Model and Algorithm for the Operational Level Service Route Planning Problem

Chengxuan CAO[1], James Soo Keng ANG[2], Heng-Qing YE[3]
[1] The Logistics Institute - Asia Pacific, National University of Singapore, Singapore 119260
[2,3] School of Business, National University of Singapore, Singapore 117592

ABSTRACT
In this paper, we consider the operational level service route planning problem in the container shipping industry. We formulate multi-period and multiple ships service route planning problem as the multi-dimension multiple knapsack problem (MDMKP). The MDMKP is an optimization model that maximizes the total profit for a service route in multiple periods of a planning horizon. A heuristic algorithm is proposed to obtain the near optimal solution for the problem. Numerical experiments demonstrate the efficiency of the algorithm.

1. INTRODUCTION AND PROBLEM DESCRIPTION

Container shipping needs large capital investments and operating costs. A ship involves a major capital investment (usually millions of US dollars), and the daily operating costs of a ship can be tens of thousands of dollars. On the other hand, the container shipping companies have to costly reposition empty containers due to trade imbalance. Therefore, liner operators have difficulty generating reasonable profit and even incur deficit. This means that improving fleet utilization can bring significant improvements in profitability.

Ships follow the available cargoes. Carriers may have a certain amount of contract cargoes that it is committed to carry, and tries to maximize the profit from optional cargoes. They operate according to a published itinerary and schedule. The fleet may contain various types of ships, ships of different sizes, ships with different cost structures and with different other ship-specific characteristics. Although the fleet size and mix of shipping companies may differ considerably, they have one main objective in common, namely to utilize their fleets (fixed or variable) optimally. Consequently, container shipping companies have many similar complex, extensive planning problems, ranging from the strategic and the tactical to operational levels. Typical examples are: optimal size and mix of the fleet, routing and scheduling for each ship in the fleet, and selecting the best course for a ship between two ports subject to prevailing weather and ocean currents. Operational level service route planning problem is an important problem, which is closely related to many other fleet planning problems.

When minimizing costs in ship scheduling, there are most often only the variable costs (mainly fuel and port costs) that are considered, since the fixed costs (typically capital and crew costs) will not be influenced by scheduling decisions. By optimizing cargo mix in a service route, the container shipping company can improve fleet utilization. This may increase total income significantly, while the variable costs often increase moderately.

The operational level service route planning problem arises when a carrier wants to select cargoes in order to maximize its revenue for a service route in a planning horizon. Containers loaded at one port, in general, have more than one port of destination, and may not fill a ship. Ships in this type of trade must call at more than two ports, unlike those in bulk trades. In a service route, a ship travels in one direction, loading and discharging cargoes at the intermediate and end ports. It may visit some ports on its inbound voyage and some on its outbound voyage. This depends on the amount of cargoes available at various ports. Generally, a schedule is decided for a fixed period, which depends upon a number of factors such as seasonal fluctuations, market requirements, company policy etc. availability of cargoes at each port is generally estimated on a weekly basis during a planning horizon.

Accordingly, a schedule is determined for a ship and the shippers are informed. The company may reject to transport some cargoes in a period, either because they are not profitable or because there are other cargoes, perhaps at other ports, that are relatively more profitable. Each type of cargo has different volume / weight characteristic and different freight rate, which is assumed to be given exogenously. The cargo selection should be done without violating the volume and weight constraints of the ship.

We present a model with an objective to maximize total profit, when multiple ships allotted to a service route in multiple periods. The constraints pertain to the ship-carrying capacity, available cargoes and the requirement of having a connected feasible route for each ship. The model needs data from the online booking system and container tracking system to provide current status and location of container inventory as well as all associated shipping costs. Historical data are gathered and maintained over time to permit forecasting of container flows and inventory levels. From demands, bookings, available capacity and average contributions, the model mathematically determines an optimal target contribution to determine whether to accept or reject a proposed booking, or to suggest a surcharge or alternate product or service at a higher/lower price. The result would be used to adjust contribution calculations to favor certain types of freight on certain voyages and discourage others.

The operational level service route planning problem is closely linked to the empties re-positioning. If the inbound vessel is full, and if this is not balanced by a full outbound, then there is the need to ship empties necessitated by the need to re-position the empties for use by the destination port. This too has to be taken into account in the operational level service route planning. Empties re-positioning is one of the single largest expenses for most container carriers. Many different models have been proposed in the literature to handle optimal empties re-positioning; see for example [1][2][6]. While these models are important in theory and practices and are appropriate under their contexts, they do not consider empties re-positioning with movement of laden containers and revenue analysis.

The rest of the paper is organized as follows. In Section 2, multi-period and multiple ships service route planning problem is formulated as the Multi-Dimension Multiple Knapsack Problem (MDMKP). Section 3 presents algorithm for the model. The heuristic algorithm with effective gradient is given 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. MATHEMATICAL FORMULATION

In this section, we develop a mathematical model for the problem under consideration. In the model, we have to decide on how to load cargoes on ships in a service route during a planning horizon, in order to maximize the profit generated from the shipment subject to some weight/volume constraints. A key constraint is that cargo weight and volume should not exceed the ship's capacity. In effect, the multi-period multiple ships operational level service route planning problem is formulated as the multi-dimension multiple knapsack problem (MDMKP).

Before formally stating the problem, we introduce some notation:

Index Sets

$\tilde{T}$: 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 t th time period.

$\tilde{J}$: Set of ports $\{1, 2, \cdots, i, \cdots, j, \cdots, J\}$.

$\tilde{K}$: Set of cargoes ready for shipment $\{1, 2, \cdots, k, \cdots, K\}$.

$\tilde{K}_{ti}$: Set of all cargoes that will be received at port i in period t, $\tilde{K}_{ti} \subset \tilde{K}$. This information about shipment requirement can be gathered from for example the advance shipment bookings. For convenience, when we refer to the cargo k, it is always associated with the receiving period t and loading port i. Thus, the cargo k implicitly depends on the time index t and port index i.

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

s_{di} : The ship which will call at port i in period d.

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

Parameters

V_{di} : Volume capacity of the ship which will call at port i in period d, i.e., volume capacity of ship s_{di}.

W_{di} : Weight capacity of the ship which will call at port i in period d, i.e., weight capacity of ship s_{di}.

r_{tkidj_k} : Per volume profit of cargo k which is received in period t and shipped from port i to port j_k by ship s_{di} in period d.

E_{di} : Total volume of available empty containers at port i in period d. Short of available empty containers also impacts the containerized shipping.

$v_{tki\tau_k j_k}$: Volume of cargo k received at port i in period t ready for shipment to port j_k before its due date τ_k.

Decision Variables

x_{tkidj_k} : Binary variable, i.e., $x_{tkidj_k} = 1$, if cargo k which received at port i in period t is ready for shipment to port j_k by ship s_{di} in period d before its due date τ_k, 0, otherwise.

The following assumptions are imposed for the model:

(1) The service route is started from port 1 and ended port J.
(2) There are several ships at the service route in the planning horizon.
(3) For each port in the service route, there is only one ship calling at it in a period.
(4) The ship in the service route starts its voyage at different port and ends its voyage at different port in a planning horizon.

For convenience, we only consider the outbound voyage in a service route. Although all possible calling ports in the service route are considered, not every one has to be visited and final executed routing sequence is allowed to pass some candidates over.

Under above assumptions, we easily obtain the following proposition and corollary.

Proposition: For each $d \in \widetilde{T}$ and $i \in \widetilde{J} - \{J\}$, we have $s_{di} = s_{d+1i+1}$.

Corollary: $V_{di} = V_{d+1i+1}$, $W_{di} = W_{d+1i+1}$, for each $d \in \widetilde{T}$ and $i \in \widetilde{J} - \{J\}$.

The multi-period multiple ships operational level service route planning problem can then be formulated as follows:

(M1) Maximize $z = \sum_{t=1}^{T} \sum_{i=1}^{J-1} \sum_{k \in \widetilde{K}_{ti}} \sum_{d=t}^{\tau_k} v_{tki\tau_k j_k} r_{tkidj_k} x_{tkidj_k}$ (1)

subject to

$$\sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_{ti} \\ \tau_k \geq d}} v_{tki\tau_k j_k} x_{tkidj_k} \leq E_{di} \quad \forall i \in \widetilde{J} - \{J\},\ d \in \widetilde{T} \qquad (2)$$

$$\sum_{h=1}^{i-1} \sum_{t=1}^{d-i+h} \sum_{\substack{k \in \widetilde{K}_{th} \\ \tau_k \geq d-i+h \\ h < j_k \leq J}} v_{tkh\tau_k j_k} x_{tkhd-i+hj_k} + \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_{ti} \\ \tau_k \geq d}} v_{tki\tau_k j_k} x_{tkidj_k} \leq V_{di}$$

for $d \geq i$, $i \in \widetilde{J} - \{J\}$ (3)

$$\sum_{h=1}^{d-1} \sum_{t=1}^{h} \sum_{\substack{k \in \widetilde{K}_{th+i-d} \\ \tau_k \geq h \\ h+i-d < j_k \leq J}} v_{tkh+i-d\tau_k j_k} x_{tkh+i-dhj_k} + \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_{ti} \\ \tau_k \geq d}} v_{tki\tau_k j_k} x_{tkidj_k} \leq V_{di}$$

for $d < i$, $i \in \widetilde{J} - \{J\}$ (4)

$$\sum_{h=1}^{i-1} \sum_{t=1}^{d-i+h} \sum_{\substack{k \in \widetilde{K}_{th} \\ \tau_k \geq d-i+h \\ h < j_k \leq J}} w_{tkh\tau_k j_k} x_{tkhd-i+hj_k} + \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_{ti} \\ \tau_k \geq d}} w_{tki\tau_k j_k} x_{tkidj_k} \leq W_{di}$$

for $d \geq i$, $i \in \widetilde{J} - \{J\}$ (5)

$$\sum_{h=1}^{d-1} \sum_{t=1}^{h} \sum_{\substack{k \in \widetilde{K}_{th+i-d} \\ \tau_k \geq h \\ h+i-d < j_k \leq J}} w_{tkh+i-d\tau_k j_k} x_{tkh+i-dhj_k} + \sum_{t=1}^{d} \sum_{\substack{k \in \widetilde{K}_{ti} \\ \tau_k \geq d}} w_{tki\tau_k j_k} x_{tkidj_k} \leq W_{di}$$

for $d < i$, $i \in \widetilde{J} - \{J\}$ (6)

$$\sum_{d=t}^{\tau_k} x_{tkidj_k} \leq 1 \quad \forall t \in \widetilde{T},\ i \in \widetilde{J} - \{J\},\ k \in \widetilde{K}_{ti} \qquad (7)$$

$$x_{tkidj_k} \in \{0,1\} \quad \forall t \in \widetilde{T}, i \in \widetilde{J} - \{J\},\ k \in \widetilde{K}_{ti},\ d \in \{t, t+1, \cdots, \tau_k\} \qquad (8)$$

The objective function (1) maximizes the total profit in the planning horizon T. The constraint (2) ensures that the demand for empty containers at port i is less than the number of all available empty containers at port i in period d. The constraint (3) and (4) ensures that the total volume of cargoes which will be carried by ship s_{di} is less than the total available volume capacity of ship s_{di}. The constraint (5) and (6) ensures that the total weight of cargoes which will be carried by ship s_{di} is less than the total available weight capacity of ship s_{di}. The constraint (7) imposes that each cargo may be carried in a certain period before its due date or be refused to carry in the planning horizon T. The constraint (8) assures that each cargo can be either accepted at its total quantity or be turned down.

We may note that the inequality sign in (7) is replaced by equality, if all received cargoes have to be carried in the planning horizon T. In this case, the MDMKP (M1) may have not feasible solution, as the total available capacity of shipment may be less than the amount of demands in the planning horizon T.

3. HEURISTIC ALGORITHM FOR MDMKP

The model formulated in the last section is a 0-1 or binary integer programming, which is an integer programming with all decision variables being 0-1 values. An important class of binary integer programming problems is the Knapsack Problem (KP). The Single Knapsack Problem (SKP) is a type of KP where there is only one knapsack to be filled and each item may be chosen at most once. If the items are chosen from disjoint classes and exactly one item from each class, it becomes the Multiple-Choice Knapsack Problem (MCKP). The Multiple Knapsack Problem (MKP) is also a type of KP where several knapsacks are to be packed simultaneously. A more general class of KP is the Multi-Dimensional Knapsack Problem (MDKP), also known as Multi-Constrained Knapsack Problem, where there are more than one capacitated resources (i.e., the resources are multi-dimensional for the knapsack). The Multi-Dimension Multiple Knapsack Problem (MDMKP) is the combination of MDKP and MKP. All Knapsack problems belong to the *NP-hard* family (see [3]). Therefore, it is impossible that polynomial time algorithms can be devised for them.

Various algorithms have been proposed for solving variants of KPs (see for example [7][9][10][11][12]). Generally, there are two classes of methods: one is a method for finding exact solutions and the other is heuristic. Finding exact solutions is NP hard. Using the branch and bound with linear programming technique, [4][5][8][13] presented exact algorithms for KP, MDKP, MCKP and MDKP respectively. A greedy approach has been proposed by [4][7][14] to find near optimal solutions of KPs.

The MDMKP (problem M1) formulated in this paper is different from all KPs studied in the literature and thus it is necessary to develop efficient algorithms. In this section, we present an efficient heuristic algorithm using the effective gradient for this problem.

Let $\overline{v}^{d}_{tkl\tau_k j_k} = v_{tkl\tau_k j_k} / V_{di}$, $\overline{w}^{d}_{tkl\tau_k j_k} = w_{tkl\tau_k j_k} / W_{di}$ in equations of (3) and (4). Then, we can rewrite the problem (M1) as follows:

(M2) Maximize $z = \sum_{t=1}^{T} \sum_{i=1}^{J-1} \sum_{k \in \widetilde{K}_{ti}} \sum_{d=t}^{\tau_k} v_{tki\tau_k j_k} r_{tkidj_k} x_{tkidj_k}$ (9)

subject to

$$\sum_{t=1}^{d}\sum_{\substack{k\in\tilde{K}_{ti}\\ \tau_k\geq d}} v_{tki\tau_k j_k} x_{tkidj_k} \leq E_{di} \quad \forall i\in\tilde{J}-\{J\},\ d\in\tilde{T} \tag{10}$$

$$\sum_{h=1}^{i-1}\sum_{t=1}^{d-i+h}\sum_{\substack{k\in\tilde{K}_{th}\\ \tau_k\geq d-i+h\\ h<j_k\leq J}} \bar{v}^{d}_{tkh\tau_k j_k} x_{tkhd-i+hj_k} + \sum_{t=1}^{d}\sum_{\substack{k\in\tilde{K}_{ti}\\ \tau_k\geq d}} \bar{v}^{d}_{tki\tau_k j_k} x_{tkidj_k} \leq 1,$$

for $d\geq i$, $i\in\tilde{J}-\{J\}$ (11)

$$\sum_{h=1}^{d-1}\sum_{t=1}^{h}\sum_{\substack{k\in\tilde{K}_{th+i-d}\\ \tau_k\geq h\\ h+i-d<j_k\leq J}} \bar{v}^{d}_{tkh+i-d\tau_k j_k} x_{tkh+i-dhj_k} + \sum_{t=1}^{d}\sum_{\substack{k\in\tilde{K}_{ti}\\ \tau_k\geq d}} \bar{v}^{d}_{tki\tau_k j_k} x_{tkidj_k} \leq 1,$$

for $d<i$, $i\in\tilde{J}-\{J\}$ (12)

$$\sum_{h=1}^{i-1}\sum_{t=1}^{d-i+h}\sum_{\substack{k\in\tilde{K}_{th}\\ \tau_k\geq d-i+h\\ h<j_k\leq J}} \bar{w}^{d}_{tkh\tau_k j_k} x_{tkhd-i+hj_k} + \sum_{t=1}^{d}\sum_{\substack{k\in\tilde{K}_{ti}\\ \tau_k\geq d}} \bar{w}^{d}_{tki\tau_k j_k} x_{tkidj_k} \leq 1,$$

for $d\geq i$, $i\in\tilde{J}-\{J\}$ (13)

$$\sum_{h=1}^{d-1}\sum_{t=1}^{h}\sum_{\substack{k\in\tilde{K}_{th+i-d}\\ \tau_k\geq h\\ h+i-d<j_k\leq J}} \bar{w}^{d}_{tkh+i-d\tau_k j_k} x_{tkh+i-dhj_k} + \sum_{t=1}^{d}\sum_{\substack{k\in\tilde{K}_{ti}\\ \tau_k\geq d}} \bar{w}^{d}_{tki\tau_k j_k} x_{tkidj_k} \leq 1,$$

for $d<i$, $i\in\tilde{J}-\{J\}$ (14)

$$\sum_{d=t}^{\tau_k} x_{tkidj_k} \leq 1 \quad \forall t\in\tilde{T},\ i\in\tilde{J}-\{J\},\ k\in\tilde{K}_{ti} \tag{15}$$

$x_{tkidj_k}\in\{0,1\}$ $\forall t\in\tilde{T}, i\in\tilde{J}-\{J\}$, $k\in\tilde{K}_{ti}$,
$d\in\{t,t+1,\cdots,\tau_k\}$ (16)

The heuristic algorithm for MDMKP is presented as follows.

Algorithm HASR:

Step 1: Initialization.

Step 1.1: Let $K_U \leftarrow \phi$, where K_U is the set of accepted items.

Step 1.2: Assign all items to $K_D = \tilde{K} - K_U$, where K_D is the set of items not in K_U.

Step 1.3: Let $A_U^{di} \leftarrow (0,0)$, where A_U^{di} is the total quantity vector of all cargoes loaded by ship which will call at port i in period d.

Step 1.4: Let the objective value be zero, i.e., $z \leftarrow 0$.

Step 1.5: Let

$$x_{tkidj_k} \leftarrow 0, \quad \forall t\in\tilde{T}, \quad k\in\tilde{K}, \quad d\in\{t,t+1,\cdots,\tau_k\}.$$

Step 2: Let

$$K_C \leftarrow \{k : k\in K_D, \exists d\leq\tau_k\ s.t. (\bar{v}^{d}_{tki\tau_k j_k}, \bar{w}^{d}_{tki\tau_k j_k}) \leq (1,1) - A_U^{di}, v_{tki\tau_k j_k} \leq E_{di} - \sum_{\substack{k'\in K_U\\ \tau_{k'}\geq d}} v_{tk'i\tau_k j_k}\},$$

where K_C is the set of candidate items.

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

Step 4: Let

$$\bar{K}_C \leftarrow \{(k,d) : k\in K_D, (\bar{v}^{d}_{tki\tau_k j_k}, \bar{w}^{d}_{tki\tau_k j_k}) \leq (1,1) - A_U^{di}, v_{tki\tau_k j_k} \leq E_{di} - \sum_{\substack{k'\in K_U\\ \tau_{k'}\geq d}} v_{tk'i\tau_k j_k}\}.$$

Compute effective gradients for the items in K_C as follows.

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

$$G_{kd} \leftarrow \frac{\sqrt{2}\, v_{tki\tau_k j_k} r_{tkidj_k}}{\bar{v}^{d}_{tki\tau_k j_k} + \bar{w}^{d}_{tki\tau_k j_k}}, \text{ for } (k,d)\in\bar{K}_C$$

Step 4.2: Otherwise, we set

$$G_{kd} \leftarrow \frac{v_{tki\tau_k j_k} r_{tkidj_k}\sqrt{(\sum_{k'\in K_U^d}\bar{v}^{d}_{tk'i\tau_{k'} j_{k'}})^2 + (\sum_{k'\in K_U^d}\bar{w}^{d}_{tk'i\tau_{k'} j_{k'}})^2}}{\bar{v}^{d}_{tki\tau_k j_k}\sum_{k'\in K_U^d}\bar{v}^{d}_{tk'i\tau_{k'} j_{k'}} + \bar{w}^{d}_{tki\tau_k j_k}\sum_{k'\in K_U^d}\bar{w}^{d}_{tk'i\tau_{k'} j_{k'}}},$$

for $(k,d)\in\bar{K}_C$, where $K_U^d = \{k' : k'\in K_U, x_{tk'idj_{k'}} = 1\}$.

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

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

Step 6: Accept k. Let $K_U \leftarrow K_U + \{k\}$,

$A_U^{di} \leftarrow A_U^{di} + (\bar{v}^{d}_{tki\tau_k j_k}, \bar{w}^{d}_{tki\tau_k j_k})$, $z \leftarrow z + v_{tki\tau_k j_k} r_{tkidj_k}$, $K_D \leftarrow K_D - \{k\}$, $x_{tkidj_k} \leftarrow 1$, and $G_{k'd'} \leftarrow 0$, for all $(k',d')\in\bar{K}_C$. Then, goto **Step 2**.

Similar with the algorithm in [14], HASR 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 HASR is some different from G_k introduced in [14], the value of G_{kd} depends on item k and its shipment period d. For a 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 MDMKP (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 HASR is usually a very well approximation of the optimal solution.

4. NUMERICAL EXPERIMENTS

In this section, we implement the heuristic algorithm HASR and compare its solution to the optimal solutions or LP optimal solutions (as the upper bound for optimal solutions). 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 J and port i at which cargo k is received is decided by:

$i = i_0$, if $\text{floor}(K/(J-1)) \times (i_0 - 1) \le k < \text{floor}(K/(J-1)) \times i_0$,

for $i_0 = 1, 2, \cdots, J-1$,

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

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 MDMKP instances, for which optimal solutions can be obtained by branch and bound algorithm. 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

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

The relative gap between heuristic solution and LP relaxation optimal solution g_L is computed as

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

Table 1 shows the results obtained for a set of test problems. Test problems 1 have 2 periods, 3 destination ports and 8 items (cargoes); test problem 2 has 3 periods, 6 destination ports and 14 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 1. Results for small test problems

T	*J*	*K*	Number of variables	Number of constraints	Instances tested	Average relative gap (%)	
						g_O	g_L
2	3	8	16	26	10	2.73	27.94
3	6	14	42	68	10	1.34	32.63
3	5	16	48	61	10	2.21	27.74
3	5	17	51	62	10	2.19	26.84
4	7	27	108	111	10	2.23	34.48
7	10	47	189	237	10	2.81	40.66

Table 2. Results for Medium Scale problems

T	*J*	*K*	*Number of variables*	*Number of constraints*	*Instances tested*	Average relative gap g_L (%)	*Average CPU time (sec)*	
							Heuristic	*LP*
5	9	185	925	320	10	8.74	1.15	6.53
3	6	370	1110	424	10	3.26	2.51	18.73
6	12	190	1140	406	10	18.17	1.41	8.51
6	10	245	1470	425	10	8.94	3.14	12.37
6	8	287	1722	431	10	10.44	2.61	22.67
7	15	525	3675	840	10	6.06	18.36	235.39

Table 2 shows the results obtained for a set of large scale problems. From our preliminary computation experiment, we believe that heuristic algorithm would be a very good candidate for solving MDMKP in time critical or real-time applications such as multi-period and multiple ships service route planning 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 and multiple ships service route planning problem as the MDMKP model, and presented effective heuristic algorithm 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 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 Center for E-Business of National University Singapore, and The Logistics Institute—Asia Pacific, a research partnership between the Georgia Institute of Technology (USA) and the National University of Singapore.

REFERENCES

[1] Cheung, R.K., C.-Y. Chen, "A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem", *Thansp. Sci.*, Vol.32, No.2, pp142-162, 1998.

[2] Crainic, T.G., M. Gendreau, P. Dejax, "Dynamic and Stochastic models for the allocation of empty containers", *Operations Research*, Vol. 41, No. 1, pp102-126, 1993.

[3] Garey, M.R., D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, freeman, 1979.

[4] Khan, S., K.F. Li, E.G. Manning, "The utility model for adaptive multimedia system", *In International Workshop on Multimedia Modeling*, pp111-126, 1997.

[5] Koleser, P., "A branch and bound algorithm for knapsack problem", *Management Science* Vol. 13, pp723-735, 1967.

[6] Lai, K.K., K. Lam, W. K. Chan, "Ship container logistics and allocation", *Journal of the Operational Research Society*, Vol. 46, No. 6, 687-697, 1995.

[7] Martello, S., P. Toth, "Algorithms for knapsack problems", *Annals of Discrete Mathematics*, Vol. 31, pp70-79, 1987.

[8] Nauss, R.M., "The Knapsack Problem with Multiple Choice Constraints", *European Journal of Operational Research*, Vol. 2, pp125-131, 1978.

[9] Pisinger, D., "Algorithms for knapsack problems", Ph.D. thesis, DIKU, University of Copenhagen, Report, 95/1 , 1995.

[10] Pisinger, D., "A minimal algorithm for the 0-1 Knapsack Problem", *Operations Research* Vol. 45, pp758-767, 1997.

[11] Pisinger, D., "Core problems in Knapsack Algorithms", *Operations Research*, Vol. 47, pp570-575, 1999.

[12] Pisinger, D., "An exact algorithm for large multiple knapsack problems", *European Journal of Operational Research*, Vol. 114, pp528-541, 1999.

[13] Shih, W., "A branch and bound method for the multiconstraint zero-one knapsack problem", *J. Opl. Res. Soc.*, Vol. 30, No. 4, pp369-378, 1979.

[14] Toyoda, Y., "A simplified algorithm for obtaining approximate solution to zero-one programming problems", *Management Science*, Vol. 21, pp1417-1427, 1975.