Research Article

A New Separable Piecewise Linear Learning Algorithm for the Stochastic Empty Container Repositioning Problem

Shaorui Zhou, Xiaopo Zhuo, Zhiming Chen, and Yi Tao

1Department of Management Sciences, Sun Yat-Sen Business School, Sun Yat-Sen University, Guangzhou 510275, China
2Department of Credit Management, School of Credit Management Guangdong University of Finance, Guangzhou 510521, China
3School of Management, Guangdong University of Technology, Guangzhou 510006, China

Correspondence should be addressed to Xiaopo Zhuo; xiaopozhuo2019@126.com

Received 17 January 2020; Revised 6 June 2020; Accepted 6 July 2020; Published 28 August 2020

A common challenge faced by liner operators in practice is to effectively allocate empty containers now in a way that minimizes the expectation of costs and reduces inefficiencies in the future with uncertainty. To incorporate uncertainties in the operational model, we formulate a two-stage stochastic programming model for the stochastic empty container repositioning (ECR) problem. This paper proposes a separable piecewise linear learning algorithm (SPELL) to approximate the expected cost function. The core of SPELL involves learning steps that provide information for updating the expected cost function adaptively through a sequence of piecewise linear separable approximations. Moreover, SPELL can utilize the network structure of the ECR problem and does not require any information about the distribution of the uncertain parameters. For the two-stage stochastic programs, we prove the convergence of SPELL. Computational results show that SPELL performs well in terms of operating costs. When the scale of the problem is very large and the dimensionality of the problem is increased, SPELL continues to provide consistent performance very efficiently and exhibits excellent convergence performance.

1. Introduction

The acceleration of global economic growth and the expansion of global trade networks have led to an increasing demand for cargo transportation. The explosion of global trade has also been accompanied by the increasing use of containers as an economic and safe mode of transportation. A container that is fully loaded with cargoes from the shipper is transported to destination ports and delivered to the consignee and unloaded. Then, the container is emptied and stored at the destination port until it is booked for another consignment. However, when there is an imbalance in the number of export and import containers, some ports have a surplus of empty containers while others have a deficit. Under this situation, empty containers must be repositioned from surplus ports to deficit ports.

The trans-Pacific trade lane is taken as an example: according to UNCTAD [1], the annual container flow from Asia to North America (i.e., the eastbound) increased to 17.8 million twenty-foot equivalent units (TEUs), while that in the opposite westbound directions increased to 8.6 million TEUs, which generated the imbalance of container flow for 9.2 million TEUs in 2017. A similar situation has been reported for the container flows between Asia and Europe. Song [2] point out that the repositioning cost for empty containers is 27% of the global fleet running cost, and 20% of total container movements by ocean transportation are empty. Thus, empty container repositioning is a crucial issue for practitioners, and liner shipping companies must find an effective way to reposition their empty containers.

1.1. Motivation. Several planning models have been developed for liner companies to improve the efficiency of ECR [3–9]. Some research has examined deterministic situations [10, 11]. However, in the highly competitive shipping industry, operational-level planning requires decision-making with uncertainties. Although the shipping company knows
the exact demand according to the long-term contracts and makes delivery schedule based on demand information, liner operators face many uncertainties, such as port congestion, weather, and supply and demand fluctuation. These uncertainties might need liner operators to change the planned decision adaptively. Consequently, taking into account the uncertain nature of parameters in the ECR problem has attracted considerable attention from academics and practitioners.

This study is primarily motivated to solve the ECR problem faced by a major Chinese shipping company that we work with. Interview with the company reveals that, among the various uncertainties, the uncertainty of future demands, remaining vessel’s capacities, and supply are the most influential. Transportation and repositioning decisions in the ECR problem are considered wait-and-see decisions in practice. Given the dynamically changing environmental information in the shipping industry, the ECR decisions must be adjusted whenever new information is updated. Hence, we develop a two-stage stochastic model in the rolling horizon policy to deal with the dynamically changing demands, remaining vessel’s capacities, and supply in this study.

A well-known approach for dynamic decision-making is stochastic programming in which the uncertainty is characterized by a known probability distribution of parameters [7, 12, 13]. However, data forecasting accuracy, such as demand forecasting with historical data, is usually low and data estimation is difficult in practice. It is impossible to achieve complete knowledge about the distributions of uncertainties with historical data. Moreover, the stochastic ECR problem with a large number of scenarios is computationally intractable [7]. To overcome this limitation, we use a time-space network to illustrate the stochastic ECR problem and propose a separable piecewise linear learning algorithm (SPELL) to solve the problem. SPELL does not require any information about the distribution of uncertain parameters, and the learning steps in SPELL can provide information for updating the expected cost function adaptively by using sample information on the objective function itself. Furthermore, since our ECR problem has a network structure (time-space network), when we use a separable approximation and the approximation function is piecewise, we can solve the problem as a pure network flow problem, which is a well-known polynomial solvable problem. Therefore, the large-scale stochastic ECR problem can be efficiently solved, and we show the efficiency of our approach numerically in this study.

1.2. Literature Review. The ECR problem has attracted considerable attention in academia [14, 15]. Generally, these studies can be categorized into the strategic, tactical, and operational levels. At the strategic level, the decisions include price strategy and competition and encroachment and canvassing strategy [16, 17]. At the tactical level, most decisions concern the service network design problem [18–21], ship deployment, and fleet sizing [9, 11]. Threshold policies are also studied at this level [22, 23]. At the operational level, a highly dynamic environment is characterized, and most decisions focus on the transportation of empty containers among terminals/depots [5, 7]. This study aims to provide operational-level decisions for shipping companies.

Several researchers have considered the uncertain nature of parameters in ECR problems. For instance, Crainic et al. [3] introduced two dynamic formulations for the allocation of empty containers in single- and multicommodity cases given that both demand and supply are stochastic in nature. Cheung and Chen [24] proposed a two-stage stochastic network formulation of the ECR problem under uncertainties. This two-stage modeling successfully combines the deterministic and uncertain information of the problem and requires the approximation functions to be strongly convex. Erera et al. [12] and Shan et al. [21] developed a robust optimization framework for dynamic ECR problems that are modeled by using time-space networks. This robust approach modeled the uncertain supply and demand by using intervals regarding the nominal forecast value, and the repositioning plan can be adjusted under a set of recovery sections. This approach is consistent with the current repositioning operation and can be easily applied. Francesco et al. [13] proposed a stochastic programming model with uncertain data for ECR and solved the model using multiscenario optimization. However, the multiscenario model built by Francesco et al. [13] is subject to a small number of scenarios model. Long et al. [7] formulated a two-stage stochastic programming model for ECR and solved the program with the sample average approximation (SAA). They utilized the scenario aggregation to handle an extremely large number of scenarios. However, in these studies, the distribution of uncertain parameters is assumed to be estimated in advance. In this study, we will propose a separable piecewise linear learning algorithm which does not require any information about the distribution of uncertain parameters in advance, and this algorithm can solve large-scale ECR problems efficiently.

To incorporate uncertain parameters, we model the problem as a two-stage stochastic program. Studies have proposed various approximation schemes for two-stage stochastic programs that can be grouped into four categories: scenario methods that use fixed samples to approximate the underlying probability space [7, 25, 26]; stochastic gradient techniques that update the solutions by using stochastic subgradients as directions [27]; primal and dual decomposition methods [28, 29]; separable approximation methods [30–37] that replace the expected recourse function with separable approximation functions. The scenario method is very efficient, but the distribution of uncertain parameters is assumed to be estimated in advance, and its solution might not always converge to the optimal one. Meanwhile, the stochastic gradient technique has been proven to be convergent by projection [38], and the drawback is that the algorithm is inefficient. Given these characteristics, we combine the stochastic subgradient technique with the separable approximation method and propose the SPELL algorithm. SPELL does not need to know the distribution of the uncertain parameters in advance. Furthermore, SPELL can utilize the network structure of the
ECR problem and solve the problem as a pure network flow problem very efficiently. To our knowledge, for the maritime empty container repositioning problem, no existing study utilized the network structure of the ECR problem, and applied the separable piecewise linear learning algorithm to solve the large-scale stochastic ECR problem efficiently. This study is to fill in this gap for maritime transportation.

Among the above methods, the methods by Godfrey and Powell [32] and Cheung and Powell [30] are the most relevant to our study. Godfrey and Powell [32] proposed the adaptive piecewise concave approximation (CAVE) algorithm, which demonstrated exceptionally good experimental performance, but they did not provide any provable convergent results. To achieve convergence to optimal solutions, Cheung and Powell [30] proposed a convergent approximation algorithm (SHAPE), but it differs from our study in two main aspects. First, SHAPE uses a sequence of strongly convex approximation functions, while SPELL uses separable piecewise linear approximation functions. Since the ECR problem has a network structure, strong convexity requires a nonlinear term in approximation functions that might destroy the pure network structure and demands additional computational effort. Second, SHAPE may be stuck in a corner solution when the second stage is stochastic linear programs without the projection step. By contrast, we introduce projection steps [36, 37] to help SPELL to jump out from a local optimum and achieve a global optimum.

1.3. Contribution. The main aim of this study is to solve the real-time maritime ECR problem with uncertainties efficiently. In summary, contributions of this paper are fourfold. First, we propose a two-stage stochastic program model for the ECR problem with uncertainties. Since this model takes into account most port operational requirements and the actual shipping service schedule, it is very easily applied in the shipping industry. The stochastic model which considers some uncertain parameters may provide more robust decisions, and therefore the operational cost of ECR can be further reduced. Second, we propose the SPELL algorithm with limited information on distribution of uncertain parameters, and the learning steps in SPELL can provide information for updating the expected cost function adaptively through a sequence of piecewise linear separable approximation. Moreover, SPELL can utilize the network structure of the ECR problem and solve the large-scale problem very efficiently. Third, we theoretically prove the convergence of SPELL for two-stage stochastic programs. Finally, we conduct a performance analysis through the practical-scaled numerical experiments. The computation results reveal the efficiency of the proposed algorithm, which does not need to know the distribution of the uncertain parameters in advance.

The rest of this paper is organized as follows. Section 2 discusses the problem definition and mathematical models. Section 3 presents the SPELL algorithm for the two-stage stochastic ECR problem. Section 4 presents the performance analysis of SPELL through the computational experiments. Section 5 concludes the paper and presents directions for future research.

2. Problem Formulation

Our problem focuses on the operational level decisions of the ECR problem. Due to the global trade imbalance, the empty containers in surplus regions like North America have to be repositioned to deficit regions like Asia. One of the trans-Pacific shipping routes as shown in Figure 1 is taken as an example. The itinerary of this route forms a loop. We can arbitrarily deem that this itinerary starts and ends at Qingdao. Let \( p \in P \) represent the index of ports on a round trip for the route service \( r \in R \). Then, we can define Qingdao as Port 1, Ningbo as Port 2, Shanghai as Port 3, Busan as Port 4, New York as Port 5, Norfolk as Port 6, and Savannah as Port 7. Shippers usually pick up the required empty containers from the port \( (p = 1, 2, 3, \text{ or } 4) \), consolidate their cargoes into these containers, and then take the laden containers back to the port to wait for vessels. These laden containers are then loaded on a vessel and will be transported to the cargo receiver for unpacking at the destination port \( (p = 5, 6, \text{ or } 7) \). After unpacking, the empty container can either be moved/stored at the port \( (p = 5, 6, \text{ or } 7) \) for future reuse or be repositioned to another port \( (p = 1, 2, 3, \text{ or } 4) \) to meet the shippers’ demand. However, it usually takes a long time to deliver the across-region containers. For instance, the transit day from New York to the first arrived port in Asia (Qingdao) is 42 days. Thus, shipping companies should place their orders about 6 weeks in advance to satisfy the demand in Asia.

In practice, whether the demand orders in the deficit region could be satisfied mainly depends on the empty container inventory stored at ports in the surplus region as well as the remaining capacities of vessels. Moreover, due to the long lead time of empty containers, liner operators usually make ordering decisions based on forecasting with historical data. However, it is usually difficult to obtain accurate forecasting for more than two or three weeks. Therefore, liner operators will be hard pressed to make operational level decisions for ECR, and inefficient ordering decisions might be made owing to low forecasting accuracy. To overcome this limitation, we propose a stochastic programming model that considers three major sources of randomness in the ECR problem, namely, the demands at ports, the supplies at ports, and the remaining capacities of vessels for empty containers. This model does not need to know the distribution of the uncertain parameters in advance. Other uncertain factors, such as the transportation time between two ports, are not considered.

Without loss of generality, we make several assumptions in our model. First, we assume that the shipping service schedule is fixed and given in the planning horizon. This assumption is valid given that the planning horizon of our model is short (several weeks) and that the service schedule is not changing frequently. Second, after laden containers are unloaded from vessels at their destination ports, they become empty and can be reused or repositioned to other ports. Third, when the demands cannot be satisfied by the empty container inventory at surplus ports, we have to lease,
buy, or temporarily use containers provided by other shipping companies, all of which involve a penalty cost [7]. Note that we do not make decisions on laden container transportation in this study, and the same assumption is common in the literature (e.g., [7, 40]). Unlike empty containers, laden containers have fixed origins and destinations. Furthermore, the laden container transportation and ECR problems are usually considered separately in the shipping industry. Therefore, we do not treat the number of laden containers to be shipped from one port to another port as a decision variable; instead, we assume that this number is given. Our model makes ECR decisions after the laden container transportation is planned.

Our problem focuses on short-term decision-making while considering the influence of these decisions in the planning horizon. To illustrate the stochastic ECR problem, let us consider a time-space network of the ECR problem as in Figure 2. This network has been divided into two parts: in stage 1, all parameters are deterministic and known, while some parameters in stage 2 are unknown when a decision is made in stage 1. The repositioning decision made in stage 1 determines the initial state when solving the problem in stage 2. S is a cross-region service. In this study, we consider each region separately. ECR decisions related to ports in the target region are made, while the coordination with other regions is not considered. We only consider the information in the target region when decisions on the number of empty containers into/out of the target region are made. To make the two-stage stochastic ECR model easy to understand, we first introduce the following deterministic ECR model.

2.1. Deterministic ECR Model. We first assume that all parameters in the planning horizon are known and then formulate a deterministic ECR problem. Then, we incorporate uncertain parameters into our model and propose a two-stage stochastic ECR model for the stochastic ECR problem. Table 1 presents the notations.

We consider several cost elements for the ECR problem, including empty container lifting-on cost \(C_0\), empty container lifting-off cost \(C_1\), inventory costs of empty container \(C_2\), empty container transportation cost \(C_3\), and empty container penalty cost \(C_4\), which can be formulated as follows:

\[
\begin{align*}
C_0 &= \sum_{t=1}^{T} \sum_{p \in P} \sum_{s \in S} \sum_{r \in R} f_{p,v} \frac{x_{r,v}^{h,t}}{1}, \\
C_1 &= \sum_{t=1}^{T} \sum_{p \in P} \sum_{s \in S} \sum_{r \in R} u_{p,v} x_{r,v}^{l,t}, \\
C_2 &= \sum_{t=1}^{T} \sum_{p \in P} \sum_{s \in S} \sum_{r \in R} h_{p,v} x_{r,v}^{l,t}, \\
C_3 &= \sum_{t=1}^{T} \sum_{p \in P} \sum_{s \in S} \sum_{r \in R} \alpha_{p,v} x_{r,v}^{l,t}, \\
C_4 &= \sum_{t=1}^{T} \sum_{p \in P} \sum_{s \in S} \sum_{r \in R} \beta_{p,v} x_{r,v}^{l,t}.
\end{align*}
\]

Afterward, we formulate the deterministic ECR problem \((D_1)\) as

\[
D_1: \min C = C_0 + C_1 + C_2 + C_3 + C_4, \tag{2}
\]

subject to

\[
\begin{align*}
\lambda_{r,v}^{c,s}(t-t') - \lambda_{r,v}^{u,s+1})(t-t') + \lambda_{r,v}^{l,s+1})(t-t') &= \lambda_{r,v}^{c,s}(s+1)(t-t') + \lambda_{r,v}^{l,s+1})(t+1-t'), \\
\forall v \in V, \forall (r,s,t) \in \{(r,s,t)|r \in R, s \in P, t \in B_v^t\}, \\
\lambda_{r,v}^{h}(t-1) + \sum_{(s,r) \in \{(s,r)|r \in R, s \in P, r \in P\}} \lambda_{s,v}^{u,s+1} x_{s,v}^{l,t} + \lambda_{p,v}^{w} + \alpha_{p,v}^{l,t} - \beta_{p,v}^{l,t} &= \lambda_{r,v}^{h}(t-1), \\
\forall v \in V, \forall p \in P, \forall t = 1, 2, \ldots, T,
\end{align*}
\]

Figure 1: A transpacific shipping service route operated by the shipping liner COSCO. Source: COSCO [39].
Table 1: Notations for the ECR problem.

| Indices and sets | Description |
|------------------|-------------|
| $p$              | The index of ports |
| $P$              | The set of ports |
| $r$              | The service route |
| $R$              | The set of service route |
| $v$              | The container size |
| $V$              | The set of container size |
| $P_r$            | The set of ports on service route $r$ |
| $p_s^r$          | The port corresponding to the stop $s$ on service route $r$ |

| Input parameters | Description |
|------------------|-------------|
| $b_r^s$          | The beginning time when service $r$ arrives at its stop $s$ |
| $B_r^s$          | The set of beginning time when service $r$ arrives at its stop $s$ |
| $c_r^s$          | The ending time when service $r$ arrives at its stop $s$ |
| $E_r^s$          | The set of ending time when service $r$ arrives at its stop $s$ |
| $t_r^s$          | The transportation time from stop $s$ to the next stop on service route $r$ |
| $d_r^s$          | The number of days that the service $r$ stays at stop $s$ |
| $cap_v$          | The capacity of one container of size $v$ (capacity here means volume and weight) |
| $α_{p,v}^t$      | The quantity of the supply of empty containers of size $v$ at port $p$ at time $t$ |
| $β_{p,v}^t$      | The quantity of the demand of empty containers of size $v$ at port $p$ at time $t$ |
| $c_r^s,t$        | The remaining capacity for empty containers on service route $r$ when leaving stop $s$ at time $t$ (laden container excluded) |
| $u_{p,v}^t$      | The cost of unloading one container with size $v$ at port $p$ at time $t$ |
| $h_{p,v}^t$      | The cost of loading one container with size $v$ at port $p$ at time $t$ |
| $p_{p,v}^t$      | Per period cost of storing one container with size $v$ at port $p$ at time $t$ |
| $w_{p,v}^t$      | The penalty cost of one container with size $v$ at port $p$ at time $t$ |

| Decision variables | Description |
|--------------------|-------------|
| $x_{r,s,t}^{c,s,t}$ | The number of empty containers of size $v$ unloaded at stop $s$ on service route $r$ at time $t$ |
| $x_{r,s,t}^{c,s,t}$ | The number of empty containers with size $v$ loaded at stop $s$ on service route $r$ at time $t$ |
| $x_{r,s,t}^{c,s,t}$ | The number of empty containers with size $v$ stored at port $p$ at time $t$ |
| $x_{r,s,t}^{c,s,t}$ | The number of empty containers with size $v$ that cannot be satisfied by the empty containers at port $p$ at time $t$ |
| $x_{r,s,t}^{c,s,t}$ | The number of empty containers with size $v$ transported from stop $s$ to the next stop on service route $r$ at time $t$ |

\[
\sum_{v \in V} (cap_v \times x_{r,s,t}^{c,s,t}) \leq y_r^s, l
\]

\[
\forall (r,s,t) \in \{(r,s,t)| r \in R, s \in P_r, t \in B_r^{s+1}\},
\]

\[
x_{r,s,t}^{c,s,t} = \alpha_{p,v}^t - \beta_{p,v}^t
\]

\[
\forall v \in V, \forall (r,s,t) \in \{(r,s,t)| r \in R, s \in P_r, t \in B_r^s\},
\]

\[
x_{r,s,t}^{c,s,t} \geq x_{r,s,t}^{u,s+1,t}
\]

\[
\forall v \in V, \forall (r,s,t) \in \{(r,s,t)| r \in R, s \in P_r, t \in B_r^s\},
\]
\( x_{r,s}^{\mu,t}, x_{r,s}^{\nu,t}, x_{p,r}^{\nu,t}, x_{p,r}^{\nu,t}, x_{r,s}^{\nu,t} \geq 0, \)

\( \forall v \in V, \forall p \in P, \forall t = 1, 2, \ldots, T, \)

\( \forall (r, s, t) \in \{(r, s, t) | r \in R, s \in P, t \in B_{r}^{t}\}. \)

Constraint (3) denotes the service flow constraint that considers the balance of container flows for each service route. Constraint (4) is the port flow constraint that guarantees the balance of container flows at each port for each time. Constraint (5) is the capacity constraint. Constraint (6) guarantees that the number of empty containers unloaded from a vessel is less than the number of empty containers on this vessel. Constraint (7) guarantees that all the decision variables should be nonnegative.

2.2. Two-Stage Stochastic ECR Model. To incorporate uncertain information, we develop a two-stage stochastic ECR problem. In stage 1, all information is known and all parameters are deterministic, while in stage 2, some parameters (supply \( x_{p,r}^{\nu,t} (\omega) \), demand \( p_{p,r}^{\nu,t} (\omega) \), and remaining capacities of vessels, \( y_{r,s}^{\nu,t} (\omega) \)) are uncertain when decisions are made in stage 1 [7]. At the beginning of stage 1, we refer the first week as stage 1 in this study, liner operators make repositioning decisions in the target region and across regions. The decisions will be made again when new information is updated later. Let \( \omega \in \Omega \) denote a realization of the unknown parameters when decisions at stage 1 are made, but these parameters are known when the second stage decisions are made.

We then develop the following two-stage stochastic model \((S_1)\):

\[
\begin{align*}
S_1: \min f(x_1) &= c_1^T x_1 + E_{\omega}[Q(x_1, \omega)], \\
\text{s.t.} & \quad A_1^1 x_1 = b_1^1, \quad i \in \mathbb{N}_1^1 \\
& \quad B_1^1 x_1 = s_i, \quad i \in \mathbb{N}_1^1, \\
& \quad x_1 \geq 0,
\end{align*}
\]

where given a realization \( \omega \) and a set of in-bound flows to stage 2, the optimal value of the stage 2 problem \( Q(x_1, \omega) \) is given by

\[
Q(x_1, \omega) = \min c_2^T x_2 (\omega),
\]

\[
\text{s.t.} \quad A_2^1 (\omega) x_2 (\omega) = b_2^1 (\omega), \quad i \in \mathbb{N}_2^1, \\
B_2^1 (\omega) x_2 (\omega) = s_i (x_1), \quad i \in \mathbb{N}_1^1, \\
x_2 (\omega) \geq 0.
\]

In the above model, equations (8) to (11) define the stage 1 problem, and the objective function of stage 1 is to minimize the total operational cost. \( c_1^T x_1 \) is the operational cost of stage 1, and \( E_{\omega}[Q(x_1, \omega)] \) is the expected operational cost of stage 2. Equations (12) to (15) define the stage 2 problem. The decision variables \( x_1 \) and \( x_2 \) denote the decision variables of the stages 1 and 2 problems, respectively, while \( c_1 \) and \( c_2 \) denote the stages 1 and 2 vectors of cost coefficients, respectively. In addition, \( A_1^1, B_1^1 \) and \( b_1^1 \) denote the constraint matrices for stage 1, while \( A_2^1 (\omega), B_2^1 (\omega) \), and \( b_2^1 (\omega) \) denote the uncertain parameters for stage 2 constraint metrics. \( s_i \) denotes the container state of stage 1. It is the empty container inventory at each port and at each vessel at the end of stage 1. \( s_i (x_1) \) denotes the vector of the initial empty container state of stage 2 for a given \( x_1 \). Decisions \( x_1 \) in the stage 1 are communicated to the stage 2 through the state variable \( s_i \). Constraints (9) and (11)\(\text{d}1\) include the constraints of the deterministic ECR problem \((D_1)\), the ship capacity constraint, the service route flow balance constraint, and the port flow balance constraint. Constraint (10) sets the empty container state at the end of stage 1 and shows the inventory at each port and vessel at the end of stage 1. Constraint (14) sets the initial container state of stage 2. For a given \( x_1 \), the initial container state \( (s_i (x_1)) \) in stage 2 must be equal to the ending container state in stage 1 \((s_i)\).

3. Solving Methodology

The main challenge for solving two-stage stochastic programs in most practical cases is that the expected recourse function \( E_{\omega}[Q(x_1, \omega)] \) is excessively complex due to uncertainty. In this section, a new separable piecewise learning algorithm (SPELL) is proposed to approximate the expected cost function of the ECR problem.

Let \( Q(x_1) \) denote the approximation function. Essentially, SPELL is a combination of the stochastic subgradient technique [27] and the separable approximation method [24]. The stochastic subgradient technique updates the solutions of problem \((S_1)\) by using stochastic subgradients as directions [27]. These subgradients can be obtained from the scenario \( (\omega) \) at each iteration, and the approximate function can be updated successively as follows:

\[
\bar{Q}(x_1) = (g^{-k})^T x_1,
\]

where \( g^{-k} \) is a smoothed estimate of the gradient of the expected cost function at iteration \( k \). For this method, convergence is achieved by projection [38] although the disadvantage is that the method is not efficient. Meanwhile, the separable approximation method replaces the expected recourse function with separable approximation functions as follows [24]:

\[
\bar{Q}(x_1) = \sum_{i=1}^{I} Q_i(x_i).
\]

Note that our ECR problem has a network structure. If the separable function \( Q_i(x_i) \) is linear or piecewise linear, and the expected recourse function in equation (8) can be
exploring the approximation function which is a well-known polynomial solvable problem.

According to their advantage and disadvantage mentioned above, we combine equations (8), (16), and (17) to form an approximation of the expected operational cost function at iteration $k$ as follows:

$$\min f(x_1) = c_1^T x + \hat{Q}^0(x_1) + (\hat{g})^T x_1.$$

(18)

The basic idea is that we would like to approximate the expected cost function of the stochastic ECR problem ($S_1$) via an initial convex approximation $\hat{Q}^k(x_1)$ with an iteratively updated linear correction term $(\hat{g})^T x_1$. The information from the linear correction term $(\hat{g})^T x_1$ generated at iteration $k$ can improve the initial approximation $\hat{Q}^0(x_1)$. Note that we use a convex initial approximation function $\hat{Q}^k(x_1)$, whereas the SHAPE algorithm by Cheung and Powell [30] uses strongly convex approximation functions. Although the strong convexity can guarantee the convergence of SHAPE, the strong convexity requires a nonlinear term in the approximation function that may destroy the pure network flow problem structure of our ECR problem and demands more computational effort. But, when the second stage is stochastic linear programs, the SHAPE may be stuck in a corner solution. In order to jump out of the local optimum, the projection step that implement the stochastic subgradient method is introduced in SPELL. Our computational results shows the effectiveness and efficiency of the projection step, and our analytical results in the appendix also prove the convergence of SPELL.

In this study, the stochastic subgradient $\hat{g}$ is not calculated in the usual manner. As shown later in Proposition 1, $\hat{g}$ can learn the information from the previous $(k-1)$ iterations and update itself adaptively. In contrast, our method uses the following approximation form:

$$\min f(x_1) = c_1^T x_1 + \hat{Q}^k(x_1) + a_k(g^k - q^k)^T x_1,$$

(19)

where $g^k$ is a gradient of $\hat{Q}^k(x_1)$ at iteration $k$, $g^k$ is stochastic subgradient of $\hat{Q}(x_1, \omega)$ at $x_1$, that is, $g^k \in Q(x_1, S_1, g^k)$, and $a_k$ is the step size and is $H_k$-measurable. Then, $\hat{Q}^k(x_1)$ is updated as

$$\hat{Q}^{k+1}(x_1) = \hat{Q}^k(x_1) + a_k(g^k - q^k)^T x_1.$$

(20)

The advantage of updating $\hat{Q}^{k+1}(x_1)$ is that the stochastic subgradients used in the previous iterations, namely, $(g^1, g^2, \ldots, g^k)$, can be retained. Thus, the objective function in iteration $k$ involves a weighted average of stochastic subgradient in previous $(k-1)$ iterations. We call this a learning step because in each iteration, we are exploring the approximation function $\hat{Q}^k(x_1)$ at previous $(k-1)$ randomly sampled scenarios. We discuss the learning step in the following proposition.

**Proposition 1.** The approximation function $\hat{Q}^k(x_1)$ at iteration $k$ can be written as $\hat{Q}^k(x_1) = \hat{Q}^0(x_1) + (\hat{g})^T x_1$, where $\hat{g}$ is a weight average of the stochastic subgradient in previous $(k-1)$ iterations and $\hat{Q}^0(x_1)$ is piecewise linear.

**Proof.** Based on equation (18), the approximation of the expected cost function at iteration $k$ is the initial approximation $\hat{Q}^0(x_1)$ plus a linear term as follows:

$$\hat{Q}^k(x_1) = \hat{Q}^0(x_1) + (\hat{g})^T x_1.$$

(21)

where $(\hat{g})^T x_1$ is the cumulative change of $\hat{Q}^0(x_1)$ up to iteration $k$. Taking the first derivation of $\hat{Q}^k(x_1)$ in equation (21), we have

$$\hat{q}^k(x_1) = \hat{q}^0(x_1) + (\hat{g})^T x_1.$$  

(22)

Based on equations (20) and (22), we can obtain $\hat{Q}^{k+1}(x_1)$ as follows:

$$\hat{Q}^{k+1}(x_1) = \hat{Q}^k(x_1) + a_k(g^k - \hat{q}^k(x_1))^T x_1$$

$$= \hat{Q}^0(x_1) + (\hat{g})^T x_1 + a_k(g^k - \hat{q}^k(x_1)) x_1$$

$$= \hat{Q}^0(x_1) + (\hat{g})^T x_1 + a_k(g^k - q^0(x_1) - \hat{g})^T x_1.$$  

(23)

Thus, we can obtain the relationship between $\hat{g}^{k+1}$ and $\hat{g}^k$ as follows:

$$\hat{g}^{k+1} = (1 - a_k)\hat{g}^k + a_k(g^k - q^0(x_1)).$$

(24)

From equation (24), we can conclude that $\hat{g}^{k+1}$ is a linear combination of $g^1, g^2, \ldots, g^k$ and is piecewise linear. Thus, $\hat{g}^{k+1}$ can be updated by the information learnt from the previous $k$ samples.

The basic idea of SPELL is that at iteration $k$, we compare the slopes of gradient $\hat{Q}^k(x_1)$ with the stochastic subgradient $g^k$. Afterward, we use the difference of these two slopes to update $\hat{Q}^k(x_1)$ linearly. When the solution $x_1^k$ is stuck in a corner solution, we implement the projection step to jump out of the local optimum. Let $P_{X_1}: R^n \rightarrow X$ be the orthogonal projection onto $X$ [38]. We apply the algorithm in Figure 3 to find a sequence of solutions $\{x_1^k\}$.

Generally, SPELL maintains two-level loops as shown in Figure 3. In the first-level loop, there are a series of update processes, in which the approximation function can be updated iteratively. However, if the solution $x_1^{k+1}$ is stuck in a corner solution and the local optimum detector is triggered, then SPELL will enter the second-level loop, which contains a series of projection steps. In the first update process, we construct an initial convex approximation function $\hat{Q}^0(x_1)$ and obtain the initial solution $x_1^0$. We then sample a realization of random parameters (supply $\alpha^f, \omega$, demand $\beta^g$, $\omega$, and remaining capacity of vessels $\omega$ such that $\omega \in \Omega$) solve the corresponding deterministic problem, and calculate the stochastic subgradient of the expected cost function, $g^k$. Subsequently, the approximation function can be updated by learning the information from the difference between $g^k$ and $q^k(x_1^k)$, and we can obtain a new solution $x_1^{k+1}$ in Step 4. However, if the newly obtained solution $x_1^{k+1}$ has an exactly identical subgradient vector $\hat{g}^{k+1}(x_1^{k+1})$ as a solution $x_1^0$ which means the newly obtained solution $x_1^{k+1}$ triggers the local optimum detector, then SPELL will enter in Step 6 and implement a series of projection steps in the second-level loop to jump out of the local optimum. Without
Step 1. Set the iteration counter \( k = 0 \) and pass counter \( m = 0 \). Construct an initial convex function \( Q_0(x_1) \). Maintain a sequence of points \( \{x^t\} \). Let \( m \) be the number of the first iteration of the \( m \) pass.

**Figure 3:** The flowchart of the SPELL algorithm.

these projection steps, if \( x_1^{k+1} \neq x_1^{m} \) and \( |\hat{m}^{k}(x_1^{k+1}) - \hat{m}^{m}(x_1^{m})| = 0 \), the SPELL algorithm proceeds directly to Step 3, and the approximation function \( \hat{Q}^{k+1}(x_1) \) will be not updated in Step 3. Thus, the SPELL algorithm might become stuck in the corner solution. On the other hand, if \( \hat{m}^{m}(x_1^{k+1}) \) is different from \( \hat{m}^{m}(x_1^{m}) \), SPELL will not enter into the projection step and proceed directly to the next update process, and the entire update process is repeated. Note that, when SPELL triggers the local optimum detector and enters into the projection step, it might need several iterations to jump out. Eventually, the algorithm will be terminated when the total absolute change in \( Q(x_1) \) over a certain number of iteration is low (e.g., \( \sum_{k=M+1}^{K} \| \hat{Q}^{k}(x_1) - \hat{Q}^{k-1}(x_1) \| < \delta \)). In the above SPELL algorithm, \( k \) represents the iteration number, and it updates when SPELL starts a new iteration to obtain a new solution \( x_1^{k+1} \) (in Steps 4 and 6); \( m \) represents the number of SPELL crossing the projection steps, and it updates when SPELL satisfies the condition in Step 5; \( M \) represents the number of the first iteration of the \( m \) update process, and it updates when SPELL enters a new update process in Step 3.

Differences between Shape and Spell. The most remarkable difference between SHAPE and SPELL is that SHAPE uses strongly convex approximation functions, whereas SPELL uses separable piecewise linear convex approximation functions. Strong convexity requires a nonlinear term in the approximation function that may destroy the pure network structure of the ECR problem and demand additional computational effort. To overcome the limitation of the SHAPE algorithm, the SPELL algorithm introduces the projection step to construct approximation functions, which are not strictly convex. As shown in the numerical experiment later, without the projection step, the SPELL algorithm might also become stuck in the corner solution for the ECR problem. Furthermore, our analytical results in the appendix prove the convergence of SPELL.

Since SPELL adjusts the approximation by a linear term at each iteration, the following property is obtained.

**Proposition 2.** For any two points \( x_1^i \) and \( x_1^j \), if \( \hat{m}^0(x_1^i) = \hat{m}^0(x_1^j) \), then \( \hat{m}^k(x_1^i) = \hat{m}^k(x_1^j) \) for any \( k > 1 \).

The proposition holds because the approximation functions are updated by linear terms.

As shown in Figure 3, if the newly obtained solution \( x_1^{k+1} \) in Step 4 has the exactly identical subgradient vector \( \hat{m}^{m}(x_1^{k+1}) \) as solution \( x_1^{m} \), then according to Proposition 2, we can conclude that there will be no update in \( \hat{Q}^{k+1}(x_1) \) in the
next iteration. Then, SPELL may be stuck into the corner solutions. Therefore, we need the projection step in Step 6 to help SPELL jump out from a local optimum.

**Proposition 3.** For any two iterations in the $m$th update process, say $x_1$ and $x_1'$, $\tilde{q}^k(x_1) = \tilde{q}^k(x_1') = \tilde{q}^{m}(x_1')$ for any $m \leq k \leq m + T$.

The proposition holds because the approximation functions are not updated when SPELL performs the projection steps. Thus, if the newly obtained solution $x_{k+1}$ produced by Step 6 can pass the locally optimum detector, then it can help SPELL to jump out from a local optimum and proceed to the next update process.

### 4. Computational Experiment

We conduct extensive computational experiments based on a real-world shipping service network to evaluate the performance of SPELL. All computational experiments are implemented on a computer with an Intel (R) Core (TM) i7-7567 CPU at 3.5 GHz and 16 GB of RAM with Java and IBM ILOG CPLEX 12.6. We initially introduce the practical ECR problem and present the effectiveness and efficiency performance of SPELL. Afterward, we demonstrate the performance of SPELL under various distributions. Finally, we present the convergence performance of SPELL.

#### 4.1. Test Case

To evaluate the performance of SPELL, we generated three different scale test instances with various numbers of ports and service routes. In this study, we selected the real-world shipping service routes operated by the shipping liner COSCO [39] to generate three different scale test instances with various network parameters. Table 2 lists the network parameters for three problem instances. To evaluate the performance of SPELL, we conduct extensive computational experiments based on a real-world shipping service network to evaluate the performance of SPELL. All computational experiments are implemented on a computer with an Intel (R) Core (TM) i7-7567 CPU at 3.5 GHz and 16 GB of RAM with Java and IBM ILOG CPLEX 12.6. We initially introduce the practical ECR problem and present the effectiveness and efficiency performance of SPELL. Afterward, we demonstrate the performance of SPELL under various distributions. Finally, we present the convergence performance of SPELL.

#### 4.2. Effectiveness and Efficiency Performance Analysis

For the stochastic ECR problem, obtaining optimal solutions is generally intractable. Therefore, it is impossible to compare the solution produced by a method with the optimal solution. One possible way is to develop some bounds on the optimal solution and then use these bounds to evaluate the quality of a solution. In this study, we use PB (posterior bound), a myopic algorithm, and the L-shaped algorithm [41] as benchmarks to test the effectiveness and efficiency of SPELL. To obtain PB, we need to solve a deterministic network flow problem with all realized parameters. Such a posterior optimization involves no uncertainty because decisions with anticipation of future demands are allowed. Therefore, the cost of PB is low and normally unreachable, and it is used as the lower bound [42]. The myopic algorithm simply solves a deterministic problem at the current stage while ignoring uncertainties in the second stage. Thus, we use the results produced by the myopic algorithm as the upper bound. For the L-shaped algorithm, it is very efficient, a group a linear programming problems with a series of cutting planes must be solved. And, we can use the results produced by the L-shaped algorithm to test the efficiency of uncertain parameters are replaced by the sample mean values. Then, we can obtain $\tilde{Q}_i^0(x)$ by

\[
\tilde{Q}_i^0(x) = c(x - \bar{x})^2, \quad x = 0, \delta, \ldots, k\delta, \ldots, k\delta,
\]

where $c$ is a positive parameter and $x \in [0, K\delta]$. Note that $x$ is an integer in the ECR problem. We use $\delta$ here to illustrate the granularity of the initial function. For the projection, we calculate the following least-square problem:

\[
x^{k+1} = \arg \min \left[ x^{k+1} - \left( x^k + \alpha_k g^k \right) \right]^2, \quad x^{k+1} \in X.
\]
SPELL. To evaluate the effect of the projection step, we also implement SPELL without the projection step, which is called SPELL-NON, in the following section.

In the experiment, 2000 samples have been run for each instance to obtain the sample means of PB. The solutions of the myopic algorithm, L-shaped algorithm, SHAPE, SPELL, and SPELL-NON can also be obtained. In this study, we select the initial functions with $\delta = 2$.

The results on the total operational cost are listed in Table 3. Column 1 shows the problem instance. Columns 2 to 5 record the solutions obtained by PB, myopic, L-shaped, SPELL-NON, and SPELL. The experiment results demonstrate that SPELL can achieve near-optimal solutions, which are extremely close to the PB (within 0.11% gap from PB). The solutions of SPELL, SPELL-NON, and L-shaped algorithm are better than those of the myopic algorithm. For the small-scale instance, the operational cost obtained by SPELL is 2.67% lower than that of the myopic algorithm. For the large-scale instance, the operational cost obtained by SPELL is 7.57% lower than that of the myopic algorithm. Therefore, compared with the deterministic model, our stochastic model can reduce the operational cost of the ECR problem. Notably, the larger is the instance, the relative improvement of the stochastic model is more significant. Furthermore, SPELL slightly outperforms SPELL-NON because the projection steps in SPELL can help jumping out of local optimum. Hence, the projection step is effective in SPELL. In most practical-scaled stochastic ECR problems, the speed of calculation is one of the major concerns of practitioners, and thus we list the computation time for different algorithms in Table 4.

As shown in Table 4, SPELL is more efficient than the L-shaped algorithm because SPELL can utilize the network structure of the ECR problem while approximating the objective function. The larger is the scale of the instance, the more efficient SPELL is. The L-shaped algorithm is time consuming because 2000 samples are used in this experiment, which corresponds to a large number of cuts for the L-shaped algorithm. Compared with SPELL-NON, SPELL is time consuming because the projection step in SPELL is effective and the calculation of the projection requires more computation time.

4.3. Performance under Various Distributions. In this section, we aim to examine whether the distribution of the uncertain parameters (supply $\alpha_{p,v}^i(\omega)$, demand $\beta_{p,v}^i(\omega)$, and remaining capacities of vessels, $\gamma_{p,v}^i(\omega)$) affects the performance of SPELL or not. Since the normal distribution has been implemented in Section 4.2, we consider two other distributions in this section: the exponential and uniform distributions. The results on the total operational cost for the exponential and uniform distributions are presented in Tables 5 and 6, respectively, and the computation time for the different algorithms is presented in Table 7.

As shown in Tables 5 and 6, the performance under exponential and uniform distributions is consistent with those of uniform distribution in Section 4.2. SPELL can achieve near-optimal solutions. According to Table 7, SPELL and SPELL-NON are also more efficient than the L-shaped algorithm. Moreover, the projection steps are effective because SPELL-NON requires more computation time than SPELL. Thus, the results are consistent with the fact that SPELL do not require any information on the distribution of the uncertain parameters.

4.4. Convergence Performance Analysis. From the perspective of liner operators in the shipping industry, one of the major concerns in solving the practical-scaled ECR problem is the rate of convergence. Thus, in this section, we will evaluate the convergence rate of SPELL. To measure the rate of convergence of different methods, we implement each algorithm for 40, 160, 640, 1200, and 4000 iterations and compare the solutions of algorithms side by side as the number of iterations increases. For our SPELL and myopic algorithms, the number of iterations refers to the number of samples used, whereas the number of iterations refers to the numbers of cuts for the L-shaped algorithm.

The results are summarized in Table 8. The figures in the table record the deviation from PB after a certain number of iterations. Table 8 also lists the computation time per iteration. We present the experimental results on the three-scale problem instances, which vary in the dimensionality of the recourse function.

As shown in Table 8, column 2 lists the methods used in the experiment. Columns 3 to 7 contain the percent deviation from the PB, and column 8 presents the computation time per iteration. The results demonstrate that SPELL can produce high-quality solutions rather quickly and achieve consistent performance under different problem instances. In particular, when the scale of the problem is very large and the dimensionality of the problem is increased, SPELL continues to provide consistent performance with a short computation time per iteration. This feature makes SPELL...
attractive for liner operators in solving the practical-scaled ECR problem. Compared with the L-shaped algorithm (bender decomposition-based algorithm), SPELL is more competitive for high-dimensional problems because separable approximations typically scale much more easily to the very high-dimensional problem. Accordingly, SPELL is a promising method for the stochastic ECR problem. Due to its simplicity and fast runtime, SPELL is an attractive candidate for problems where the computational cost for one experiment is high.

5. Conclusion

In order to incorporate uncertainties in the ECR problem, we build a two-stage stochastic model with uncertain demand, supply, and ship capacity, and this model could be easily applied to the shipping industry. To solve the model, we propose a separable piecewise linear learning algorithm which does not require any information about the distribution of uncertain parameters, and the learning steps can provide information for updating the expected cost function adaptively by using sample information on the objective function itself. Moreover, the proposed algorithm can utilize the network structure of the ECR problem and can solve the large-scale stochastic ECR problem very efficiently. To evaluate the performance of the algorithm, we solve three scale problems which are based on three real-world shipping service networks operated by the shipping company COSCO. Numerical experiments indicate that SPELL is efficient and effective. Moreover, the proposed method is found to be very competitive for high-dimensional ECR problems. In order to achieve a global optimum, a projection step that implements the stochastic subgradient method is performed to jump out from a local optimum, and the computational result shows the effectiveness of the projection step in SPELL. As the two-stage model lives with the assumption that we know the future at the end of the first stage, it is unable to handle sequential decisions. We may extend the two-stage problem to the multistage problem which does not require such an assumption to solve the maritime ECR problem in our future study.
Appendix

A. Convergence Theorem of SPELL

In this appendix, we state the convergence of the theorem of SPELL. Subsequently, we list several properties of the approximation that are used to prove the convergence of SPELL. Finally, we present the proof for our theorem.

Without loss of generality, we make the following assumptions:

(A.1) $X \subset \mathcal{R}^n$ is convex and compact
(A.2) $E_q Q(x_1, \omega)$ is convex, finite, and continuous on $X$
(A.3) $g^k$ is bounded such that $\|g^k\| \leq c_1$ for each $\omega \in \Omega$; $\tilde{q}^k$ is bounded such that $\|\tilde{q}^k\| \leq c_2$ for each $\omega \in \Omega$
(A.4) Piecewise linear function $\tilde{Q}^k (x)$ is convex, implying that

$$\tilde{Q}^k (x_1) - \tilde{Q}^k (y_1) \leq \tilde{q}^k (x_1^T - y_1).$$

(A.5) The step size $\alpha_k$ is $H_k$-measurable and satisfies

$$\sum_{k=0}^{\infty} E[\alpha_k^2] < \infty, \quad 0 < \alpha_k < 1.$$  \hspace{1cm} (A.2)

In addition to the assumption from (A.1) to (A.5), the following assumption is required to characterize the piecewise linear convex approximation functions.

(A.6) There exists a constant $\delta$ and a positive $b$, such that for any two points $x_1, y_1 \in X$, if $|x_1 - y_1| > \delta$, then $|\tilde{q}(x_1) - \tilde{q}(y_1)| \geq b|x_1 - y_1|$. If there exists $\tilde{q}(x_1)$ and $\tilde{q}(y_1)$ such that $\tilde{q}^k (x_1) - \tilde{q}^k (y_1) = 0$, then $|x_1 - y_1| \leq \delta$. If $\delta \to \infty$, then the function becomes purely linear; if $\delta \to 0$, then the function corresponds to a strongly convex function.

Given assumptions (A.1) to (A.6), we obtain the following theorem of SPELL.

Theorem 1. If (A.1)–(A.6) are satisfied, then the sequence of $[x^k]$ generated by the algorithm SPELL converges almost surely to the optimal solution $x^*_1 \in X^*$ of problem (8).

To prove Theorem 1, we need to use the Martingale convergence theorem and three lemmas.

Martingale Convergence Theorem. A sequence of random variables $\{W^k\}$, which are $H_k$-measurable, is said to be a supermartingale if the sequence of conditional expectations $E[\{W^k\} | H_k]$ exists and satisfies $E[\{W^k\} | H_k] \leq W^k$.

Theorem 2 (from [43], p.26). Let $W^k$ be a positive supermartingale. Then, $W^k$ converges to a finite random variables a.s.

The above definition indicates that $W^k$ is essentially the stochastic analogue of a decreasing sequence.

On the basis of the convexity property, the optimal solution for problem (i) in Figure 3 at iteration $m$ is characterized by the following inequality:

$$(\tilde{q}^m(\tilde{X}_m)) (x_1 - x^*_m) \geq 0, \quad \forall x_1 \in X.$$  \hspace{1cm} (A.3)

To obtain Theorem 1, we require the following three lemmas. The first lemma indicates that the difference between the solutions of two consecutive update processes is bounded by the step size and magnitude of the stochastic gradient. The second lemma shows that the approximation $Q^k (x)$ is finite, and the third lemma indicates that $T^k$ (which will be explained later) is bounded.

Lemma 1. For any two iterations $i \in [\bar{m}, m + 1]$ and $j \in [m + 1, m + 2]$, solutions $x^*_i$ and $x^*_j$ produced by SPELL satisfy the following:

$$\alpha_m g^{\bar{m}} (x^*_i - x^*_j) \leq \frac{(\alpha_m c_1)^2}{b}.$$  \hspace{1cm} (A.4)

Proof. Consider a special case, where $i$ and $j$ correspond to two consecutive iterations. Let $i = m + 1 - 1$ and $j = m + 1$. On the basis of (A.3), we know that

$$(\tilde{q}^{m+1}(\tilde{X}_m)) (x_1 - x^{m+1}_1) \geq 0, \quad \forall x_1 \in X.$$  \hspace{1cm} (A.5)

On the basis of the updating rule of the approximation function, we can obtain

$$\alpha_{m+1} g^{m+1} (x^*_1 - x^*_j) \leq \frac{(\alpha_{m+1} c_1)^2}{b}.$$  \hspace{1cm} (A.6)

We substitute $x_1$ with $x^{m+1}_1$ in equation (A.6) and obtain

$$\alpha_{m+1} (g^{m+1} - \tilde{g}^{m+1}) (x^{m+1}_1 - x^*_1) \geq q^{m+1} (x^{m+1}_1 - x^*_1) (x^*_1 - x^{m+1}_1).$$  \hspace{1cm} (A.7)
We rearrange the terms to obtain the following expression:

$$
\begin{align*}
\alpha_{m+1}^{-1} \left( g^{m+1-1} \right)^T \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
\geq \frac{1}{d} \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
- \alpha_{m+1}^{-1} \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T \right) \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
= \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T - \frac{1}{d} \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T \right) \right) \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
+ \left( 1 - \alpha_{m+1}^{-1} \right) \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T \right) \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
= \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T - \frac{1}{d} \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T \right) \right) \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
+ \left( 1 - \alpha_{m+1}^{-1} \right) \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T \right) \left( x_1^{m+1-1} - x_1^{m-1} \right).
\end{align*}
$$

(A.8)

If iteration $m + 1$ and $m + 1$ are not in the same update process, this implies that $q^{m+1-1} \left( x_1^{m+1-1} \right) \neq q^{m+1-1} \left( x_1^{m+1-1} \right)$. Based on assumption (A.6), we know that $|q^{m+1-1} \left( x_1^{m+1-1} \right) - q^{m+1-1} \left( x_1^{m+1-1} \right)| \geq b |x_1^{m+1-1} - x_1^{m+1-1}|$. On the basis of equations (A.3) and (A.6) and $0 < \alpha_{m+1}^{-1} < 1$, we obtain

$$
\alpha_{m+1}^{-1} \left( g^{m+1-1} \right)^T \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
\geq \frac{b}{x_1^{m+1-1} - x_1^{m-1}} \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
+ \left( 1 - \alpha_{m+1}^{-1} \right) \left( q^{m+1-1} \left( x_1^{m+1-1} \right)^T \right) \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
\geq \frac{b}{x_1^{m+1-1} - x_1^{m-1}} \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
\leq \alpha_{m+1}^{-1} \left( g^{m+1-1} \right)^T \left( x_1^{m+1-1} - x_1^{m-1} \right).
$$

(A.9)

We apply Schwartz’ inequality to obtain the following expression:

$$
\alpha_{m+1} \left\| g^{m+1-1} \right\| \left\| x_1^{m+1-1} - x_1^{m-1} \right\| \geq \alpha_{m+1} \left( g^{m+1-1} \right)^T \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
\geq \frac{b}{x_1^{m+1-1} - x_1^{m-1}} \left( x_1^{m+1-1} - x_1^{m-1} \right) \\
\geq \frac{b}{x_1^{m+1-1} - x_1^{m-1}} \left( x_1^{m+1-1} - x_1^{m-1} \right).
$$

(A.10)

Thus, $\left\| x_1^{m+1-1} - x_1^{m-1} \right\| \leq \alpha_{m+1}^{-1} \cdot c_1 / b$. The following expression is applicable:

$$
\alpha_{m+1}^{-1} \left( g^{m+1-1} \right)^T \left( x_1^{m+1-1} - x_1^{m-1} \right) \leq \frac{\left( \alpha_{m+1}^{-1} c_1 \right)^2}{b}
$$

(A.11)

For any $i \in [m, m + 1 - 1]$, $g^i = g^{m+1-1}$ and $\tilde{q}^i \left( x_1 \right) = \frac{1}{d} \left( q^{m+1-1} \left( x_1 \right)^T \right) \left( x_1^{m+1-1} - x_1^{m-1} \right)$. Therefore,

$$
\alpha_m \left( g^m \right)^T \left( x_1^m - x_1^m \right) \leq \frac{\left( \alpha_m c_1 \right)^2}{b}
$$

(A.12)

For any $i \in [m, m + 1 - 1]$, $j \in [m + 1, m + 2 - 1]$ and $g^j = g^{m+1} = g^{m+2-1}$ for any $x_1 \in X$. Therefore,

$$
\alpha_m \left( g^m \right)^T \left( x_1^i - x_1^i \right) \leq \frac{\left( \alpha_m c_1 \right)^2}{b}
$$

(A.13)

\[\square\]

Lemma 2. The approximation function $\tilde{q}^k \left( x_1 \right)$ in iteration $k$ can be written as $Q^k \left( x_1 \right) = Q^0 \left( x_1 \right) + (\tilde{g}^k)^T x_1$, where $\tilde{g}^k$ is a finite vector.

Proof. According to equation (24) in Proposition 1, we can conclude that $\tilde{g}^k$ is a linear combination of $g^1, g^2, \ldots, g^k$. Since $\tilde{g}^k$ and $Q^k \left( x_1 \right)$ are finite, there exists a finite and positive vector $d$ such that

$$
\tilde{d} \geq \max_k \left| g^k - Q^k \left( x_1^* \right) \right|.
$$

(A.14)

According to lemma 2 by Cheung and Powell [30], we can conclude that $\tilde{g}^k \geq \tilde{d}$.

Let $T^k = Q^k \left( x_1^* \right) - Q^k \left( x_1^* \right)$, where $x_1^*$ denotes the optimal solution. The following lemma characterizes the difference between $T^{k+1}$ and $T^k$.

Lemma 3. For any two iterations $i \in [m - 1, m - 1]$ and $j \in [m, m + 1 - 1]$, $T^{i-j}$ and $T^i$ satisfy

$$
T^{i-j} \leq \alpha_m \left( g^m \right)^T \left( x_1^i - x_1^i \right) + \alpha_m \left( g^m \right)^T \left( x_1^* - x_1^i \right).
$$

(A.15)

Proof. We consider the special case. Let $i = m$ and $j = m + 1$. By rewriting $x_1^* - x_1^{m+1} = x_1^* - x_1^{m} + x_1^{m} - x_1^{m+1}$, we obtain

$$
\begin{align*}
Q^m \left( x_1 \right) = Q^m \left( x_1 \right) + \alpha_m \left( \tilde{g}^m - \tilde{q}^m \left( x_1 \right) \right)^T x_1 \\
\left( Q^m \left( x_1^{m+1} \right) + \alpha_m \left( \tilde{g}^m - \tilde{q}^m \left( x_1 \right) \right)^T \right) x_1 \\
= Q^m \left( x_1 \right) + \alpha_m \left( g^m - \tilde{q}^m \left( x_1 \right) \right)^T \left( x_1^* - x_1 \right)
\end{align*}
$$

(A.16)

Subsequently,

$$
\begin{align*}
T^{m+1} = Q^m \left( x_1 \right) + \alpha_m \left( g^m - \tilde{q}^m \left( x_1 \right) \right)^T \left( x_1^* - x_1 \right) \\
= \alpha_m \left( g^m - \tilde{q}^m \left( x_1 \right) \right)^T \left( x_1^* - x_1 \right)
\end{align*}
$$

(A.17)
We consider each part individually. Given that $\bar{Q}^m \in \partial Q^m(x_i^m)$, by convexity of $Q^m(x_i^m)$, we obtain

$$Q^m(x_i^m) - Q^m(x_i^{m+1}) \leq (\bar{Q}^m)^T (x_i^m - x_i^{m+1}). \quad (A.18)$$

Hence, the expression is applicable:

$$Q^m(x_i^m) - Q^m(x_i^{m+1}) \leq (\bar{Q}^m)^T (x_i^m - x_i^{m+1})$$

so that

$$= (1 - \alpha_m)(\bar{Q}^m)^T (x_i^m - x_i^{m+1}) + \alpha_m (\bar{Q}^m)^T (x_i^m - x_i^{m+1}).$$

(A.19)

Given equation (A.3) and $0 < \alpha_m < 1$, we know that (I) $\leq 0$. Additionally, from equation (A.3) and $0 < \alpha_m < 1$, we know that (II) $\geq 0$. Thus, $T^{m+1} - T^m \leq \alpha_m (\bar{Q}^m)^T (x_i^m - x_i^{m+1}) + \alpha_m (\bar{Q}^m)^T (x_i^m - x_i^{m+1})$. For any $i \in [m, m+1 - 1]$, $g^m = g^{m+1 - 1}$ and $\bar{Q}^m(x_i) = \bar{Q}^m(x_i^m) = \bar{Q}^m(x_i^{m+1})$ for any $x_i \in X$. Therefore,

$$T^{m+1} - T^m \leq \alpha_m (\bar{Q}^m)^T (x_i^m - x_i^{m+1}) + \alpha_m (\bar{Q}^m)^T (x_i^m - x_i^{m+1}).$$

(A.20)

For any $i \in [m, m+1 - 1]$ and $j \in [m+1, m+2 - 1]$, $Q^j = Q^{m+1} = \bar{Q}^{m+2}$ for any $x_i \in X$. Therefore,

$$T^{j} - T^{i} \leq \alpha_m (\bar{Q}^m)^T (x_i^m - x_i^{m+1}) + \alpha_m (\bar{Q}^m)^T (x_i^m - x_i^{m+1}).$$

(A.21)

With regard to the proof of Theorem 1, we consider two cases. In the first case, the algorithm does not terminate in a given update process. Therefore, any update process before the algorithm terminates exhibits finite iterations, that is, $m + T - m < M$ for any $m$, where $M$ denotes a large number. In the second case, the algorithm may halt in a given update process. We prove Theorem 1 in each case.

**Case 1.** SPELL does not terminate in a given update process.

In this case, we consider a subsequence of $\{x_i^m\}, \{x_i^{m+1}\}. We prove that the subsequence $\{x_i^m\}$ converges to the true optimal $x_i^*$. By the definition of $g^m \in \partial Q(x_i^m, \omega^{k+1})$, we can obtain

$$(g^m)^T (x_i^m - x_i^*) \leq Q(x_i^*, \omega^{k+1}) - Q(x_i^m, \omega^{k+1}), \quad (A.22)$$

where $Q(x_i^*, \omega^{k+1})$ denotes the operational cost function given the outcome $\omega^{k+1}$.

Lemma 1 implies the following:

$$\alpha_m (g^m)^T (x_i^m - x_i^*) \leq \frac{(\alpha_m c_1)^2}{b}. \quad (A.23)$$

Based on Lemma 3, the difference $T^{m+1} - T^m$ is as follows:

$$T^{m+1} - T^m \leq \alpha_m (g^m)^T (x_i^m - x_i^{m+1}) + \alpha_m (g^m)^T (x_i^* - x_i^m)$$

$$\leq -\alpha_m Q(x_i^m, \omega^{m+1}) - Q(x_i^*, \omega^{m+1})$$

$$+ \alpha_m (g^m)^T (x_i^* - x_i^m)$$

$$\leq -\alpha_m Q(x_i^m, \omega^{m+1}) - Q(x_i^*, \omega^{m+1})$$

$$+ \frac{(\alpha_m c_1)^2}{b}. \quad (A.24)$$

We take conditional expectation of equation (A.24) with respect to $\mathcal{F}$ on both sides and obtain

$$E \left( T^{m+1} \mid \mathcal{F}_m \right) \leq T^m - \alpha_m (Q(x_i^m) - Q(x_i^*)) + \frac{(\alpha_m c_1)^2}{b}, \quad (A.25)$$

where $Q(x_i^m)$ denotes the expected recourse function, that is, $E_0 Q(x_i, \omega); T^m, \alpha_m,$ and $x_i^m$ on the right-hand side are deterministic given the conditioning on $\mathcal{F}_k$. The conditioning $\mathcal{F}_k$ does not provide any information on $\omega^{m+1}$. Thus, we replace $Q(x_i^m, \omega^{m+1})$ (for $x_i = x_i^k$ and $x_i = x_i^*$) with its expectation $Q(x_i^*)$. Given that $\alpha_m (Q(x_i^m) - Q(x_i^*)) \geq 0$, the sequence

$$W^m = T^m + \frac{(\alpha_m c_1)^2}{b}, \quad (A.26)$$

is a positive supermartingale. Theorem 2 implies the almost sure convergence of $W^m$. Thus,

$$T^m \longrightarrow T^*, \quad \text{a.s.} \quad (A.27)$$

We perform the summation of equation (A.24) from 0 to $M$ and obtain the following expression:

$$T^M - T^0 \leq \sum_{m=0}^{M} \alpha_m Q(x_i^m, \omega^{m+1}) - Q(x_i^*, \omega^{m+1})$$

$$+ \sum_{m=0}^{M} \frac{(\alpha_m c_1)^2}{b}. \quad (A.28)$$

We take the expectation of both sides. For the first term on the right-hand side, we take the conditional expectation with respect to $\mathcal{F}_m$ and then over all $\mathcal{F}_m$: 
$$E\left(T^{MT_1} - T^6\right) \leq - \sum_{m=0}^{MT_1} E\left[\alpha_m Q\left(\bar{x}_1^m, \omega_{m+1}\right) - Q\left(x_1^*, \omega_{m+1}\right)\right] \leq - \sum_{m=0}^{MT_1} E\left[\alpha_m \left(Q\left(x_1^m, \omega_{m+1}\right) - Q\left(x_1^*, \omega_{m+1}\right)\right)\right] + E\left[\frac{\sum_{m=0}^{MT_1} (\alpha_m)^2}{b}\right]$$

(A.29)

**Conflicts of Interest**

The authors declare that they have no conflicts of interest.

**Acknowledgments**

This work was partially supported by the National Nature Science Foundation of China (NSFC), under Grant numbers: 71701221, 71601054, 71901227, and 71431007, the MOE (Ministry of Education in China) Project of Humanities and Social Sciences, under Grant number: 17YJ630235, and the Natural Science Foundation of Guangdong Province, China, under Grant number: 2019A1515011127.

**References**

[1] UNCTAD, *Review of Maritime Transport*, United Nation Publication, Geneva, Switzerland, 2018, http://unctad.org/en/PublicationsLibrary/rmt2018_en.pdf.

[2] D. P. Song, "Optimal threshold control of empty vehicle redistribution in two depot service systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 1, pp. 87–90, 2005.

[3] T. G. Crainic, M. Gendreau, and P. Dejax, "Dynamic and stochastic models for the allocation of empty containers," *Operations Research*, vol. 41, no. 1, pp. 102–126, 1993.

[4] Y. M. Ermoliev, T. Krivets, and V. Petukhov, "Planning of shipping empty seaborne containers," *Cybernics and Systems Analysis*, vol. 12, no. 4, pp. 644–646, 1977.

[5] K. A. Kuzmicz and E. Pesch, "Approaches to empty container repositioning problems in the context of Eurasian intermodal transportation," *Omega*, vol. 85, pp. 194–213, 2019.

[6] S. Lee and I. Moon, "Robust empty container repositioning considering foldable containers," *European Journal of Operational Research*, vol. 280, no. 3, pp. 909–925, 2020.

[7] Y. Long, L. H. Lee, and E. P. Chew, "The sample average approximation method for empty container repositioning with uncertainties," *European Journal of Operational Research*, vol. 222, no. 1, pp. 65–75, 2012.

[8] I. Moon, A.-D. Do Ngoc, and R. Konings, "Foldable and standard containers in empty container repositioning," *Transportation Research Part E: Logistics and Transportation Review*, vol. 49, no. 1, pp. 107–124, 2013.

[9] K. Shintani, R. Konings, and A. Imai, "Combining containers: a container innovation to save container fleet and empty container repositioning costs," *Transportation Research Part E*, vol. 130, no. 1, pp. 248–272, 2020.

[10] C.-M. Feng and C.-H. Chang, "Empty container reposition planning for intra-Asia liner shipping," *Maritime Policy & Management*, vol. 35, no. 5, pp. 469–489, 2008.

[11] K. Wang, S. Wang, L. Zhen, and X. Qu, "Ship type decision considering empty container repositioning and foldable containers," *Transportation Research Part E: Logistics and Transportation Review*, vol. 108, no. 1, pp. 97–121, 2017.
[12] A. L. Erera, J. C. Morales, and M. Savelsbergh, "Robust optimization for empty repositioning problems," *Operations Research*, vol. 57, no. 2, pp. 468–483, 2009.

[13] M. D. Francesco, T. G. Crainic, and P. Zaddas, "The effect of multi-scenario policies on empty container repositioning," *Transportation Research Part E*, vol. 45, no. 5, pp. 758–770, 2009.

[14] C.-Y. Lee and D.-P. Song, "Ocean container transport in global supply chains: overview and research opportunities," *Transportation Research Part B: Methodological*, vol. 95, pp. 442–474, 2017.

[15] S. Kong and Q. Meng, "Container liner fleet deployment: a systematic overview," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 389–404, 2017.

[16] Z. Song, W. Tang, and R. Zhao, "Encroachment and canvassing strategy in a sea-cargo service chain with empty container repositioning," *European Journal of Operational Research*, vol. 276, no. 1, pp. 175–186, 2019.

[17] W.-H. Zhou and C.-Y. Lee, "Pricing and competition in a transportation market with empty equipment repositioning," *Transportation Research Part B: Methodological*, vol. 43, no. 6, pp. 677–691, 2009.

[18] J.-X. Dong, C.-Y. Lee, and D.-P. Song, "Joint service capacity planning and dynamic container routing in shipping network with uncertain demands," *Transportation Research Part B: Methodological*, vol. 78, pp. 404–421, 2015.

[19] Y. Jeong, S. Saha, D. Chatterjee, and I. Moon, "Direct shipping service routes with an empty container management strategy," *Transportation Research Part E: Logistics and Transportation Review*, vol. 118, no. 1, pp. 123–142, 2018.

[20] Q. Meng and S. Wang, "Liner shipping service network design with empty container repositioning," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 5, pp. 695–708, 2011.

[21] W. Shan, Z. Peng, J. Liu, B. Yao, and B. Yu, "An exact algorithm for inland container transportation network design," *Transportation Research Part B: Methodological*, vol. 135, no. 1, pp. 41–82, 2020.

[22] D.-P. Song and J.-X. Dong, "Empty container management in cyclic shipping routes," *Maritime Economics & Logistics*, vol. 10, no. 4, pp. 335–361, 2008.

[23] Y. Xie, X. Liang, L. Ma, and H. Yan, "Empty container management and coordination in intermodal transport," *European Journal of Operational Research*, vol. 257, no. 1, pp. 223–232, 2017.

[24] R. K. Cheung and C.-Y. Chen, "A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem," *Transportation Science*, vol. 32, no. 2, pp. 142–162, 1998.

[25] X. Fei, N. Gulpinar, and J. Branke, "Efficient solution selection for two-stage stochastic programs," *European Journal of Operational Research*, vol. 277, no. 3, pp. 918–929, 2019.

[26] L. Yang, Y. Xiong, and X. Tong, "A smoothing algorithm for a new two-stage stochastic model of supply chain based on sample average approximation," *Mathematical Problems in Engineering*, vol. 2017, Article ID 5681502, 7 pages, 2017.

[27] Y. M. Ermoliev, "Stochastic quasigradient methods," in *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, New York, NY, USA, 1988.

[28] S. Rebenneck, "Combining sampling-based and scenario-based nested benders decomposition methods: application to stochastic dual dynamic programming," *Mathematical Programming*, vol. 156, no. 1-2, pp. 343–389, 2016.

[29] G. Steeger, T. Lohmann, and S. Rebenneck, "Strategic bidding for a price-maker hydroelectric producer: stochastic dual dynamic programming and Lagrangian relaxation," *IIE Transactions*, vol. 50, no. 11, pp. 929–942, 2018.

[30] R. K.-M. Cheung and W. B. Powell, "Shape—a stochastic hybrid approximation procedure for two-stage stochastic programs," *Operations Research*, vol. 48, no. 1, pp. 73–79, 2000.

[31] G. A. Godfrey and W. B. Powell, "An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution," *Management Science*, vol. 47, no. 8, pp. 1101–1112, 2001.

[32] G. A. Godfrey and W. B. Powell, "An adaptive dynamic programming algorithm for dynamic fleet management, II: multiperiod travel times," *Transportation Science*, vol. 36, no. 1, pp. 40–54, 2002.

[33] J. Nascimento and W. B. Powell, "An optimal approximate dynamic programming algorithm for concave, scalar storage problems with vector-valued controls," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 2995–3010, 2013.

[34] W. B. Powell, B. Bouzaiene-Ayari, C. Lawrence, C. Cheng, S. Das, and R. Fiorillo, "Locomotive planning at Norfolk southern using approximate dynamic programming," *Interfaces*, vol. 44, no. 6, 2014.

[35] W. B. Powell and G. A. Godfrey, "An adaptive dynamic programming algorithm for dynamic fleet management, I: single period travel times," *Transportation Science*, vol. 36, no. 1, pp. 40–54, 2002.

[36] W. Powell, A. Ruszczyński, and H. Topaloglu, "Learning algorithms for separable approximations of discrete stochastic optimization problems," *Mathematics of Operations Research*, vol. 29, no. 4, pp. 814–836, 2004.

[37] S. Zhou, H. Zhang, N. Shi, Z. Xu, and F. Wang, "A new convergent hybrid learning algorithm for two-stage stochastic programs," *European Journal of Operational Research*, vol. 283, no. 1, pp. 33–46, 2020.

[38] R. T. Rockafellar and J. B. Wets, "A note about projection in the implementation of stochastic quasi-gradient methods," *Numerical Techniques for Stochastic Optimization*, Series in Computational Mathematics, vol. 10, pp. 385–392, Springer, Berlin, Germany, 1988.

[39] COSCO, *Lines and Services*, COSCO, Shanghai, China, 2019, http://lines.coscoshipping.com/home/Services/route.

[40] Y. Zhao, Q. Xue, Z. Cao, and X. Zhang, "A two-stage chance constrained approach with application to stochastic intermodal service network design programs," *Journal of Advanced Transportation*, vol. 2018, Article ID 6051029, 18 pages, 2018.

[41] R. M. Van Slyke and J. B. Wets, "L-shaped linear program with uncertain demands," *Mathematics of Operations Research*, vol. 29, no. 4, pp. 814–836, 2004.

[42] S. Zhou, H. Zhang, N. Shi, Z. Xu, and F. Wang, "A new convergent hybrid learning algorithm for two-stage stochastic programs," *European Journal of Operational Research*, vol. 283, no. 1, pp. 33–46, 2020.

[43] R. T. Rockafellar and J. B. Wets, "A note about projection in the implementation of stochastic quasi-gradient methods," *Numerical Techniques for Stochastic Optimization*, Series in Computational Mathematics, vol. 10, pp. 385–392, Springer, Berlin, Germany, 1988.

[44] COSCO, *Lines and Services*, COSCO, Shanghai, China, 2019, http://lines.coscoshipping.com/home/Services/route.

[45] Y. Zhao, Q. Xue, Z. Cao, and X. Zhang, "A two-stage chance constrained approach with application to stochastic intermodal service network design programs," *Journal of Advanced Transportation*, vol. 2018, Article ID 6051029, 18 pages, 2018.

[46] R. M. Van Slyke and J. B. Wets, "L-shaped linear programs with applications to optimal control and stochastic programming," *SIAM Journal on Applied Mathematics*, vol. 17, no. 4, pp. 638–663, 1969.

[47] R. K. Cheung and W. B. Powell, "An algorithm for multistage stochastic networks with random arc capacities, with an application to dynamic fleet management," *Operations Research*, vol. 44, no. 6, pp. 951–963, 1996.

[48] J. Neveu, *Discrete Parameter Martingales*, North Holland, Amsterdam, Netherlands, 1975.