2, it seems to converge and its objective value becomes very close to optimal. From Figure 5 showing the convergence of variable S, the model with a equal to 2 performs the best. The other models with $a = 5$ and 10 do not converge.

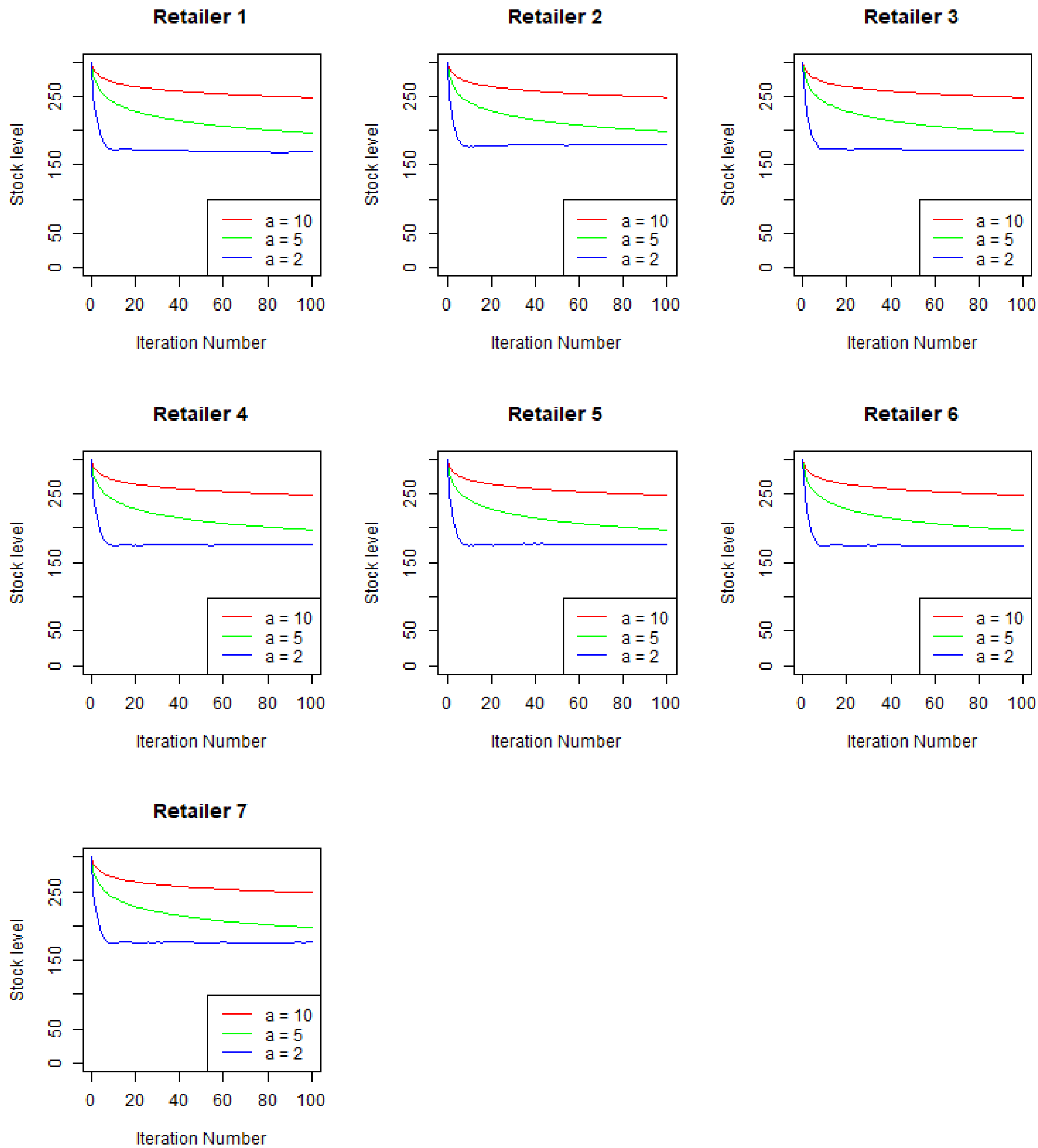

Figure 5. Convergence Process of Stock level S in SQG model (first replication) with initial stock level equal to three hundred

6. Conclusion

By using the all-in-one stochastic linear program method, the exact solution could be solved under pre-assumed variable distribution. Therefore, a variable distribution which is close enough to reality is required. But it also brings challenges to the solver for solving such a huge LP program. While using the stochastic quasi-gradient method, an approximation solution could be solved. It only needs to solve many much smaller LP programs consecutively. Similar to the all-in-one method, it also requires a variable distribution which is close to reality to ensure the solution closer to the true value.

According to the result in Section 5, it should pay more attention when using SQG method. Different starting points and learning rates may lead to very different results. It is recommended to try many different combinations of these parameters before giving conclusions. In the transshipment problem discussed here, a smaller starting point converges faster than a larger one. Besides, increasing number of iterations is also a good direction after finding out a set of suitable combinations of starting point and learning rate.