

# A Two-Phase Approach for a Trampler Ship Scheduling Problem

<sup>1</sup>Kazuhiro Kobayashi   <sup>2</sup>Mikio Kubo   <sup>1</sup>Toshiyuki Kano

<sup>1</sup>National Maritime Research Institute, Japan

<sup>2</sup>Tokyo University of Marine Science and Technology, Japan

May 10, 2008

SIAM Conference on Optimization 08

May 10-13, 2008

# Plan of this Presentation

- 1 Introduction
- 2 Algorithm and result
- 3 Conclusion

# 1 Introduction

## 2 Algorithm and result

## 3 Conclusion

# Introduction: Background and Motivation

Given a set of orders, **ship scheduling problem** is to find visiting sequences of ports to be visited by ships.

Two issues:

- reduction of workloads for planning
- finding efficient schedules (operational cost, fuel consumption)

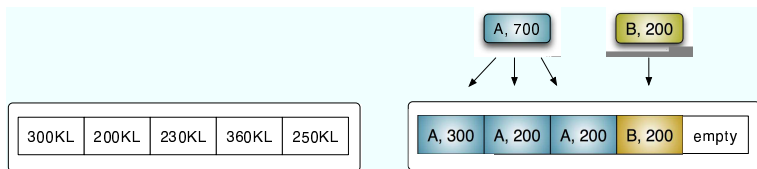
**Objective of this research:**

Development of an algorithm to generate efficient schedules with short computational time.

# Introduction: Background and Motivation

Similar to the Vehicle Routing Problem (VRP). However, some constraints are imposed **which rarely appear in the VRP**.

- multiple time windows
- allocation to the holds should be determined
- time dependent service time



# Introduction: Background and Motivation

Previous work:

1. Appelgren('69): column generation
2. Ronen ('83, '93): survey  
Christiansen et al. ('04): survey, set covering formulation
3. Christiansen et al. ('02): multiple time windows
4. Fagerholt('00): TSP with allocation, time window and precedence constraints

# Introduction: Background and Motivation

Formulation: set covering problem

Solution procedure: dynamic column generation

## Previous work

- optimal solution
- **subproblem**: shortest path problem with time windows, precedence constraint and capacity constraints  $\Rightarrow$  **NP-hard, long computational time**

## Our contribution (two-phase procedure)

- suboptimal solution
- **subproblem**: a shortest path problem on an acyclic network  $\Rightarrow$  **polynomial-time algorithm, short computational time**

1 Introduction

2 Algorithm and result

3 Conclusion

# Two-phase solution procedure

Our two-phase solution procedure:

Given data



Pairing problem ( 1st phase )

- Find the *pairings*

Routing problem ( 2nd phase )

- Find the visiting sequence of pairings



Schedule for each ship in the fleet

# Pairing problem (1st phase)

## Pairing

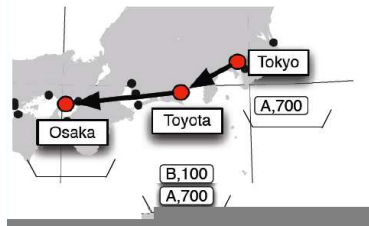
A set of orders delivered simultaneously in the same ship

Example: pairings  $\{1, 2\}$

id	orig.	dist.	load dt.	unload. dt.	prod.	quant.
1	Tokyo	Osaka	4	6	A	700
2	Toyota	Osaka	5	6	B	100

The sequence of ports to visits: Tokyo  $\Rightarrow$  Toyota  $\Rightarrow$  Osaka

Tokyo: load (A,700)  
 Toyota: load (B,100)  
 Osaka: unload (A,700)  
 unload (B,100)



# Pairing problem (1st phase)

## Pairing problem

Given a set of **orders**, find a set of efficient **pairings** such that efficient schedules can be obtained **in the 2nd phase**.

$$\begin{aligned}
 & \text{minimize} && \sum_{p \in P} C_p x_p \\
 & \text{subject to} && \sum_{p: k \in K_p} x_p = 1 \quad (\forall k \in K) \\
 & && x_p \in \{0, 1\} \quad (\forall p \in P)
 \end{aligned}$$

$C_p$  : cost of pairing  $p$

$x_p = 1$  ( pairing  $p$  is chosen ), 0 (o.w.)

The 1st constraint: each order is included in a certain pairing.

# Pairing problem (1st phase)

$$\begin{array}{ll}
 \text{minimize} & \sum_{p \in P} C_p x_p \\
 \text{subject to} & \sum_{p: k \in K_p} x_p = 1 \quad (\forall k \in K) \\
 & x_p \in \{0, 1\} \quad (\forall p \in P)
 \end{array}$$

Example:

Given orders:  $\{1, 2, 3, 4, 5, 6, 7\}$

$\Rightarrow$  Candidate pairings:

$P = \{\{1, 2\}, \{2, 3\}, \{6, 7\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\}$

Solution of the set partitioning formulation:

$x_2 = 1, x_3 = 1, x_4 = 1, x_7 = 1, x_8 = 1$

$\Rightarrow$  Efficient pairings:

$\{\{2, 3\}, \{6, 7\}, \{1\}, \{4\}, \{5\}\}$

# Routing problem (2nd phase)

## Route

A visiting sequence of pairings

A schedule for a ship is defined as a visiting sequence of pairings.

id	orig.	dist.	load dt.	unload. dt.	prod.	quant.
1	Osaka	Toyota	2	3	A	700
⋮	⋮	⋮	⋮	⋮	⋮	⋮
4	Toyota	Osaka	5	6	B	100
⋮	⋮	⋮	⋮	⋮	⋮	⋮
6	Osaka	Fukuoka	7	8	C	200
7	Osaka	Fukuoka	7	8	A	500
⋮	⋮	⋮	⋮	⋮	⋮	⋮

An example of a route:

$$\{1\} \rightarrow \{4\} \rightarrow \{6, 7\}$$

# Routing problem (2nd phase)

## Routing problem

Given a set of **pairings**, find a **visiting sequence of pairings** for each ship **such that** the cost is minimized and all pairings are processed.

$$\begin{aligned}
 &\text{minimize} && \sum_{v \in V} \sum_{r \in R} C_r x_{rv} + \sum_{p \in P} F_p y_p \\
 &\text{subject to} && \sum_{(r,v): p \in P_r} x_{rv} + y_p \geq 1 \quad (\forall p \in P) \\
 &&& \sum_{r \in R_v} x_{rv} \leq 1 \\
 &&& x_p \in \{0, 1\} \quad (\forall p \in P), y_p \in \{0, 1\} \quad (\forall p \in P)
 \end{aligned}$$

$C_r$ : cost of visiting sequence  $r$

$F_p$ : cost of delivering pairing  $p$  with spot chartered ship

$x_{rv} = 1$  (visiting sequence  $r$  is used for  $v$ ), 0 (o.w.)

$y_p = 1$  (pairing  $p$  is delivered by a spot chartered ship)

# Routing problem (2nd phase)

## Routing problem

Given a set of **pairings**, find a **visiting sequence of pairings** for each ship **such that** the cost is minimized and all pairings are processed.

$$\begin{aligned}
 &\text{minimize} && \sum_{v \in V} \sum_{r \in R} C_r x_{rv} + \sum_{p \in P} F_p y_p \\
 &\text{subject to} && \sum_{(r,v): p \in P_r} x_{rv} + y_p \geq 1 \quad (\forall p \in P) \\
 &&& \sum_{r \in R_v} x_{rv} \leq 1 \\
 &&& x_p \in \{0, 1\} \quad (\forall p \in P), y_p \in \{0, 1\} \quad (\forall p \in P)
 \end{aligned}$$

The 1st constraint: each pairing is delivered at least once.

The 2nd constraint: at most one route is assigned to ship  $v$ .

# Routing problem (2nd phase)

## Routing problem

Given a set of **pairings**, find a **visiting sequence of pairings** for each ship **such that** the cost is minimized and all pairings are processed.

$$\begin{aligned}
 &\text{minimize} && \sum_{v \in V} \sum_{r \in R} C_r x_{rv} + \sum_{p \in P} F_p y_p \\
 &\text{subject to} && \sum_{(r,v): p \in P_r} x_{rv} + y_p \geq 1 \quad (\forall p \in P) \\
 &&& \sum_{r \in R_v} x_{rv} \leq 1 \\
 &&& x_p \in \{0, 1\} \quad (\forall p \in P), y_p \in \{0, 1\} \quad (\forall p \in P)
 \end{aligned}$$

Example:

Given pairings:  $\{\{2, 3\}, \{6, 7\}, \{1\}, \{4\}, \{5\}\} \Rightarrow$

Routes: ship A:  $\{2, 3\} \rightarrow \{1\} \rightarrow \{5\}$ , ship B:  $\{6, 7\} \rightarrow \{4\}$

# Routing problem

Number of candidate routes for ship  $v$ , i.e.,  $|R_v|$  is huge  $\Rightarrow$  dynamic column generation

## Subproblem in the column generation

- Shortest path problem on an acyclic network  $\Rightarrow$  **polynomial-time algorithm, low computational cost**

e.g. Bellman-Ford algorithm

NOTICE:

### [Previous work:](#)

- Subproblem: shortest path problem with time windows, precedence constraints and capacity constraints  $\Rightarrow$  **NP-hard, high computational cost**

# Numerical results

## Environment:

Processor:	Intel Core 2 6700 2.66GHz
Memory:	2GB RAM,
OS:	Windows XP
Modelling language:	Xpress Mosel Version 1.6.3
LP and IP Solver:	Xpress Optimizer Version 17.10.04

## Data:

- planning horizon: 36 days
- number of orders: 128
- visiting ports: 25 ports,
- number of ships in the fleet: 7 ships
- number of spot chartered ships: 10 ships
- 38973 total mile

# Numerical results: data 1

Schedule used (generated by hand at the shipping company):

number of spot chartered ships	total miles
10	38973

Schedule obtained by our two-phase approach:

obj.	spot	total miles	cpu time (seconds)	decrease (%) ( miles)
min. spot	9	38708	547	0.7 %
min. dist	10	36105	558	7.9 %
s.t. spot $\leq$ 10				
min. dist	11	35846	539	8.0 %

## Numerical results: data 2

Schedule used (generated by hand at the shipping company):

number of spot chartered ships	total miles
9	27086

Schedule obtained by our two-phase approach:

obj.	spot	total miles	cpu time (seconds)	decrease (%) ( miles)
min. spot	6	27000	240	0.3 %
min. dist	9	25617	249	5.4 %
s.t. spot $\leq$ 9				
min. dist	10	25325	238	6.5 %

1 Introduction

2 Algorithm and result

**3 Conclusion**

# Conclusion

## Our contribution:

- Two-phase approach:
  - Pairing problem ( 1st problem )
  - Routing problem ( 2nd problem )
- Find suboptimal solution with short computational time.
  - Column generation
  - Subproblem: shortest path problem on an acyclic network.
- Efficient schedule can be obtained in a reasonable amount of computational time ( - 10 minutes ).

## Future work

- Numerical experiments: comparison with previous works
- Use information obtained in the 2nd-phase routing problem to choose efficient pairings in the 1st-phase pairing problem