Sustainability considerations in container pre-marshalling problem

EURO2021, Athens (hybrid), University of West Attica, 11-14 July 2021

Cihan Bütün
Afshin Mansourir
Ran Wang
Brunel Business School
Brunel University London
Agenda

- Research motivations
- Problem definition & literature review
- Model formulation
- Solution heuristic development
- Computational experiments
- Conclusions
Research motivations

Reduce average vessel turnaround time

Smooth and fast cargo transfer between container yard and berth

Prevent non-productive container movements in the container yard

Pre-marshalling (or housekeeping) to prevent “reshuffles”

A major source of inefficiency (Choe et al., 2011)

Also called “rehandle”, “restow”, or “repositioning”

(Source: https://www.termavi.com/en/multimedia-2/)
Research motivations

What is a reshuffle?

Numbers inside the containers indicate their priority orders.

To reach container 1, 2 and 3 must be reshuffled. Both movements are non-productive.

How does pre-marshalling work?

Reorder containers during off-peak hours to eliminate reshuffles during loading/retrieval in a yard bay.

Why is pre-marshalling needed?

Scarcity of empty yard slots

Random container arrivals

Imprecise container information

(Kang et al., 2006; Lee and Chao, 2009; Choe et al., 2011).
Research motivations

Environmental sustainability in maritime shipping and port operations

**Common practices**
- Speed reduction
- Re-routing
- Berth scheduling

*(Mansouri et al., 2015)*

**Main research areas**
- Emission reduction
- Eco-efficient operations
- Climate change and regulation
- Ship design and mobility
- Carbon footprint case studies

*(Davarzani et al., 2016)*

**Regulatory developments**
- MARPOL Convention (IMO, 1973-1997)
- Ecoports (ESPO, 1997, 2011)
Problem definition

Common terminology

**Blocks (or streets) are collections of container bays**

**Bay**: A rectangular configuration of container slots divided column-wise into stacks and row-wise into tiers.

Container terminal layout

Each column is a bay

4 x 4 = 16 container slots
Problem definition

A pre-marshalling example

An “unordered” bay configuration: 2 blocks 1, and 4 blocks 3

An “ordered” bay configuration: None of the containers are blocked by others.

Initial bay configuration

Move container 4 from stack 2 to stack 3

Move container 2 from stack 4 to stack 1
Literature review

Container pre-marshalling problem: Minimize the number of container movements required to obtain an ordered bay

- Lee and Hsu, 2007: Multi-commodity flow formulation
- Lee and Chao, 2009: Neighbourhood search + integer programming
- Bortfeldt and Forster, 2012: Tree search heuristic
- Expósito-Izquierdo et al., 2012: A* heuristic, instance generator
- Rendl and Prandstetter, 2013: Constraint programming
- Jovanovic et al., 2017: Greedy heuristic
- Tierney et al., 2017: A* & IDA algorithms
- Tanaka and Tierney, 2018: Iterative deepening branch and bound
- Tierney et al., 2019: Branch and bound algorithm
- Parreño-Torres et al., 2019: New integer programming formulations
- Hottung et al., 2020: Deep learning heuristic tree search
Literature review

Non-conventional problem definitions appear recently:

- Parreño-Torres et al., 2020: Minimize crane times in pre-marshalling problems

- Boge et al., 2020: Robust optimization for premarshalling with uncertain priority classes

- Zweers et al., 2020: Stochastic container relocation problem with constrained pre-processing
Literature review

Number of publications

<table>
<thead>
<tr>
<th>Years</th>
<th>2007-2011</th>
<th>2012-2016</th>
<th>2017-2021</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publications</td>
<td>4</td>
<td>17</td>
<td>17</td>
</tr>
</tbody>
</table>
Problem definition

Container pre-marshalling problem with sustainability considerations

For a given initial bay configuration and the (retrieval) priority order of the containers, determine the sequence of the container movements so as to minimize the total energy consumption of the yard cranes and achieve an ordered bay.

(Source: https://www.termavi.com/en/multimedia-2/)
Problem definition

Assumptions:

• All containers are assumed to be of the same size and dimensions (1 TEU).

• The priority order of the containers for retrieval is known in advance.

• Only one crane operates on the bay, and only a single container can be moved at a time.

• The number of containers in the bay does not change during the pre-marshalling operation.

• All stacks and tiers in the bay are identical.

• Each slot can contain at most one container.

• The unit idle and operational energy consumption of the cranes are known and constant for each container.

(Lee and Hsu, 2007; Lee and Chao, 2009; Wang et al., 2015; 2017; Tierney et al., 2017; Tanaka and Tierney, 2018)
Model formulation

- The proposed mixed integer programming model is formulated by partially adopting the classical pre-marshalling problem formulation proposed by Parreño-Torres et al. (2019).
- The crane energy consumption of a single container movement is calculated by considering six crane manoeuvres:

1. Movement of the crane’s trolley from its previous position to the source stack $s$
2. Lowering of the spreader to the topmost container of stack $s$
3. Hoisting of the topmost container with the spreader
4. Movement of the crane’s trolley from $s$ to the target stack $k$
5. Lowering of the spreader to place the container on top of stack $k$
6. Hoisting of the spreader back to the trolley
## Model formulation

<table>
<thead>
<tr>
<th>Set</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S$</td>
<td>Set of stacks, $s, k, l = {1, ..., S}$</td>
</tr>
<tr>
<td>$\mathcal{H}$</td>
<td>Set of tiers, $h, e = {1, ..., H}$</td>
</tr>
<tr>
<td>$\mathcal{T}$</td>
<td>Set of time segments, $t = {1, ..., T}$</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>Set of container retrieval priorities, $p, r = {1, ..., P}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_p$</td>
<td>The number of containers in priority group $p$,</td>
</tr>
<tr>
<td>$t_{fl_{sk}}$</td>
<td>The energy consumed for moving the trolley loaded with a container from stack $s$ to $k$</td>
</tr>
<tr>
<td>$t_{fu_{ls}}$</td>
<td>The energy consumed for moving the trolley unloaded from stack $l$ to $s$</td>
</tr>
<tr>
<td>$t_{fi_h}$</td>
<td>The energy consumed by keeping the trolley idle during the spreader’s movement between trolley level and tier $h$</td>
</tr>
<tr>
<td>$s_{fl_h}$</td>
<td>The energy consumed for lowering the spreader empty to tier $h$ and hoisting a container to the trolley level</td>
</tr>
<tr>
<td>$s_{fu_h}$</td>
<td>The energy consumed for lowering the spreader loaded with a container to tier $h$ and hoisting it empty to the trolley level</td>
</tr>
<tr>
<td>$s_{fi_{sk}}$</td>
<td>The energy consumed by keeping the spreader idle during the trolley’s movement from stack $s$ to $k$</td>
</tr>
</tbody>
</table>
Model formulation

Decision variables

\( Y_{shkept} \)  
1 if a container of priority \( p \) is moved from stack \( s \) tier \( h \) to stack \( k \) tier \( e \) between the time segments \( t \) and \( t + 1 \), 0 otherwise  
\( \forall s \in S, h \in \mathcal{H}, k \in S \setminus \{s\}, e \in \mathcal{H}, p \in \mathcal{P}, t \in \mathcal{T} \setminus \{T\} \)

\( X_{shpt} \)  
1 if a container with priority \( p \) occupies stack \( s \) tier \( h \) at the beginning of time segment \( t \), 0 otherwise  
\( \forall s \in S, h \in \mathcal{H}, p \in \mathcal{P}, t \in \mathcal{T} \)

\( C_{lst} \)  
1 if the trolley is moved from stack \( l \) to \( s \) for moving a container from stack \( s \) between the time segments \( t \) and \( t + 1 \), 0 otherwise  
\( \forall l \in S, s \in S \setminus \{l\}, t \in \mathcal{T} \setminus \{1, T\} \)
Model formulation

Minimise

\[
\sum_{l=1}^{S} \sum_{s=1}^{S} \sum_{t=2}^{T-1} (tf u_{ls} * c_{lst}) +
\]

\[
\sum_{s=1}^{S} \sum_{h=1}^{H} \sum_{k=1}^{S} \sum_{e=1}^{H} \sum_{p=1}^{P} \sum_{t=1}^{T-1} (tf l_{sk} + tf i_h + tf e + sf i_{sk} + sf l_h + sf u_e) * Y_{shkept}
\]

such that

\[
\sum_{s=1}^{S} \sum_{h=1}^{H} X_{shpt} = N_p \quad \forall p \in P, t \in T
\]

\[
\sum_{s=1}^{S} \sum_{h=1}^{H} \sum_{k=1}^{S} \sum_{e=1}^{H} \sum_{p=1}^{P} Y_{shkept} \leq 1 \quad \forall t \in T \setminus \{T\}
\]

\[
\sum_{s=1}^{S} \sum_{h=1}^{H} \sum_{k=1}^{S} \sum_{e=1}^{H} \sum_{p=1}^{P} Y_{shkept+1} \leq \sum_{s=1}^{S} \sum_{h=1}^{H} \sum_{k=1}^{S} \sum_{e=1}^{H} \sum_{p=1}^{P} Y_{shkept} \quad \forall t \in T \setminus \{T-1, T\}
\]

(1) Minimize the total energy consumption of the crane to perform all movements

(2) Assign a slot position to all containers at all time segments

(3) 1 movement per time segment

(4) Perform the movements at earliest time segments
Model formulation

\[ \sum_{p=1}^{P} X_{s1pt} \leq 1 \quad \forall s \in S, t \in T \setminus \{1\} \]  
\[ (5) \]

(5)-(6) Each tier at most 1 container

\[ \sum_{p=1}^{P} X_{sh+1pt} \leq \sum_{p=1}^{P} X_{shpt} \quad \forall s \in S, h \in \mathcal{H} \setminus \{H\}, t \in T \setminus \{1,T\} \]  
\[ (6) \]

(6) Only move the topmost container at each stack

\[ X_{shpt} + \sum_{r=1}^{P} X_{sh+1rt} \leq 1 + X_{shpt+1} \quad \forall s \in S, h \in \mathcal{H} \setminus \{H\}, p \in \mathcal{P}, t \in T \setminus \{T\} \]  
\[ (7) \]

(7) Only move the topmost container at each stack

\[ \sum_{s=1}^{S} \sum_{h=1}^{H} \sum_{e=1}^{H} Y_{shkept} + \sum_{s=1}^{S} \sum_{h=1}^{H} \sum_{e=1}^{H} Y_{keshpt+1} \leq 1 \quad \forall k \in S, p \in \mathcal{P}, t \in T \]  
\[ (8) \]

(8) Prevent transitive moves (Tierney et al., 2017)

\[ \sum_{k=1}^{S} \sum_{e=1}^{H} Y_{shkept} + \sum_{k=1}^{S} \sum_{e=1}^{H} Y_{keshpt+1} \leq 1 \quad \forall s \in S, h \in \mathcal{H}, p \in \mathcal{P}, t \in T \setminus \{T-1,T\} \]  
\[ (9) \]

(9) Prevent symmetrical moves (Tanaka and Tierney, 2018)

\[ X_{shpt} + \sum_{k=1}^{S} \sum_{e=1}^{H} Y_{keshpt} = X_{shpt+1} + \sum_{k=1}^{S} \sum_{e=1}^{H} Y_{shkept} \quad \forall s \in S, h \in \mathcal{H}, p \in \mathcal{P}, t \in T \setminus \{T\} \]  
\[ (10) \]

(10) Slot occupation constraint
Model formulation

\[
\sum_{r=p}^{P} X_{s_h+1rT} \leq \sum_{r=p}^{P} X_{shrT} \quad \forall \ s \in S, h \in H \setminus \{H\}, p \in P
\]

\(\text{(11) The bay is ordered at } T\)

\[
C_{lst} \leq \sum_{h=1}^{H} \sum_{e=1}^{S} \sum_{p=1}^{H} \sum_{k \neq e}^{P} Y_{shkpt} \quad \forall \ l \in S, s \in S, t \in T \setminus \{1, T\}
\]

\(\text{(12)-(14) } C_{lst} \text{ is 1 if and only if the target stack of the movement during } t-1 \text{ was } l \text{ and the source stack of the movement during } t \text{ is } s\)

\[
C_{lst} \leq \sum_{k=1}^{S} \sum_{h=1}^{H} \sum_{h=1}^{P} \sum_{p=1}^{P} Y_{kelhpt-1} \quad \forall \ l \in S, s \in S, t \in T \setminus \{1, T\}
\]

\(\text{(13)}
\]

\[
\sum_{k \neq l}^{S} \sum_{h=1}^{H} \sum_{h=1}^{P} \sum_{k \neq s}^{P} Y_{kelhpt-1} + \sum_{h=1}^{H} \sum_{e=1}^{S} \sum_{k \neq s}^{P} Y_{shkpt} \leq C_{lst} + 1 \quad \forall \ l \in S, s \in S, t \in T \setminus \{1, T\}
\]

\(\text{(14)}
\]

\[
Y_{shkpt}, C_{lst} \in \mathbb{R}^+, X_{shpt} \in \{0,1\} \quad \forall \ s, k, l, e, h, p, t
\]

\(\text{(15) Integrality and nonnegativity constraints}\)
Solution heuristic development

- An A* algorithm-based solution heuristic is being developed.
- A* algorithm is a well-known pathfinding algorithm and was proposed to solve the conventional pre-marshalling problem (Exposito-Izquierdo et al., 2012; Tierney et al., 2017)
- In a graph of nodes, iteratively construct the shortest path between a starting location and a destination using a function on the sum of exact and estimated costs (distances):

\[ f(n) = g(n) + h(n) \]

- The exact cost from start to node \( n \)
- The estimated cost from node \( n \) to destination
Solution heuristic development

- In the proposed solution method, the energy consumption function is incorporated into the A* algorithm, and path selection is made based on the realized and estimated energy consumptions.

- Assume initial bay configuration as the starting location and any ordered bay configuration as the destination.

- Each intermediate bay configuration refers to the nodes visited along the path between the starting location and destination.

- The algorithm favours the paths which have the lowest total energy consumption.

- **(Under development):** To intensify the search in promising paths, hybridize the A* algorithm with an evolutionary heuristic.
  - Select “m” elite paths with minimum total energy consumption
  - Generate random move sequences from each elite path.
  - Apply evolutionary mechanisms to optimize the move sequences
  - Stop if a sequence yields an ordered bay.
## Computational experiments

### Experiment instances:

<table>
<thead>
<tr>
<th></th>
<th>Stacks</th>
<th>Tiers</th>
<th>Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee and Hsu (2007)</td>
<td>6</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>Expósito-Izquierdo et al. (2012)</td>
<td>4</td>
<td>4</td>
<td>8 /12 / 16</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4</td>
<td>7 / 14 /28</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>4</td>
<td>10 / 20 / 40</td>
</tr>
<tr>
<td>This paper</td>
<td>6</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Use case Spanish terminal</td>
<td>6</td>
<td>4</td>
<td>Various</td>
</tr>
</tbody>
</table>
Computational experiments

Preliminary observations

Container movements and energy consumption do not conflict with each other according to the observations.

4-6% energy reduction with the proposed objective function.

The solution heuristic can find good solutions for instances with low bay occupation ratio (=containers/total bay capacity) in a reasonable computation time.

The heuristic slows down with difficult instances (i.e. high bay occupation ratio and/or a bay with more stacks/tiers) due to the memory issues.
Computational experiments

PortForward project funded under EU Horizon 2020 program (2018-2021)

Green Yard Scheduler: A DSS to optimize container yard operations

Case Study

Design and develop a Housekeeping optimization module

Use case: A Spanish container terminal
Conclusions

• Container pre-marshalling problem with sustainability considerations is introduced.

• An integer programming model formulation is presented and a solution heuristic is discussed.

• Preliminary experiment results are discussed.

Future research:

• Complete the hybridization of A* algorithm with evolutionary heuristic to find good solution for difficult pre-marshalling problem instances.

• Expand the research to intra-bay pre-marshalling movements.

Acknowledgements: This project receives funding in the European Commission’s Horizon 2020 Research Program under Grant Agreement Number 769267
References


References


Thank you for your attention!
Questions, comments?

Contact Details

Dr Cihan Bütün

Address: Brunel University London,
Uxbridge, Middlesex UB8 3PH,
United Kingdom

Department: Brunel Business School
Email: cihan.butun@brunel.ac.uk

Dr Cihan Butun | Introduction | Brunel University London

https://www.linkedin.com/in/cihanbutun/