Two-stage stochastic mixed integer programming model for multi-period sea cargo mix problem

Chengxuan CAO
The Logistics Institute-Asia Pacific, National University of Singapore, AS6, Singapore 119260
tliccx@nus.edu.sg

James Soo Keng ANG
Business School, National University of Singapore, BIZ 1, Singapore 117592
bizangsk@nus.edu.sg

Heng-Qing YE
Business School, National University of Singapore, BIZ 1, Singapore 117592
bizyehq@nus.edu.sg

Abstract

We present a two-stage stochastic mixed integer programming model and a heuristic algorithm for multi-period sea cargo mix problem. Multi-period sea cargo mix problem occurs in the context of container ship planning and revenue analysis in international ocean container shipping industry. We introduce a two-stage stochastic mixed integer programming formulation for the problem, which accounts for total profit in multi-period, constraints of limited shipping capacity, limited number of empty containers in the origin port, etc. The solution to the problem is found by maximizing the expected profit over the possible control decisions under the uncertainty of shipping capacity. A heuristic algorithm is proposed to obtain a near optimal solution for the problem. Finally, we give numerical experiments demonstrate the efficiency of the algorithm.

Keywords: Logistics; Sea cargo mix; Stochastic mixed integer programming; Heuristic algorithm

1. Introduction and Problem Description

Competition in the modern shipping industry is intense. Profit margins are razor-thin, especially after the US deregulated shipping rates in late 1999. Amid such upheaval, any advantage was vital. Shipping companies have to look for substantial efficiencies through better management. More efficient management had become critical to the success of the business of a shipping company. The company had to track each container on every vessel, determine the most profitable mix of cargo, and ensure that they were operating as near capacity as possible. Moreover, they had to make sure they met the shipping needs of their premium customers, even on short notice. How to increase their revenue by improving the container ship planning system is the main challenge of a global ocean containerized shipping company.

Shipping companies employing their own ships are considered. They must maximize their profits on a highly competitive quality-sensitive market by optimizing the purchase and use of their ships (resources). In particular, company decision makers try to attain this objective by acting on the number and type of ships, type of service (regular or on request) and on dates of ship sailings. A mathematical model will be developed to cope with the problem of maximizing a company's profit, while ensuring satisfactory service. The model to be constructed can be subdivided into three main hierarchical decision levels: a strategic, a tactical, and an operational level (see Pesenti (1995)).

Strategic decisions involve capital acquisition issues, such as the sale or purchase of a particular ship (or ships), including which shipyard to buy it from, when to do so, whether it is new or second-hand, what form of financing should be used, and so on. The planning horizon for such decisions is on the order of 3-20 years. Tactical decisions involve the allocation/utilization of ships owned by the operator, including the issue of which charter should be fixed, whether to offer the ship in the spot market or in the term market, whether the ship should be laid up, how the ship should be "positioned" to be best used in future charters, what should be the ship's operating speed, routine maintenance, and so on. Tactical decisions have a shorter planning horizon, on the order of a few months to 2 years. Finally, operational decisions have a much shorter time horizon (a few days to a few months), and involve issues of the day-to-day operation of the ship, such as management of stores and supplies, bunkering, non-routine maintenance, etc. The line between the tactical and operational decision levels is to some extent arbitrary and depends on the idiosyncrasy of the particular owner.

In practice, an optimal solution to the problem of maximizing the net profit of a company is generally too complex. Even in the cases of a-priori known customer arrivals and ships already assigned to routes, the scheduling problem would be NP-hard. In addition, as known data are often inaccurate, it may not be worth performing an exhaustive search for an optimal solution that might not be completely suitable for a real case. A certain analysis might be performed in a possible feedback phase of the decision process in order to evaluate the stability of a proposed solution or to establish whether environmental changes have become so significant that previous decisions must be modified (also see Magirou et al (1992)).

An important problem arises when a liner operator wants to select cargoes in order to maximize his profit for a particular trip. Each type of cargo has different volume/weight characteristic and different freight rate. The cargo selection should be done without violating the volume and weight constraints of the ship. If the freight rates are known and the operator can refuse or delay to carry cargoes, then the cargo mix that optimizes the total profit can be determined by mixed integer programming techniques. Evans and Marlow (1985) present some empirical solutions for the problem that do not require programming techniques, and thus were useful before the computers made these particular techniques easily accessible.

One of the single largest expenses for most container operators is the repositioning of empty containers. Regarding container demand and supply it minimizes empty repositioning costs and maximizes vessels utilization and it lead to a pro-active pricing mechanism for improved revenues and reduced costs. In order to optimally use vessels optimally matched where possible according to customer requirements to minimize empty repositioning costs. This takes into account excess empties and will result in optimal utilization of vessels while total transportation costs (for both full and empty containers) are reduced and trade imbalances will be compensated.

To our knowledge, there is no description and formulation for the multi-period sea cargo mix problem in the literature. Single period sea cargo mix problem is proposed by Evans and Marlow (1985) and it is formulated as linear programming model by Magirou et al (1992), when all cargoes are considered separately for shipment. Literature on related mathematical programming models and algorithms used in this paper are provided in related sections.

The rest of the paper is organized as follows. In Section 2, the multi-period sea cargo mix problem is formulated as two-stage stochastic mixed integer programming model. Section 3 presents algorithm for the model. The heuristic algorithm with effective gradient and its modified algorithm are given in this section. We demonstrate the algorithm with effective gradient by a simple example in this section, followed in Section 4 by numerical experiments of the algorithm with a wide range of problem instances. Section 5 concludes the paper.

2. Two-stage stochastic mixed integer programming model

There are several technical constraints relative to how a ship should be loaded. The obvious constraints are that cargo weight and volume should not exceed the ship's capacity. The operational objective is to select the loading parameters in order to maximize total profit subject to the mathematical expression of the above weight/volume constraints. When there are several different cargoes to be transported, a reasonable criterion might be to maximize a weighted sum of the quantities transported. The relative coefficients for the cargoes could reflect their unit profits.

In practice, the problem parameters associated with the sea cargo mix problem are rarely known with certainty. The uncertainty of ship scheduling is due to bad weather condition at sea and unpredictable service times in ports. In addition, liner operators cannot plan their vessels to load what they have booked; they must plan their vessels for what is going to show up. Not all cargo is delivered when originally scheduled, nor does all cargo show up in the same quantity and configuration that was booked. Often this changing total cargo mix will necessitate a change in vessel. In the same respect, sometimes a ship that was supposed to be available on a given date runs late. As the cargo mix (or vessel availability) changes, they will constantly match what they believe is the best ship to the expected cargo.

We assume that the sea cargo mix decisions (for the entire planning horizon) have to be made, with only some knowledge of future scenarios of parameters. The overall objective is to determine a sea cargo mix plan, such that the sum of each expected profit is maximized. Although the problem is a multi-period one, since the sea cargo mix planning decisions for all periods are made in period one, the decisions are strategic in nature and need to be decided over longer planning periods. However, it is also used as the operational level planning tools and can be decided when more information becomes available. In principle, the planning decisions to be revised as time progresses and more information becomes available. To incorporate the uncertainty in the parameters, we assume that these parameters can be realized as one of S scenarios. The probability of scenario s will be denoted by p^s.

Before formally stating the problem, we introduce some notation:

Index Sets

$\widetilde{T}$: set of time periods $\{1, 2, \cdots, t, \cdots, T\}$. For the ease of discussion, we assume that each period represents one day. The variable t is always used to represent the tth time period.

$\widetilde{J}$: set of ports $\{1, 2, \cdots, j, \cdots, J\}$. For completeness, we may also denote the origin port as *O*, which is not explicitly used in our model and thus can be omitted.

$\widetilde{K}_t$: set of all cargoes received in period t, i.e. $\widetilde{K}_t = \{1, 2, \cdots, k, \cdots, K_t\}$. This information about shipment requirement can be gathered from for example the advance shipment booking. For convenience, when we refer to the cargo *k*, it is always associated with a period *t* or the set $\widetilde{K}_t$. Thus, the cargo *k* implicitly depends on the time index *t*. As a special case, we may assume that all cargoes will be received in a same period.

$\widetilde{S}$: set of all scenarios $\{1, 2, \cdots, s, \cdots, S\}$.

Deterministic Parameters

j_k : The destination port for cargo k. (By default, we have $k \in \widetilde{K}_t$.) Cargo k will be received in period t and will be shipped to its destination port j_k before its due date.

r_{tkdj_k} : per volume profit of cargo k which is received in period t and shipped to port j_k in period d. It can be interpreted as the per volume net profit of cargo k, i.e., per volume profit of cargo k minus its per volume inventory cost.

τ_k : due date of cargo k. Each cargo has its due date requested by shipper in its booking status.

$v_{tk\tau_k j_k}$: the volume of cargo k received in period t ready for shipment to port j_k before its due date τ_k.

$w_{tk\tau_k j_k}$: weight of cargo k received in period t ready for shipment to port j_k before its due date τ_k.

Random Data

E_t^s : total volume of available empty containers at origin port in period t under scenario s. Short of available empty containers also impacts the containerized shipping.

V_{tj}^s : total available volume capacity of shipment to port j in period t under scenario s.

W_{tj}^s : maximum allowable weight capacity of shipment to port j in period t under scenario s.

q_{dj1}^s : overage cost per unit overage of volume capacity for delivery to port j in period d under scenario s.

q_{dj2}^s : shortage cost per unit short of volume capacity for delivery to port j in period d under scenario s.

q_{dj3}^s : overage cost per unit overage of weight capacity for delivery to port j in period d under scenario s.

q_{dj4}^s : shortage cost per unit short of weight capacity for delivery to port j in period d under scenario s.

q_{d5}^s : overage cost per unit overage of allowable empty containers in period d under scenario s.

q_{d6}^s : shortage cost per unit short of allowable empty containers in period d under scenario s.

Decision Variables

x_{tkdj_k} : binary variable, i.e., $x_{tkdj_k} = 1$ if cargo k is received in period t and is ready for shipment to port j_k in period d before its due date τ_k, 0, otherwise.

y_{dj1} : amount of overage of available volume capacity for delivery to port j in period d.

y_{dj2}: amount of shortage of available volume capacity for delivery to port j in period d.

y_{dj3}: amount of overage of allowable weight capacity for delivery to port j in period d.

y_{dj4}: amount of shortage of allowable weight capacity for delivery to port j in period d.

y_{d5}: amount of overage of allowable empty containers in period d.

y_{d6}: amount of shortage of allowable empty containers in period d.

{Insert here the Figure 1}

In this paper, we assume that there are one origin port and J destination ports (see Figure 1). All cargoes under consideration will be received at origin port during time horizon T. The carrier decides which cargoes to ship in the current period, which to delay to other periods within the time horizon T, and which to shut out of the time horizon T. The multi-period sea cargo mix problem can then be formulated as following two-stage stochastic mixed integer programming model:

(M1) $$Maximize\, z = \sum_{t=1}^{T} \sum_{k \in \widetilde{K}_t} \sum_{d=t}^{\tau_k} v_{tk\tau_k j_k} r_{tkdj_k} x_{tkdj_k} - \sum_{s=1}^{S} p^s Q^s(x) \quad (1)$$

subject to

$$\sum_{d=t}^{\tau_k} x_{tkdj_k} \le 1 \quad \forall\, t \in \widetilde{T}, \quad k \in \widetilde{K}_t \quad (2)$$

$$x_{tkdj_k} \in \{0, 1\} \quad \forall\, t \in \widetilde{T}, \quad k \in \widetilde{K}_t, \quad d \in \{t, t+1, \cdots, \tau_k\} \quad (3)$$

where for all s,

$$Q^s(x) = Minimize \sum_{d=1}^{T} (q_{d5}^s y_{d5} + q_{d6}^s y_{d6} + \sum_{j=1}^{J} \sum_{i=1}^{4} q_{dji}^s y_{dji}) \quad (4)$$

subject to

$$y_{d5} - y_{d6} = E_d^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \ge d}} v_{tk\tau_k j_k} x_{tkdj_k} \quad \forall\, d \in \widetilde{T} \quad (5)$$

$$y_{dj1} - y_{dj2} = V_{dj}^{s} - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad j \in \widetilde{J} \tag{6}$$

$$y_{dj3} - y_{dj4} = W_{dj}^{s} - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad j \in \widetilde{J} \tag{7}$$

where all y_{d5}, y_{d6}, y_{dji} are non-negative for $d \in \widetilde{T}$, $j \in \widetilde{J}$, $i = 1,2,3,4$.

The objective function of first stage program (1) maximizes the total profit in the time horizon T. Constraint (2) imposes that each cargo may be carried in a certain period before its due date or be refused to carry in the time horizon T. Constraint (3) assures that each cargo can be either accepted at its total quantity or be turned down. The objective function of second stage program (4) minimizes overall cost for overage and shortage of available capacities under scenario s in the time horizon T. Constraints of second stage program (5), (6) and (7) indicate amount of overage or shortage of available capacities under scenario s in the time horizon T.

We may note that the inequality sign in (2) is replaced by equality, if all received cargoes have to be carried in the time horizon T. In this case, (M1) may have not feasible solution, as the total available capacity of shipment may be less than the amount of demands in the time horizon T.

Because all random variables in (M1) are discretely distributed, and their joint distribution has a finite number of realizations, (M1) can be rewritten as the following large-scale mixed integer programming model:

$$\text{(M2)} \quad Maximize\, z = \sum_{t=1}^{T} \sum_{k \in \widetilde{K}_t} \sum_{d=t}^{\tau_k} v_{tk\tau_k j_k} r_{tkdj_k} x_{tkdj_k} - \sum_{s=1}^{S} p^s \sum_{d=1}^{T} (q_{d5}^{s} y_{d5}^{s} + q_{d6}^{s} y_{d6}^{s} + \sum_{j=1}^{J} \sum_{i=1}^{4} q_{dji}^{s} y_{dji}^{s}) \tag{8}$$

subject to

$$\sum_{d=t}^{\tau_k} x_{tkdj_k} \leq 1 \quad \forall t \in \widetilde{T}, \quad k \in \widetilde{K}_t \tag{9}$$

$$y_{d5}^{s} - y_{d6}^{s} = E_d^{s} - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad s \in \widetilde{S} \tag{10}$$

$$y_{dj1}^{s} - y_{dj2}^{s} = V_{dj}^{s} - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} \quad \forall\, d \in \widetilde{T}, \quad j \in \widetilde{J}, \quad s \in \widetilde{S} \tag{11}$$

$$y_{dj3}^{s} - y_{dj4}^{s} = W_{dj}^{s} - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} \quad \forall\, d \in \widetilde{T}, \quad j \in \widetilde{J}, \quad s \in \widetilde{S} \tag{12}$$

where $x_{tkdj_k} \in \{0, 1\} \quad \forall\, t \in \widetilde{T}, \quad k \in \widetilde{K}_t, \quad d \in \{t, t+1, \cdots, \tau_k\}$, and all y_{d5}^{s}, y_{d6}^{s}, y_{dji}^{s} are non-negative for $s \in \widetilde{S}$, $d \in \widetilde{T}$, $j \in \widetilde{J}$, $i = 1, 2, 3, 4$.

The problem (M2) can be solved using standard mixed integer programming method, but it is doomed to failure since no advantage of the special problem structure is exploited. In next section, we give an efficient heuristic algorithm of problem (M2).

3. Heuristic Algorithm for the Model

In this section, we present a simple and quick method for obtaining approximate solutions to the stochastic mixed integer programming model. A common attitude in solving NP-hard combinatorial optimization problems is to not insist on optimality but dedicate research efforts to designing fast and high quality approximation methods. A greedy algorithm is chosen for solve the problem (M2). Its robust and implicit enumerative character ensures to achieve the optimal solution or a near optimal solution. In cases like this we can sacrifice the guarantee of optimality that is provided by it in favor of getting a reasonable answer quickly.

Let $E_d = \min\{E_d^s : s \in \widetilde{S}\}$, $V_{dj} = \min\{V_{dj}^s : s \in \widetilde{S}\}$, $W_{dj} = \min\{W_{dj}^s : s \in \widetilde{S}\}$, $\overline{v}_{tk\tau_k j_k}^{d} = v_{tk\tau_k j_k} / V_{dj_k}$, $\overline{w}_{tk\tau_k j_k}^{d} = w_{tk\tau_k j_k} / W_{dj_k}$. Then, we get a binary integer programming as follows:

(M3)
$$Maximize\, z = \sum_{t=1}^{T} \sum_{k \in \widetilde{K}_t} \sum_{d=t}^{\tau_k} v_{tk\tau_k j_k} r_{tkdj_k} x_{tkdj_k} \tag{13}$$

subject to

$$\sum_{d=t}^{\tau_k} x_{tkdj_k} \leq 1 \quad \forall\, t \in \widetilde{T}, \quad k \in \widetilde{K}_t \tag{14}$$

$$\sum_{t=1}^{d}\sum_{\substack{k\in\widetilde{K}_t\\ \tau_k\ge d}} v_{tk\tau_k j_k} x_{tkdj_k} \le E_d \quad \forall d\in\widetilde{T} \tag{15}$$

$$\sum_{t=1}^{d}\sum_{\substack{k\in\widetilde{K}_t\\ \tau_k\ge d\\ j_k=j}} \overline{v}^{d}_{tk\tau_k j_k} x_{tkdj_k} \le 1 \quad \forall d\in\widetilde{T},\quad j\in\widetilde{J} \tag{16}$$

$$\sum_{t=1}^{d}\sum_{\substack{k\in\widetilde{K}_t\\ \tau_k\ge d\\ j_k=j}} \overline{w}^{d}_{tk\tau_k j_k} x_{tkdj_k} \le 1 \quad \forall d\in\widetilde{T},\quad j\in\widetilde{J} \tag{17}$$

where $x_{tkdj_k}\in\{0,1\} \quad \forall t\in\widetilde{T},\quad k\in\widetilde{K}_t,\quad d\in\{t,t+1,\cdots,\tau_k\}$.

From the structure of the binary integer programming (M3), we easily obtain following two propositions:

Proposition 1. A feasible solution of the binary integer programming (M3) is also a feasible solution of the mixed integer programming (M2).

Proposition 2. Optimal solution of the binary integer programming (M3) is a lower bound for optimal solution of the mixed integer programming (M2).

When each of the candidate items (i.e. cargoes) involves only one limited resource, it is easy to pick up more profitable items. When there exists more than one limited resource in the candidate items, it is difficult to evaluate the amount of limited resources used. Toyoda (1975) introduced the concept of penalty vector, which results in a single index based on the many limited resources. Toyoda's heuristic for the Multi-Dimensional Knapsack Problem (MDKP) starts with no items (or all *x*'s being zero), and adds one item at a time iteratively as long as the solution is feasible. At any iteration, the heuristic picks the item which provides the maximum value per unit of aggregate resource or the effective gradient among the not-yet-picked items. Computational experiments show that this heuristic provides very good near optimal solutions to the MDKP using very short computation time, and it can be applied to large problem instances, such as problems with more than a thousand variables (see Toyoda (1975)). Following his approach, we propose a heuristic algorithm HTSS that provides a near-optimal solution to (M2) by means of concepts such as penalty vector and effective gradient introduced in Toyoda (1975).

The heuristic algorithm for the mixed integer programming (M2) is presented as follows.

Algorithm HTSS:

Step 1: Initialization.

Step 1.1: Let $\hat{z} \leftarrow 0$, $z \leftarrow 0$, $x_{tkdj_k} \leftarrow 0$, $\hat{x}_{tkdj_k} \leftarrow 0$, $\forall\, t \in \widetilde{T}, k \in \widetilde{K}_t, d \in \{t, t+1, \cdots, \tau_k\}$.

Step 1.2: Let $K_U \leftarrow \phi$, where K_U is the set of accepted items.

Step 1.3: Assign all items to $K_D = K - K_U$, where K_D is the set of items not in K_U and $K = \{k : k \in \widetilde{K}_t, t \in \widetilde{T}\}$.

Step 1.4: Let $A_U^{dj} \leftarrow (0, 0)$, where A_U^{dj} is the total quantity vector of accepted items shipping to port j in period d.

Step 2: Let

$$K_C \leftarrow \{k : k \in K_D, \exists d \le \tau_k \; s.t. (\overline{v}^d_{tk\tau_k j_k}, \overline{w}^d_{tk\tau_k j_k}) \le (1,1) - A_U^{dj_k}, v_{tk\tau_k j_k} \le E_d - \sum_{\substack{k' \in K_U \\ \tau_{k'} \ge d}} v_{tk'\tau_k j_k}\},$$

where K_C is the set of candidate items.

Step 3: Check K_C. If K_C is empty, goto Step 7. Otherwise, proceed to the next step.

Step 4: Let $\overline{K}_C \leftarrow \{(k,d) : k \in K_D, (\overline{v}^d_{tk\tau_k j_k}, \overline{w}^d_{tk\tau_k j_k}) \le (1,1) - A_U^{dj_k}, v_{tk\tau_k j_k} \le E_d - \sum_{\substack{k' \in K_U \\ \tau_{k'} \ge d}} v_{tk'\tau_k j_k}\}$.

Compute effective gradients for the items in K_C as follows.

Step 4.1: If $A_U^{dj_k}$ is a zero vector, then we set

$$G_{kd} \leftarrow \frac{\sqrt{2}\, v_{tk\tau_k j_k} r_{tkdj_k}}{\overline{v}^d_{tk\tau_k j_k} + \overline{w}^d_{tk\tau_k j_k}}, \text{ for } (k,d) \in \overline{K}_C$$

Step 4.2: Otherwise, we set

$$G_{kd} \leftarrow \frac{v_{tk\tau_k j_k} r_{tkdj_k} \sqrt{(\sum_{k' \in K_U^d} \overline{v}^d_{tk'\tau_k j_{k'}})^2 + (\sum_{k' \in K_U^d} \overline{w}^d_{tk'\tau_k j_{k'}})^2}}{\overline{v}^d_{tk\tau_k j_k} \sum_{k' \in K_U^d} \overline{v}^d_{tk'\tau_k j_{k'}} + \overline{w}^d_{tk\tau_k j_k} \sum_{k' \in K_U^d} \overline{w}^d_{tk'\tau_k j_{k'}}}, \text{ for } (k,d) \in \overline{K}_C,$$

where $K_U^d = \{k' : k' \in K_U, x_{tk'dj_{k'}} = 1\}$.

Step 5: Find that item k whose effective gradient is the largest in a period, i.e., $G_{kd} \leftarrow \max\{G_{k'd'} : (k', d') \in \overline{K}_C\}$.

Step 6: Accept k. Let $K_U \leftarrow K_U + \{k\}$, $A_U^{dj_k} \leftarrow A_U^{dj_k} + (\overline{v}^d_{tk\tau_k j_k}, \overline{w}^d_{tk\tau_k j_k})$, $z \leftarrow z + v_{tk\tau_k j_k} r_{tkdj_k}$,

$K_D \leftarrow K_D - \{k\}$, $x_{tkdj_k} \leftarrow 1$, and $G_{k'd'} \leftarrow 0$, for all $(k',d') \in \overline{K}_C$. Then, goto Step 2.

Step 7: Let

$$y_{d5}^s \leftarrow E_d^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad s \in \widetilde{S},$$

$$y_{dj1}^s \leftarrow V_{dj}^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad j \in \widetilde{J}, \quad s \in \widetilde{S},$$

$$y_{dj3}^s \leftarrow W_{dj}^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad j \in \widetilde{J}, \quad s \in \widetilde{S},$$

$$z \leftarrow z - \sum_{s=1}^{S} p^s \sum_{d=1}^{T} (q_{d5}^s y_{d5}^s + \sum_{j=1}^{J} (q_{dj1}^s y_{dj1}^s + q_{dj3}^s y_{dj3}^s)),$$

$$\hat{z} \leftarrow z, \ \hat{x}_{tkdj_k} \leftarrow x_{tkdj_k}, \text{ for } t \in \widetilde{T}, \quad k \in \widetilde{K}_t, \quad d \in \{t, t+1, \cdots, \tau_k\}.$$

Step 8: If $K_D = \phi$, the procedure terminates. Otherwise, let $z \leftarrow 0$, $y_{d5}^s \leftarrow 0$, $y_{d6}^s \leftarrow 0$, $y_{dji}^s \leftarrow 0$, for $s \in \widetilde{S}$, $d \in \widetilde{T}$, $j \in \widetilde{J}$, $i = 1,2,3,4$, and proceed to the next step.

Step 9: we set

$$G_{kd} \leftarrow \frac{\sqrt{0.125}\, v_{tk\tau_k j_k} r_{tkdj_k}}{\overline{v}_{tk\tau_k j_k}^d + \overline{w}_{tk\tau_k j_k}^d}, \text{ for } k \in K_D, \ d \in \{t, t+1, \cdots, \tau_k\}.$$

Step 10: Find that item k whose effective gradient is the largest in a period, i.e., $G_{kd} \leftarrow \max\{G_{k'd'} : k' \in K_D, d' \in \{t, t+1, \cdots, \tau_k\}\}$.

Step 11: Let $K_D \leftarrow K_D - \{k\}$, $x_{tkdj_k} \leftarrow 1$.

Step 12: For $d \in \widetilde{T}, \quad s \in \widetilde{S}$,

$$\text{if } E_d^s \geq \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k}, \text{ then } y_{d5}^s \leftarrow E_d^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k},$$

$$\text{otherwise, } y_{d6}^s \leftarrow \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k} - E_d^s;$$

$$\text{if } V_{dj}^s \geq \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} \text{, then } y_{dj1}^s \leftarrow V_{dj}^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} ,$$

$$\text{otherwise, } y_{dj2}^s \leftarrow \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} - V_{dj}^s ;$$

$$\text{if } W_{dj}^s \geq \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} \text{, then } y_{dj3}^s \leftarrow W_{dj}^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} ,$$

$$\text{otherwise, } y_{dj4}^s \leftarrow \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} - W_{dj}^s ;$$

$$z \leftarrow \sum_{t=1}^{T} \sum_{k \in \widetilde{K}_t} \sum_{d=t}^{\tau_k} v_{tk\tau_k j_k} r_{tkdj_k} x_{tkdj_k} - \sum_{s=1}^{S} p^s \sum_{d=1}^{T} (q_{d5}^s y_{d5}^s + q_{d6}^s y_{d6}^s + \sum_{j=1}^{J} \sum_{i=1}^{4} q_{dji}^s y_{dji}^s)$$

Step 13: if $\hat{z} < z$, then $\hat{z} \leftarrow z$, $\hat{x}_{tkdj_k} \leftarrow x_{tkdj_k}$. Otherwise, $x_{tkdj_k} \leftarrow 0$.

Step 14: If $K_D = \phi$, the procedure terminates. Otherwise, let $z \leftarrow 0$, $y_{d5}^s \leftarrow 0$, $y_{d6}^s \leftarrow 0$, $y_{dji}^s \leftarrow 0$, for $s \in \widetilde{S}$, $d \in \widetilde{T}$, $j \in \widetilde{J}$, $i = 1,2,3,4$, and goto Step 10.

Similar with the algorithm in Toyoda (1975), HTSS also starts with no items, and add one item at a time iteratively as long as the solution is feasible. The profitability of item k in period d is evaluated by the effective gradient G_{kd}. The effective gradient G_{kd} in HTSS is some different from G_k introduced in Toyoda (1975), the value of G_{kd} depends on item k and its shipment period d. For an item k, it has different effective gradient G_{kd} in different period d. The effective gradient G_{kd} is a rate of profit per unit of aggregate necessary resources and can be used as a single index of the profitability based on the different limited resources for item k in period d. To solve (M2), evaluate the profitability of each item in each period and accept the most profitable one, repeating this procedure as long as (8)-(12) still hold. The solution obtained by HAEG is usually a very well approximation of the optimal solution.

We present a simple example to illustrate the above algorithm.

Example: Let $\hat{r}_{tkdj_k} = v_{tk\tau_k j_k} r_{tkdj_k}$, $\widetilde{T} = \{1, 2\}$, $\widetilde{J} = \{1, 2\}$, $\widetilde{K}_1 = \{1, 2, 3, 4\}$, $\widetilde{K} = \widetilde{K}_1$,

$\widetilde{S}=\{1,2\}$; $j_1=j_2=1$, $j_3=j_4=2$, $\tau_k=2$, for $1\le k\le 4$, $p^s=1/2$, for $s\in\widetilde{S}$;

$(V_{11}^1,V_{12}^1,V_{21}^1,V_{22}^1,V_{11}^2,V_{12}^2,V_{21}^2,V_{22}^2)=(2076, 2018, 2018, 2160, 2018, 2160, 2160, 2102)$,

$(W_{11}^1,W_{12}^1,W_{21}^1,W_{22}^1,W_{11}^2,W_{12}^2,W_{21}^2,W_{22}^2)=(2038, 2086, 2086, 2134, 2086, 2134, 2134, 2182)$,

$(\hat{r}_{1111},\hat{r}_{1211},\hat{r}_{1312},\hat{r}_{1412},\hat{r}_{1121},\hat{r}_{1221},\hat{r}_{1322},\hat{r}_{1422})=(202.41,50.45,437.4,273.26,51.9,408.65,280.8,120.87)$,

$(v_{1121},v_{1221},v_{1322},v_{1422})=(1038,1009,1080,1051)$, $(w_{1121},w_{1221},w_{1322},w_{1422})=(1019,1043,1067,1091)$,

$(E_1^1,E_2^1,E_1^2,E_2^2)=(2076,2018,2018,2160)$, $(q_{15}^1,q_{16}^1,q_{15}^2,q_{16}^2)=(0.013,0.02,0.003,0.044)$,

$(q_{25}^1,q_{26}^1,q_{25}^2,q_{26}^2)=(0.003,0.044,0.027,0.068)$, $(q_{111}^1,q_{112}^1,q_{113}^1,q_{114}^1)=(0.013,0.06,0.013,0.114)$,

$(q_{111}^2,q_{112}^2,q_{113}^2,q_{114}^2)=(0.003,0.132,0.028,0.282)$, $(q_{211}^1,q_{212}^1,q_{213}^1,q_{214}^1)=(0.003,0.132,0.028,0.282)$,

$(q_{211}^2,q_{212}^2,q_{213}^2,q_{214}^2)=(0.027,0.204,0.021,0.249)$, $(q_{121}^1,q_{122}^1,q_{123}^1,q_{124}^1)=(0.003,0.132,0.028,0.282)$,

$(q_{121}^2,q_{122}^2,q_{123}^2,q_{124}^2)=(0.027,0.204,0.021,0.249)$, $(q_{221}^1,q_{222}^1,q_{223}^1,q_{224}^1)=(0.027,0.204,0.021,0.249)$,

$(q_{221}^2,q_{222}^2,q_{223}^2,q_{224}^2)=(0.017,0.276,0.003,0.213)$.

The procedure following the above heuristic algorithm HTSS is briefly given as follows.

By $\overline{v}_{tk\tau_k j_k}^d=v_{tk\tau_k j_k}/V_{dj_k}$, $\overline{w}_{tk\tau_k j_k}^d=w_{tk\tau_k j_k}/W_{dj_k}$, we have

$(\overline{v}_{1121}^1,\overline{v}_{1221}^1,\overline{v}_{1322}^1,\overline{v}_{1422}^1,\overline{v}_{1121}^2,\overline{v}_{1221}^2,\overline{v}_{1322}^2,\overline{v}_{1422}^2)=(0.513,0.5,0.535,0.52,0.513,0.5,0.514,0.5)$,

$(\overline{w}_{1121}^1,\overline{w}_{1221}^1,\overline{w}_{1322}^1,\overline{w}_{1422}^1,\overline{w}_{1121}^2,\overline{w}_{1221}^2,\overline{w}_{1322}^2,\overline{w}_{1422}^2)=(0.5,0.512,0.512,0.523,0.489,0.5,0.512,0.5)$

Step 1: Let $K_U\leftarrow\phi$, $\hat{z}\leftarrow 0$, $z\leftarrow 0$, $x_{tkdj_k}\leftarrow 0$, $\hat{x}_{tkdj_k}\leftarrow 0$, $\forall\, t\in\widetilde{T}, k\in\widetilde{K}_1, d\in\{1,2\}$.

Assign all items to $K_D=K-K_U$. Let $A_U^{dj}\leftarrow(0,0)$.

Step 2. Let

$$K_C\leftarrow\{k:k\in K_D,\exists d\le\tau_k\ s.t.(\overline{v}_{tk\tau_k j_k}^d,\overline{w}_{tk\tau_k j_k}^d)\le(1,1)-A_U^{dj_k},\ v_{tk\tau_k j_k}\le E_d-\sum_{\substack{k'\in K_U\\ \tau_{k'}\ge d}}v_{tk'\tau_k j_k}\}.$$

Step 3. Since $K_C\ne\phi$, proceed to the next step.

Step 4. Since $A_U^{dj_k}=(0,0)$, the effective gradients are calculated by formula:

$$G_{kd}=\frac{\hat{r}_{tkdj_k}\sqrt{2}}{\overline{v}_{tk\tau_k j_k}^d+\overline{w}_{tk\tau_k j_k}^d},\ \text{for } (k,d)\in\overline{K}_C,$$

we have $G_{11}=282.2$, $G_{21}=70.52$, $G_{31}=590.99$, $G_{41}=370.22$, $G_{12}=73.19$, $G_{22}=577.91$, $G_{32}=391.71$, $G_{42}=169.03$.

Step 5. The largest effective gradient is $G_{31} = 590.99$.

Step 6. Accept item 3. Let $K_U \leftarrow K_U + \{3\}$, $A_U^{11} \leftarrow A_U^{11} + (\bar{v}_{1322}^1, \overline{w}_{1322}^1)$, $z \leftarrow z + \hat{r}_{1312}$, $K_D \leftarrow K_D - \{3\}$, $x_{1312} \leftarrow 1$. Then, go to step 2.

After it is repeated several times, K_C becomes empty and we obtain the final solution of (M3): $z = 846.05$, $x_{1221} = x_{1312} = 1$, $x_{1111} = x_{1121} = x_{1211} = x_{1322} = x_{1412} = x_{1422} = 0$.

Step 7: We have $y_{15}^1 = 996$, $y_{25}^1 = 1009$, $y_{15}^2 = 938$, $y_{25}^2 = 1151$, $y_{111}^1 = 2076$, $y_{121}^1 = 938$, $y_{211}^1 = 1009$, $y_{221}^1 = 2160$, $y_{111}^2 = 2018$, $y_{121}^2 = 980$, $y_{211}^2 = 1151$, $y_{221}^2 = 2102$, $y_{113}^1 = 2038$, $y_{123}^1 = 1019$, $y_{213}^1 = 1043$, $y_{223}^1 = 2134$, $y_{113}^2 = 2086$, $y_{123}^2 = 1067$, $y_{213}^2 = 1091$, $y_{223}^2 = 2182$, $z = 604.2$. We set $\hat{z} \leftarrow z$, $\hat{x}_{1kdj_k} \leftarrow x_{1kdj_k}$, for $t \in \widetilde{T}, k \in \widetilde{K}_1, d \in \{1, 2\}$.

Step 8: Since $K_D = \{1, 4\}$, let $z \leftarrow 0$, $y_{d5}^s \leftarrow 0$, $y_{d6}^s \leftarrow 0$, $y_{dji}^s \leftarrow 0$, for $s \in \widetilde{S}$, $d \in \widetilde{T}$, $j \in \widetilde{J}$, $i = 1,2,3,4$, and proceed to the next step.

Step 9: We have $G_{11} = 70.55$, $G_{12} = 18.3$, $G_{41} = 92.56$, $G_{42} = 42.26$

Step 10: The largest effective gradient is $G_{41} = 92.56$.

Step 11: Let $K_D \leftarrow K_D - \{4\}$, $x_{1412} \leftarrow 1$.

Step 12: We have $y_{16}^1 = 55$, $y_{25}^1 = 1009$, $y_{16}^2 = 113$, $y_{25}^2 = 1151$, $y_{111}^1 = 2076$, $y_{122}^1 = 113$, $y_{211}^1 = 1009$, $y_{221}^1 = 2160$, $y_{111}^2 = 2018$, $y_{121}^2 = 29$, $y_{211}^2 = 1151$, $y_{221}^2 = 2102$, $y_{113}^1 = 2038$, $y_{124}^1 = 72$, $y_{213}^1 = 1043$, $y_{223}^1 = 2134$, $y_{113}^2 = 2086$, $y_{124}^2 = 24$, $y_{213}^2 = 1091$, $y_{223}^2 = 2182$, $z = 901.9$.

Step 13: Since $\hat{z} < z$, we set $\hat{z} \leftarrow z$, $\hat{x}_{1221} \leftarrow x_{1221}$.

Step 14: Since $K_D = \{1\}$, we set $z \leftarrow 0$, $y_{d5}^s \leftarrow 0$, $y_{d6}^s \leftarrow 0$, $y_{dji}^s \leftarrow 0$, for $s \in \widetilde{S}$, $d \in \widetilde{T}$, $j \in \widetilde{J}$, $i = 1,2,3,4$, and goto Step 10.

After it is repeated, K_D becomes empty and we obtain the final solution:

$z = 1100.63$, $x_{1111} = x_{1221} = x_{1312} = x_{1412} = 1$, and other $x_{1kdj_k} = 0$; $y_{16}^1 = 1093$, $y_{25}^1 = 1009$, $y_{16}^2 = 1151$, $y_{25}^2 = 1151$, $y_{111}^1 = 1038$, $y_{122}^1 = 113$, $y_{211}^1 = 1009$, $y_{221}^1 = 2160$, $y_{111}^2 = 980$, $y_{121}^2 = 29$, $y_{211}^2 = 1151$, $y_{221}^2 = 2102$, $y_{113}^1 = 1019$, $y_{124}^1 = 72$, $y_{213}^1 = 1043$,

$y_{223}^{1} = 2134$, $y_{113}^{2} = 1067$, $y_{124}^{2} = 24$, $y_{213}^{2} = 1091$, $y_{223}^{2} = 2182$, and other $y_{di}^{s} = y_{dji}^{s} = 0$.

In fact, this solution is the same as the optimal one obtained by complete enumeration.

3.2 Modified Heuristic Algorithm

In this section, we modify the effective gradient G_{kd} , which introduced in Section 3.1 to improve the accuracy of the heuristic algorithm. Nine different methods are used, each having a different effective gradient. When a better approximation to the optimal solution is desired, solutions may be obtained at several different values of the effective gradient and the maximum selected from among them.

Algorithm MHTSS:

Step 1: Initialization.

Step 1.1: Let $l \leftarrow 0$, $\hat{z} \leftarrow 0$, $\hat{x}_{tkdj_k} \leftarrow 0, \forall\, t \in \widetilde{T}, k \in \widetilde{K}_t, d \in \{t, t+1, \cdots, \tau_k\}$.

Step 1.2: Let $K_U \leftarrow \phi$, where K_U is the set of accepted items.

Step 1.3: Assign all items to $K_D = K - K_U$, where K_D is the set of items not in K_U and

$$K = \{k : k \in \widetilde{K}_t, t \in \widetilde{T}\} .$$

Step 1.4: Let $A_U^{dj} \leftarrow (0, 0)$, where A_U^{dj} is the total quantity vector of accepted items shipping to port j in period d .

Step 1.5: Let the objective value be zero, i.e., $z \leftarrow 0$.

Step 1.6: Let $x_{tkdj_k} \leftarrow 0, \forall\, t \in \widetilde{T}, k \in \widetilde{K}_t, d \in \{t, t+1, \cdots, \tau_k\}$.

Step 2: Let

$$K_C \leftarrow \{k : k \in K_D, \exists d \le \tau_k \; s.t. (\overline{v}_{tk\tau_k j_k}^{d}, \overline{w}_{tk\tau_k j_k}^{d} \le (1,1) - A_U^{dj_k}, v_{tk\tau_k j_k} \le E_d - \sum_{\substack{k' \in K_U \\ \tau_{k'} \ge d}} v_{tk'\tau_k j_k}\} ,$$

where K_C is the set of candidate items.

Step 3: Check K_C . If K_C is empty, goto Step 7. Otherwise, proceed to the next step.

Step 4: Let $\overline{K}_C = \{(k,d) : k \in K_D, (\overline{v}_{tk\tau_k j_k}^{d}, \overline{w}_{tk\tau_k j_k}^{d}) \le (1,1) - A_U^{dj_k}, v_{tk\tau_k j_k} \le E_d - \sum_{\substack{k' \in K_U \\ \tau_{k'} \ge d}} v_{tk'\tau_k j_k}\}$.

Compute effective gradients for the items in K_C as follows.

Step 4.1: If $A_U^{dj_k}$ is a zero vector, then we set

$$G_{kd} \leftarrow \frac{\sqrt{2}\, v_{tk\tau_k j_k} r_{tkdj_k}}{\overline{v}^d_{tk\tau_k j_k} + \overline{w}^d_{tk\tau_k j_k}}, \text{ for } (k,d) \in \overline{K}_C$$

Step 4.2: Otherwise, let $K_U^d = \{k' : k' \in K_U,\, x_{tk'dj_{k'}} = 1\}$, for $l = 0, 1, 2$, we set

$$\widetilde{V}_d \leftarrow \max\Big(\sum_{k' \in K_U^d} \overline{v}^d_{tk'\tau_k j_{k'}} - \alpha \cdot \max\Big(\sum_{k' \in K_U^d} \overline{v}^d_{tk'\tau_k j_{k'}}, \sum_{k' \in K_U^d} \overline{w}^d_{tk'\tau_k j_{k'}}\Big), 0\Big),$$

$$\widetilde{W}_d \leftarrow \max\Big(\sum_{k' \in K_U^d} \overline{w}^d_{tk'\tau_k j_{k'}} - \alpha \cdot \max\Big(\sum_{k' \in K_U^d} \overline{v}^d_{tk'\tau_k j_{k'}}, \sum_{k' \in K_U^d} \overline{w}^d_{tk'\tau_k j_{k'}}\Big), 0\Big),$$

$$G_{kd} \leftarrow \frac{v_{tk\tau_k j_k} r_{tkdj_k} \sqrt{(\widetilde{V}_d)^2 + (\widetilde{W}_d)^2}}{\overline{v}^d_{tk\tau_k j_k} \widetilde{V}_d + \overline{w}^d_{tk\tau_k j_k} \widetilde{W}_d},$$

for $(k,d) \in \overline{K}_C$, where $\alpha = 0,\, 0.2,\, 0.9$ respectively, for $l = 0, 1, 2$;

for $l = 3$, we set

$$\widetilde{V}_d \leftarrow \max\Big(\sum_{k' \in K_U^d} \overline{v}^d_{tk'\tau_k j_{k'}} - \Big(\max\Big(\sum_{k' \in K_U^d} \overline{v}^d_{tk'\tau_k j_{k'}}, \sum_{k' \in K_U^d} \overline{w}^d_{tk'\tau_k j_{k'}}\Big)\Big)^2, 0\Big),$$

$$\widetilde{W}_d \leftarrow \max\Big(\sum_{k' \in K_U^d} \overline{w}^d_{tk'\tau_k j_{k'}} - \Big(\max\Big(\sum_{k' \in K_U^d} \overline{v}^d_{tk'\tau_k j_{k'}}, \sum_{k' \in K_U^d} \overline{w}^d_{tk'\tau_k j_{k'}}\Big)\Big)^2, 0\Big),$$

$$G_{kd} \leftarrow \frac{v_{tk\tau_k j_k} r_{tkdj_k} \sqrt{(\widetilde{V}_d)^2 + (\widetilde{W}_d)^2}}{\overline{v}^d_{tk\tau_k j_k} \widetilde{V}_d + \overline{w}^d_{tk\tau_k j_k} \widetilde{W}_d}, \text{ for } (k,d) \in \overline{K}_C;$$

for $l = 4, 5, 6, 7, 8$, we set

$$G_{kd} \leftarrow \frac{\sqrt{\alpha}\, v_{tk\tau_k j_k} r_{tkdj_k}}{\overline{v}^d_{tk\tau_k j_k} + \overline{w}^d_{tk\tau_k j_k}}, \quad \text{for } (k,d) \in \overline{K}_C, \quad \text{where } \alpha = 0.125,\, 0.25, 2,\, 1.2,\, 1$$

respectively.

Step 5: Find that item k whose effective gradient is the largest in a period, i.e.,

$$G_{kd} = \max\{G_{k'd'} : (k', d') \in \overline{K}_C\}.$$

Step 6: Accept k. Let $K_U \leftarrow K_U + \{k\}$, $A_U^{dj_k} \leftarrow A_U^{dj_k} + (\overline{v}^d_{tk\tau_k j_k}, \overline{w}^d_{tk\tau_k j_k})$, $z \leftarrow z + v_{tk\tau_k j_k} r_{tkdj_k}$,

$K_D \leftarrow K_D - \{k\}$, $x_{tkdj_k} \leftarrow 1$. Then, goto Step 2.

Step 7: If $z > \hat{z}$, then $\hat{z} \leftarrow z$, $\hat{x}_{tkdj_k} \leftarrow x_{tkdj_k}$, for $t \in \widetilde{T}$, $\quad k \in \widetilde{K}_t$, $\quad d \in \{t, t+1, \cdots, \tau_k\}$.

Step 8: If $l = 8$, then goto Step 9. Otherwise, let $l \leftarrow l + 1$, goto Step 1.1.

Step 9: Let $z \leftarrow \hat{z}$, $x_{tkdj_k} \leftarrow \hat{x}_{tkdj_k}$, for $t \in \widetilde{T}, k \in \widetilde{K}_t, d \in \{t, t+1, \cdots, \tau_k\}$, we set

$$y_{d5}^s \leftarrow E_d^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad s \in \widetilde{S},$$

$$y_{dj1}^s \leftarrow V_{dj}^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad j \in \widetilde{J}, \quad s \in \widetilde{S},$$

$$y_{dj3}^s \leftarrow W_{dj}^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} \quad \forall d \in \widetilde{T}, \quad j \in \widetilde{J}, \quad s \in \widetilde{S},$$

$$z \leftarrow z - \sum_{s=1}^{S} p^s \sum_{d=1}^{T} (q_{d5}^s y_{d5}^s + \sum_{j=1}^{J} (q_{dj1}^s y_{dj1}^s + q_{dj3}^s y_{dj3}^s)),$$

$$\hat{z} \leftarrow z, \ \hat{x}_{tkdj_k} \leftarrow x_{tkdj_k}, \text{ for } t \in \widetilde{T}, \quad k \in \widetilde{K}_t, \quad d \in \{t, t+1, \cdots, \tau_k\}.$$

Step 10: If $K_D = \phi$, the procedure terminates. Otherwise, let $z \leftarrow 0$, $y_{d5}^s \leftarrow 0$, $y_{d6}^s \leftarrow 0$, $y_{dji}^s \leftarrow 0$, for $s \in \widetilde{S}$, $d \in \widetilde{T}$, $j \in \widetilde{J}$, $i = 1,2,3,4$, and proceed to the next step.

Step 11: we set

$$G_{kd} \leftarrow \frac{\sqrt{0.125}\, v_{tk\tau_k j_k} r_{tkdj_k}}{\overline{v}_{tk\tau_k j_k}^d + \overline{w}_{tk\tau_k j_k}^d}, \text{ for } k \in K_D, \ d \in \{t, t+1, \cdots, \tau_k\}.$$

Step 12: Find that item k whose effective gradient is the largest in a period, i.e., $G_{kd} \leftarrow \max\{G_{k'd'} : k' \in K_D, d' \in \{t, t+1, \cdots, \tau_k\}\}$.

Step 13: Let $K_D \leftarrow K_D - \{k\}$, $x_{tkdj_k} \leftarrow 1$.

Step 14: For $d \in \widetilde{T}, \quad s \in \widetilde{S}$,

$$\text{if } E_d^s \geq \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k}, \text{ then } y_{d5}^s \leftarrow E_d^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k},$$

$$\text{otherwise, } y_{d6}^s \leftarrow \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d}} v_{tk\tau_k j_k} x_{tkdj_k} - E_d^s;$$

$$\text{if } V_{dj}^s \geq \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} \text{, then } y_{dj1}^s \leftarrow V_{dj}^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} \text{ ,}$$

$$\text{otherwise, } y_{dj2}^s \leftarrow \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} v_{tk\tau_k j_k} x_{tkdj_k} - V_{dj}^s \text{;}$$

$$\text{if } W_{dj}^s \geq \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} \text{, then } y_{dj3}^s \leftarrow W_{dj}^s - \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} \text{ ,}$$

$$\text{otherwise, } y_{dj4}^s \leftarrow \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_t \\ \tau_k \geq d \\ j_k = j}} w_{tk\tau_k j_k} x_{tkdj_k} - W_{dj}^s \text{;}$$

$$z \leftarrow \sum_{t=1}^{T} \sum_{k \in \widetilde{K}_t} \sum_{d=t}^{\tau_k} v_{tk\tau_k j_k} r_{tkdj_k} x_{tkdj_k} - \sum_{s=1}^{S} p^s \sum_{d=1}^{T} (q_{d5}^s y_{d5}^s + q_{d6}^s y_{d6}^s + \sum_{j=1}^{J} \sum_{i=1}^{4} q_{dji}^s y_{dji}^s)$$

Step 15: if $\hat{z} < z$, then $\hat{z} \leftarrow z$, $\hat{x}_{tkdj_k} \leftarrow x_{tkdj_k}$. Otherwise, $x_{tkdj_k} \leftarrow 0$.

Step 16: If $K_D = \phi$, the procedure terminates. Otherwise, let $z \leftarrow 0$, $y_{d5}^s \leftarrow 0$, $y_{d6}^s \leftarrow 0$, $y_{dji}^s \leftarrow 0$, for $s \in \widetilde{S}$, $d \in \widetilde{T}$, $j \in \widetilde{J}$, $i = 1,2,3,4$, and goto Step 12.

From the steps of the modified heuristic algorithm, we may easily notice that the solution of MHTSS is closer to the optimal solution than HTSS, but the computation time of MHTSS is longer than HTSS, because we take the maximum value among solutions given by nine methods for MHTSS including the method used in HTSS.

4. Numerical Experiments

In this section, we implement the modified heuristic algorithm MHTSS and compare its solution to optimal solution or LP (relaxation) optimal solution (as the upper bound for optimal solution). The algorithm has been coded in C++ and run under Microsoft Windows Server 2003 Standard Edition using a Server (Intel(R) Xeon(TM) CPU 3.06GHz and 1.0GB of RAM). CPU times were obtained through the C++ function clock(). To conduct our experiments we used randomly generated instances. For simplicity of implementation, we assumed that destination port j_k of each cargo k is decided by:

$j_k = j$, if $\text{floor(K/J)} \times (\text{j} - 1) \le \text{k} < \text{floor(K/J)} \times \text{j}$, for $j = 1, 2, \cdots, J-1$,

$j_k = J$, if $\text{floor(K/J)} \times (\text{J} - 1) < \text{k} \le J$,

where floor(x) is a function of C++ which returns a floating-point value representing the largest integer that is less than or equal to x.

For each set of parameters J, T and K, we generated 10 random small scale instances, for which optimal solutions can be obtained by CPLEX 8.0. We tested heuristic solutions and optimal solutions or LP optimal solutions for all 10 instances, and tabulated the average relative gap and average computation time. In table 1, the relative gap between heuristic solution and optimal solution g_O is computed as

$$\frac{\text{total profit of optimal solution - total profit of heuristic solution}}{\text{total profit of optimal solution}} \times 100\% .$$

In table 1 and table 2, the relative gap between heuristic solution and LP relaxation optimal solution g_L is computed as

$$\frac{\text{total profit of LP relaxation optimal solution - total profit of heuristic solution}}{\text{total profit of LP relaxation optimal solution}} \times 100\% .$$

{Insert here the Table 1}

Table 1 shows the results obtained for a set of small test problems. Test problems 1 have 2 scenarios, 2 periods, 2 destination ports and 150 items (cargoes); test problem 2 has 3 scenarios, 3 periods, 4 destination ports and 71 items, and so on. For comparison, the optimal solution has been computed using CPLEX 8.0. As can be seen from table 1, the obtained results seem to be encouraging. The gap between the optimal solution and the heuristic solution is small and the computation time is very short. Table 2 shows the results obtained for a set of medium scale problems.

{Insert here the Table 2}

In addition, table 3 shows the computation times for a set of large scale problems, for which the optimal solution is obvious not a feasible alternative. It shows that the computation time of the heuristic algorithm is very short even for very large scale problems with tens of thousands decision variables.

{Insert here the Table 3}

From our preliminary computation experiment, we believe that heuristic algorithm would be a very good candidate for solving the problem in time critical or real-time applications such as sea cargo mix problems where a near optimal solution is acceptable, and fast computation is more important than guaranteeing optimal value.

5. Conclusions

We have formulated the multi-period sea cargo mix problem as the two-stage stochastic mixed integer programming model, and presented effective heuristic algorithms which provide fast and near optimal solution. We also presented experimental results to evaluate the algorithm using a wide range of problem instances. The results strongly suggest that the heuristic algorithm is very effective for time critical tactical or operations level decisions, where a near optimal solution is acceptable and fast computation is more important than guaranteeing optimal value.

Acknowledgement. This research is supported in part by the Academic Research Fund and the Centre for E-Business of National University Singapore, and the Strategic Research Programme (SRP) sponsored by Agency for Science, Technology and Research (A*STAR) of Singapore.

References

R. D. ARMSTRONG, D. S. KUNG, P. SINHA, AND A. A. ZOLTNERS, "A Computational Study of a Multiple choice Knapsack Algorithm," *ACM Transactions on Mathematical Software* **9**, 184-198 (1983).

C.C. CARØE AND J. TIND, "A cutting-plane approach to mixed 0 − 1 stochastic integer programs," *European Journal of Operational Research* **101(2),** 306–316 (1997).

R. K. CHEUNG AND C.-Y. CHEN, "A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem," *Thansp. Sci.* **32(2)**, 142-162 (1998).

S.-C. CHO AND A. N. PERAKIS, "Optimal liner fleet routing strategies", *MARIT. POL. MGMT.* **23(3)**, 249-259(1996).

A. CIANCIMINO AND G. INZERILLO, "A mathematical programming approach for the solution of the railway yield management problem," *Thansp. Sci.* **33(2)**, 168-181 (1999).

T. G. CRAINIC, M. GENDREAU, AND P. DEJAX, "Dynamic and Stochastic models for the allocation of empty containers," *Operations Research* **41(1)**, 102-126 (1993).

M. CHRISTIANSEN AND K. FAGERHOLT, "Robust ship scheduling with multiple time windows," *Naval Research Logistics* **49**, 611-625 (2002).

J. J. EVANS AND P. MARLOW, *Quantitative Methods in Maritime Economics*, Fairplay Publications (1985).

R. G. KASILINGAM, "Air cargo revenue management: characteristics and complexities," *Euro. J. of Opns. Res.* **96(1)**, 36-44 (1996).

W.K. KLEIN HANEVELD AND M.H. VAN DER VLERK, "Stochastic integer programming: general models and algorithms," *Annals of Operations Research* **85**, 39-57 (1999).

K. K. LAI, K. LAM, AND W. K. CHAN, "Ship container logistics and allocation," *Journal of the Operational Research Society* **46(6)**, 687-697 (1995).

E. F. MAGIROU, H. N. PSARAFTIS, AND N. M. CHRISTODOULAKIS, "Quantitative methods in shipping: a survey of current use and future trends," *ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS*, Report No. **E115** (1992).

R. PESENTI, "Hierarchical resource planning for shipping companies," *European Journal of Operational Research* **86**, 91-102 (1995).

T. A. POWELL AND T. A. CARVALHO, "Dynamic control of logistics queueing networks for large-scale fleet management," *Transportation Science* **32(2)**, 90-109 (1998)

K. RANA AND R. G. VICKSON, "Routing container ships using Lagrangean relaxation and decomposition," *Transportation Science* **25(3)**, 201-214 (1991).

R. SCHULTZ, "On structure and stability in stochastic programs with random technology matrix and complete integer recourse," *Mathematical Programming* **70(1)**, 73–89

(1995).

W. SHIH, “A branch and bound method for the multiconstraint zero-one knapsack problem,” *J. Opl. Res. Soc.* **30(4)**, 369-378(1979).

Y. TOYODA, “A simplified algorithm for obtaining approximate solution to zero-one programming problems,” *Management Science* **21**, 1417-1427 (1975).

I. F. A. VIS AND R. KOSTER, “Transshipment of containers at a container terminal: an overview,” *European Journal of Operational Research* **147**, 1-16 (2003).

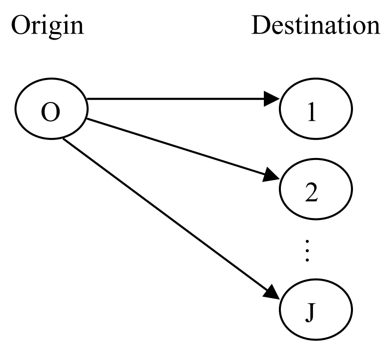

Figure 1. Network of ports and directions

S	T	J	K	Number of variables	Number of constraints	Instances tested	Average relative gap (%)	
							g_O	g_L
2	2	2	150	340	170	10	0.21	1.32
3	3	4	71	375	152	10	1.21	2.96
4	4	4	23	380	167	10	0.71	1.82
3	3	4	78	396	159	10	1.28	2.23
3	3	3	91	399	154	10	1.31	1.84
3	4	4	48	408	156	10	1.42	1.89

Table 1. Results for small test problems

S	T	J	K	Number of variables	Number of constraints	Instances tested	Average relative gap g_L (%)	Average CPU time (sec)	
								Heuristic	*LP*
3	4	4	897	3804	1005	10	1.63	11.44	1242.23
3	4	8	986	4352	1190	10	1.83	24.29	1630.06
3	4	27	972	5208	1632	10	2.31	43.05	2133.36
2	2	2	2890	5820	2910	10	0.18	32.49	1691.66
2	2	2	3870	7780	3890	10	0.13	63.11	4784.78
3	5	37	1972	12110	3097	10	1.71	304.97	36331.1

Table 2. Results for medium scale problems

S	T	J	K	Number of variables	Number of constraints	Instances tested	Average CPU time (sec)
4	3	4	6578	19950	6686	10	915.87
3	4	6	5278	21424	5434	10	715.16
3	4	14	5489	22652	5837	10	1166.01
4	10	46	1537	22810	5257	10	374.55
3	5	9	4949	25315	5234	10	971.74
3	8	27	3236	28528	4556	10	1124.89

Table 3. Computation time for large scale problems