A NOTE ON ALGORITHMS FOR DUAL SOURCING INVENTORY SYSTEMS WITH TAILORED BASE-SURGE POLICIES

CHENBO ZHU¹, ZEQIONG REN¹, LEI LEI²*, XING WANG³ AND YUE XIE⁴

Abstract. This paper proposes two algorithms for dual sourcing inventory systems with Tailored Base-Surge policies. The first algorithm solves a system of linear equations to calculate inventory positions, performance measures and their derivatives, and the second algorithm then uses them in the gradient-based type of method to find the optimal Tailored Base-Surge policy. Numerical experiments show that the two algorithms work well.

Mathematics Subject Classification. 90B05.

Received March 7, 2023. Accepted April 26, 2024.

1. Introduction

Nowadays supply chain risk particularly supplier reliability is a growing concern for companies, because there exist many external reasons (such as weather and natural disasters, and COVID-19) and internal reasons (such as machine failure, and defect products) which cause supply disruptions. A common mitigation strategy is for the buying companies to source from dual or multiple suppliers. Using the dual sourcing strategy, a company usually has a regular supplier with a low unit cost but a long lead time and an emergency supplier with a high unit cost but a short lead time. In general, dual sourcing can improve supplier reliability, reduce procurement costs, and make supply chains more responsive and flexible. For example, HP has a major supplier in Singapore and a local supplier in Vancouver for its DeskJet printers, the local supplier can respond more quickly to increased demands or supply disruptions in the North American market [2].

However, the optimal inventory policy is complicated for the dual sourcing inventory system. Sethi et al. [12] investigated a dual sourcing periodic review inventory system, in which inventory and demand information can be updated in each period, and they proved that a forecast-update-dependent \((s, S)\)-type policy is optimal for the system. Both Zhang et al. [19] and Zhang and Hua [18] studied an inventory system with two suppliers, one with a high variable cost but negligible fixed cost, and the other with a low variable cost but high fixed cost, and they partially characterized the structure of the optimal policy for the system by using an approach based on the preservation of quasi-convexity and the notions of quasi-convexity and strong CK-convexity respectively. Hua

Keywords. Dual sourcing inventory systems, tailored base-surge policies, infinite linear equations, optimization.

¹ School of Management, Zhejiang University of Technology, Hangzhou, Zhejiang 310023, P.R. China.
² Economics and Business Administration, Chongqing University, Chongqing 400044, P.R. China.
³ China Mobile Group Zhejiang Co., Ltd., TaiZhou, Zhejiang 318014, P.R. China.
⁴ School of Economics, Zhejiang University of Technology, Hangzhou, Zhejiang 310023, P.R. China.
*Corresponding author: leilei312@cqu.edu.cn

© The authors. Published by EDP Sciences, ROADEF, SMAI 2024

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
et al. [8] considered dual sourcing inventory systems with general lead times, and characterized the structural properties of the optimal orders by using the notion of \(L^3\)-convexity and proposed a heuristic policy for the system. Recently, Chen and Yang [3] studied the periodic dual sourcing inventory system with uncertainty of demand and supply, and designed a linear programming greedy heuristic based on the quadratic approximation of the \(L^3\) convex value-to-go function to calculate the optimal policy.

Several approximate or heuristic policies for the dual sourcing inventory system have been proposed in the literature, such as the Dual Index (DI) policy [13,16], the Capped Dual Index (CDI) policy [15], and the Tailored Base-Surge (TBS) policy [1,9,17]. TBS policy is a very simple but effective practical policy for the dual sourcing inventory system, which orders at a constant rate from the regular supplier and replenishes from the emergency supplier by using a base-stock policy. Since Allon and Van Mieghem [1] proposed the TBS policy, several papers have been studied it. Janakiraman et al. [9] proved that the best TBS policy is close to the optimal inventory policy for a dual sourcing system, and numerically showed that the inventory cost difference between the best TBS policy and the optimal policy decreases as the lead time difference between the two suppliers increases. Xin and Goldberg [17] proved that the TBS policy is asymptotically optimal as the lead time difference between the two sources grows large. Dong et al. [5] developed a modal split transport model, which is a generalization of the classical TBS policy and assumes the fast mode delivers exactly twice as frequent as the slow mode, and provided approximate closed-form solutions of the modal split. Dong and Transchel [4] then constructed a more generalised TBS dual sourcing inventory model to support firms’ modal split transport optimization, and obtained structural properties of the model, which allow finding optimal solutions by using dynamic programming and bisection search. Recently, Sripad et al. [14] considered the Tailored Base-Surge policy for managing the dual sourcing system, and proved an asymptotic convergence result on the cost of an optimal TBS policy, which can provide the best known upper bound on the optimal cost of the dual sourcing system with the TBS policy. Hamdouch et al. [7] investigated a dual sourcing inventory system in a two-echelon setting, where the buyer makes ordering decisions and the two suppliers make production decisions, and compared the performance of the Dual-Index policy and the Tailored Base-Surge policy by using a simulation-based optimization approach.

It would be important to compute various performance metrics for a given TBS policy and determine the optimal TBS policy. Besides simulation and approximation methods, Zhu [20] studied a dual sourcing inventory system with TBS policies, and proposed a method to calculate the moments of the inventory position and performance measures by using the techniques of MacLaurin series analysis. Numerical experiments showed that their method is effective and efficient. However, none of previous studies provide a good method to determine the optimal TBS policy, mainly because it is not easy to calculate inventory positions, performance measures and their derivatives quickly for the dual sourcing inventory system.

In this paper, we would first develop an analytical solution to calculate the moments of the inventory position, performance measures and their derivatives based on solving a system of linear equations, and numerically show that our method performs better than the method proposed in Zhu [20] in general. Once we can quickly obtain the inventory system’s performance measures and their derivatives, which could be used in any gradient-based type of algorithm, we would then apply a gradient-based type of algorithm to calculate the optimal TBS policy for the dual sourcing inventory system. Numerical examples show that the analytical solution of performance measures and their derivatives and the gradient-based type of algorithm work well, though they have convergence radii.

The rest of this paper is organized as follows. In Section 2, we introduce the dual sourcing inventory system with TBS policies. In Section 3, we show how the inventory position, the performance measure and its derivatives can be calculated, and how to determine the optimal TBS policy by using the gradient-based type of algorithm. In Section 4, we provide some numerical experiments. Finally, Section 5 contains a conclusion and some discussions.

## 2. The dual sourcing inventory system with TBS policies

Consider a periodic review dual sourcing inventory system with stationary demands, TBS Policies, full backlogging and the long-run average-cost criterion, and note that the TBS policy requires stationary demand
processes and the demand can be correlated. The system has a regular supplier with a low unit cost but a long lead time, and an emergency supplier with a high unit cost but a short lead time. \( t^R \) and \( t^E \) denote the lead time for the regular supplier and the emergency supplier respectively, and let \( l^S = t^R - t^E \). Let \( D_n \) be the demand in period \( n \) (independent and identically distributed over all periods), and \( D \) be a generic demand in a period with mean \( \mu \), distribution function \( F(\cdot) \), and density function \( f(\cdot) \). Suppose the inventory position (the inventory level on hand plus on order) at the beginning of period \( n \) is \( X_n \), and the expedited inventory position (the inventory level on hand and all the outstanding orders from both suppliers that will be delivered within the next \( t^E \) periods) at the beginning of period \( n \) can be calculated as \( X_n - l^E Q \). Under the TBS policy, at the beginning of period \( n \), the system places an order of a quantity of \( Q \) (\( Q < \mu \) is required to make the system stable) from the regular supplier, and if \( X_n - l^E Q \) is less than a target inventory level \( S \), it also places an order for an amount to bring the inventory position up to \( S \) from the emergency supplier; otherwise, it does not order from the emergency supplier.

Let \( \hat{S} = \hat{S} + l^E Q \), and \( O_n = X_{n+1} + D_n - \hat{S} - Q \). Under some mild conditions by using the wide-sense regenerative structure of the process (Harris recurrence), the process \( \{X_n, O_n, D_n\} \) is stable and ergodic, i.e., \( \{X_n, O_n, D_n\} \xrightarrow{d} \{X, O, D\} \) (\( \xrightarrow{d} \) denotes the convergence in distribution), see Fu and Hu [6] for reference. Zhu [20] derived the following equation

\[
E[O^k] = \sum_{j=0}^{\infty} \alpha_{kj} Q^{i+k+1} + \sum_{j=0}^{\infty} \alpha_{kj} \sum_{r=0}^{j+k} \binom{j+k+1}{r} Q^r E[O^{i+k+1-r}],
\]

for \( k = 1, 2, \cdots \), where \( \frac{n!}{m!(n-m)!} \) and \( \alpha_{kj} = \frac{k!f^{(i)}(0)}{(k+j+1)!} \). Once \( E[O^k] \) is obtained, \( E[X^k] \) can be calculated easily.

To derive the performance measure of the system, let \( h \) be the per-unit holding cost, \( b \) be the per-unit backlog cost, \( c^R \) be the unit-cost charged by the regular supplier, and \( c^E \) be the unit-cost charged by the emergency supplier. We first define the emergency order quantity for period \( n \) as \( U_n = \max\{\hat{S} - X_n, 0\} \). Then the inventory level on hand, the surplus, and the backlog at the beginning of period \( n \) can be defined as \( Y_n = X_n - t^E Q - \sum_{i=0}^{t^E-1} U_{n-i}, V_n = \max\{Y_n, 0\}, \) and \( W_n = \max\{-Y_n, 0\} \), respectively. Similar to the process \( \{X_n, O_n, D_n\} \), under some mild conditions, \( \{U_n, Y_n, V_n, W_n\} \) is stable and ergodic, i.e., \( \{U_n, Y_n, V_n, W_n\} \xrightarrow{d} \{U, Y, V, W\} \).

We now define the performance measure as \( G(Q, S) = c^R Q + c^E E[U] + h E[V] + b E[W] \), where \( c^R Q \) denotes the procurement cost for the regular supplier per period, \( c^E E[U] \) denotes the expected procurement cost for the emergency supplier per period, \( h E[V] \) denotes the expected holding cost per period, and \( b E[W] \) denotes the expected backlog cost per period. According to Zhu [20], \( E[U] = E[D] - Q, E[V] \) can be calculated as

\[
E[V] = \sum_{j=0}^{\infty} \frac{\alpha_{1j}}{(l^E + 1)^{j+1}} \sum_{i=0}^{j+2} \sum_{l=0}^{j-i} \frac{S^{l-i}(j+2)!Q^i}{(j+2-i)!(l-i)!!} E[O^{i+j-2}],
\]

and \( E[W] = E[V] + (l^E + 1) E[D] - E[O] - S - Q \). Therefore, given \( Q \) and \( S \), the performance measure \( G(Q, S) \) can be calculated once \( E[O^k] \) \( (k = 1, 2, \cdots) \) are obtained.

3. Algorithms to calculate inventory positions and optimal TBS policies

Zhu [20] proposed a numerical approach based on the method of MacLaurin series expansion to calculate \( E[O^k] \) \( (k = 1, 2, \cdots) \), and showed that the method works well. In this section, we first develop an analytical solution to calculate \( E[O^k] \) \( (k = 1, 2, \cdots) \) directly based on solving a system of linear equations (we name it Algorithm 1), and we will show that Algorithm 1 performs better than the approach proposed in Zhu [20] in general in Section 4. We then propose a simple method (we name it Algorithm 2) to find optimal values of \( Q \) and \( S \) for the system, and will validate the method in Section 4. Our approaches are suitable for environments with
stationary demands, e.g., we consider inventory systems with i.i.d. demands in this paper, and our approaches can be further extended to dual sourcing inventory systems with correlated demands.

We consider an infinite system of linear equations

\[
\begin{bmatrix}
\xi_1 \\
\xi_2 \\
\xi_3 \\
\xi_4 \\
\vdots
\end{bmatrix} = \begin{bmatrix}
a_{11} & a_{12} & a_{13} & a_{14} & \cdots \\
a_{21} & a_{22} & a_{23} & a_{24} & \cdots \\
a_{31} & a_{32} & a_{33} & a_{34} & \cdots \\
a_{41} & a_{42} & a_{43} & a_{44} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix} \begin{bmatrix}
\xi_1 \\
\xi_2 \\
\xi_3 \\
\xi_4 \\
\vdots
\end{bmatrix} = \begin{bmatrix}
\gamma_1 \\
\gamma_2 \\
\gamma_3 \\
\gamma_4 \\
\vdots
\end{bmatrix},
\]

(2)

where \( \xi_i \)'s are variables, \( a_{ij} \)'s and \( \gamma_i \)'s are parameters. Kantorovich and Akilov [10] proved that, if \( a_{ij} \)'s satisfy \( \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} |a_{ij}|^2 < \infty \), then this system has one and only one bounded solution \( \{ \xi_i^* ; i = 1, 2, \ldots, N \} \). Furthermore, if \( \{ \xi_i^N ; i = 1, 2, \ldots, N \} \) is the solution of the finite system of linear equations \( \xi_i - \sum_{j=1}^{N} a_{ij} \xi_j = \gamma_i (i = 1, 2, \ldots, N) \), then \( \{ \xi_i^N ; i = 1, 2, \ldots, N \} \) is uniformly bounded, i.e., \( |\xi_i^N| < M \) for some constant \( M \), and \( \xi_i^* = \lim_{N \to \infty} \xi_i^N, i = 1, 2, \ldots, N \). (1) is a special example of (2), therefore, (1) has a unique solution, and \( E[O^k] \) can be calculated directly by solving a finite system of linear equations. Note that this approach has a convergence condition, i.e., \( \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} |a_{ij}|^2 < \infty \), if the condition is not satisfied, the approach would diverge. For this approach, the computational complexity with respect to \( Q, s \), and \( N \) (the number of linear equations) are \( O(c), O(c), \) and \( O(N^2) \), respectively. We note that

\[
G(Q, S) = c^R Q + c^E E[U] + h E[V] + b E[W] = c^R Q + c^E E[D] - Q + b \left( (l^E + 1) E[D] - E[O] - S - Q \right) + (h + b) \left( \sum_{j=6}^{\infty} \frac{\alpha_{1j}}{(l^E + 1)^{j+1}} \sum_{l=0}^{j} \sum_{i=0}^{l} S^{l-i}(j+2)!Q^{i}E[O^{i+2-l}] \right). \]

(3)

It is clear that the derivatives of \( G(Q, S) \) with respect to \( S \) can be calculated directly, and the derivatives of \( G(Q, S) \) with respect to \( Q \) can be computed by using finite difference methods. We can obtain both their first and second derivatives, therefore gradient-based algorithms such as gradient descent methods and Newton type of algorithms can be used to find the optimal values of \( Q \) and \( S \). Li and Fukushima [11] proposed a cautious BFGS (CBFGS) algorithm, and proved that the method can converge globally. We will apply the CBFGS algorithm to illustrate how our method can calculate optimal TBS policies. The first derivatives of \( G(Q, S) \) with respect to \( Q \) and \( S \) can be calculated as:

\[
\frac{\partial G(Q, S)}{\partial Q} = \frac{G(Q + \frac{\Delta Q}{2}, S) - G(Q - \frac{\Delta Q}{2}, S)}{\Delta Q},
\]

\[
\frac{\partial G(Q, S)}{\partial S} = -b + (h + b) \left( \sum_{j=6}^{\infty} \frac{\alpha_{1j}}{(l^E + 1)^{j+1}} \sum_{l=0}^{j} \sum_{i=0}^{l} S^{l-i-1}(j+2)!Q^{i}E[O^{i+2-l}] \right). \]

We then present the detailed steps of the CBFGS algorithm integrated with Algorithm 1 below as an illustration of Algorithm 2 to calculate the optimal \( Q \) and \( S \).
Algorithm: The CBFGS algorithm integrated with Algorithm 1.

1. Let \( k = 0 \). Choose an initial point \( x_0 = (Q_0, S_0) \in \mathbb{R}^2 \), an initial symmetric and positive definite matrix \( B_0 \in \mathbb{R}^{2 \times 2} \), \( 0 < \beta_1 < \beta_2 < 1 \), and \( \epsilon > 0 \).
2. Let \( g_k = (\frac{\partial G(Q,S)}{\partial Q}|_{Q=(Q_k, S_k)}, \frac{\partial G(Q,S)}{\partial S}|_{S=(Q_k, S_k)}) \). IF \( \|g_k\| \leq \epsilon \), stop; ELSE solve the linear equation \( B_k d_k + g_k = 0 \) to get \( d_k \).
3. Determine a stepsize \( \alpha_k > 0 \) by
\[
\begin{align*}
G(x_k + \alpha_k d_k) &\leq G(x_k) + \alpha_k g_k^T d_k, \\
G(x_k + \alpha_k d_k) &\geq G(x_k) + \alpha_k g_k^T d_k.
\end{align*}
\]
4. Let the next iterate be \( x_{k+1} = x_k + \alpha_k d_k \).
5. Determine \( B_{k+1} \) by
\[
B_{k+1} = \begin{cases} 
B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{\|y_k\|^2}, & \text{if } \frac{y_k^T s_k}{\|y_k\|} \geq \delta \|g_k\|, \\
B_k, & \text{otherwise},
\end{cases}
\]
where \( s_k = x_{k+1} - x_k \), \( y_k = g_{k+1} - g_k \), \( \delta > 0 \), and \( u > 0 \).
6. Let \( k = k + 1 \) and go to Step 2.

4. Numerical experiments

In this section, we carry out some numerical experiments to test the algorithms we proposed in Section 3. Our numerical experiments are run on a 1.60–2.11 GHz Intel i5-10210U processor laptop with 16 GB RAM. We use the \( k \)-stage Erlang distribution with mean \( \frac{1}{k} \) for \( f(\cdot) \) (note that when \( k = 1 \), the Erlang distribution degenerates into the exponential distribution). For the first proposed algorithm based on solving the system of linear equations (Algorithm 1), we conduct experiments for 16 different parameter settings. We compare the first two moments of \( O \) calculated by our method (Algorithm 1) with those obtained from the algorithm proposed in Zhu [20] (we call it Zhu algorithm) as well as simulation. For each parameter setting, the simulation results are obtained based on 500 million replications, and the errors of simulation results are smaller than 0.6%. Numerical results are presented in Table 1. \( N \) in Table 1 denotes the least number of linear equations for Algorithm 1 or coefficients of the MacLaurin series for Zhu algorithm to ensure the convergence of numerical values in the calculations. The time in Table 1 is measured in seconds. We can see from Table 1 that our method performs well with smaller errors in comparison with the simulation results, and slightly more precise than Zhu algorithm but with shorter time consuming on computation. Note that the computation time 0.000 s. However, either our method or Zhu algorithm diverges in 4 parameter settings. In practice, we need to check if the condition \( \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} |a_{ij}|^2 < \infty \) holds before applying Algorithm 1, and find its convergence domain. If we need to study the system outside the convergence domain, then we can use Zhu algorithm if it converges, otherwise, we can use simulation methods instead.

For the second proposed algorithm (Algorithm 2), we set \( t^R = 3, t^E = 1 \), and conduct experiments for 8 different parameter settings of \( c^R, c^E, h, b \), and distribution parameters. We compare the results calculated by our method (Algorithm 2) with those obtained from “brute-force” exhaustive search method. In the exhaustive search method, we use simulation to evaluate \( G(Q,S) \) over \( (Q,S) \in [0,20] \times [0,40] \) for \( \lambda = 0.05 \) and \( (Q,S) \in [0.5] \times [0.20] \) for \( \lambda = 0.25 \) with \( \Delta Q = \Delta S = 0.01 \), and then find \( (Q^*, S^*) \) that minimizes \( G(Q,S) \). For all experiments, optimal values of \( (Q,S) \) fall well within the search areas. In our method, we set \( N = 20 \) for 1-stage Erlang demands and \( N = 100 \) for 2-stage Erlang demands. As we mentioned in Section 3, any gradient-based algorithm can be used in our method, and we apply the CBFGS algorithm in the search area (the convergence radius of Algorithm 1) of \( (Q,S) \in (-\infty,12.4] \times (-\infty, +\infty) \) for \( \lambda = 0.05 \) and \( (Q,S) \in (-\infty, 2.5] \times (-\infty, +\infty) \) for \( \lambda = 0.25 \) to find optimal values of \( (Q,S) \). Numerical results are given in Table 2. We can see from Table 2 that our method obtains the results that are very close to the results by using the exhaustive search method,
Table 1. Numerical results for the first two moments of $O$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$\lambda$</th>
<th>$Q$</th>
<th>Algorithm 1</th>
<th>Zhu algorithm</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>2.5</td>
<td>0.1786</td>
<td>0.3614</td>
<td>0.3614</td>
<td>0.1786</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.8333</td>
<td>4.1667</td>
<td>4.1667</td>
<td>0.8333</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>5.0000</td>
<td>83.3333</td>
<td>83.3333</td>
<td>5.0000</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>0.25</td>
<td>2.5</td>
<td>0.0240</td>
<td>0.0318</td>
<td>0.0318</td>
<td>0.0240</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.1897</td>
<td>0.5645</td>
<td>0.5653</td>
<td>0.1897</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.7674</td>
<td>15.6592</td>
<td>15.6592</td>
<td>1.7674</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 2. Numerical results for the optimal values of $(Q, S)$ and $G(Q, S)$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>$(c^R, c^E, h, b)$</th>
<th>$\lambda$</th>
<th>Algorithm 2</th>
<th>Exhaustive search</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(Q^<em>, S^</em>)$</td>
<td>$G(Q^<em>, S^</em>)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(Q^<em>, S^</em>)$</td>
<td>$G(Q^<em>, S^</em>)$</td>
</tr>
<tr>
<td>1</td>
<td>(19, 20, 4, 5)</td>
<td>0.05</td>
<td>(12.01, 22.12)</td>
<td>486.9883</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.25</td>
<td>(2.40, 4.42)</td>
<td>97.3977</td>
</tr>
<tr>
<td>2</td>
<td>(19, 20, 5, 4)</td>
<td>0.05</td>
<td>(11.68, 15.57)</td>
<td>480.9321</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.25</td>
<td>(2.34, 3.11)</td>
<td>96.1864</td>
</tr>
<tr>
<td>1</td>
<td>(9, 10, 2, 4)</td>
<td>0.05</td>
<td>(11.24, 26.62)</td>
<td>278.8868</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.25</td>
<td>(2.25, 5.32)</td>
<td>55.7774</td>
</tr>
<tr>
<td>2</td>
<td>(9, 10, 4, 2)</td>
<td>0.05</td>
<td>(9.62, 2.67)</td>
<td>257.3969</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.25</td>
<td>(1.92, 0.53)</td>
<td>51.4794</td>
</tr>
</tbody>
</table>

and spends only 0.05 to a few seconds, which depends on the value of $N$, to obtain a result, whereas it usually takes many hours for the exhaustive search method to get a result.

5. Conclusion

In this paper, we propose two algorithms for dual sourcing inventory systems with tailored base-surge policies. We first develop an algorithm to calculate inventory positions, performance measures and their derivatives by solving a system of linear equations, and then use them in the gradient-based type of method to find the optimal tailored base-surge policy. Although there exists a convergence condition for calculating the moments of inventory positions, the proposed algorithms work well as long as it is within the convergence radius. Future research directions include improving the convergence of our method.
ACKNOWLEDGEMENTS

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 72271221, 71720107003, 72002201, 72301042), and Zhejiang Provincial Natural Science Foundation of China (Grant No. LY20G010012).

REFERENCES

Please help to maintain this journal in open access!

This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting subscribers@edpsciences.org.

More information, including a list of supporters and financial transparency reports, is available at https://edpsciences.org/en/subscribe-to-open-s2o.