Solving the yard crane scheduling problem with Input/Output assignment

Author: Hongtao Wang

Advisor: Rubén Ruiz; Eva Vallada; María Fulgencia Villa

Ph.D Program in Statistic and Operational Research

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEIOAC VALENCIA — DEPARTAMENTO DE ESTADÍSTICA E INV. OPERAT. APLICADAS Y CALIDAD

1-Abstract

This poster introduces a combined metaheuristic, Iterated Greedy (IG) method for the single yard crane scheduling problem with input/output points (IO) assignment.

- In the scheduling problem we consider the occupancy of IOs with the release time or due date of the container among which the caused tardiness/earliness and congestion are penalized.
- For the complexities of the problem, we introduce a series of constructive and heuristic methods for both container sequencing and IO assignment. The methods are assembled as well as calibrated IG and GRASP structure coupled with local search and IO heuristics.

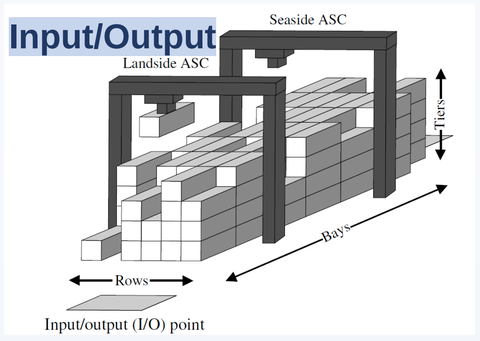
2-Background

1. we consider a European terminal configuration which is also popular for the full automatic terminals.
2. In the terminal, all the containers are delivered between yard and two sides by one yard crane over the block.
3. Every time the crane can transport only one container without break or temporary stop in our assumptions and it must be delivered by one IO in each side.
4. We presume that most of the containers are already placed on the top of each stack, and for containers located on the middle which need to be reshuffled, the reshuffling times are considered as the constants.

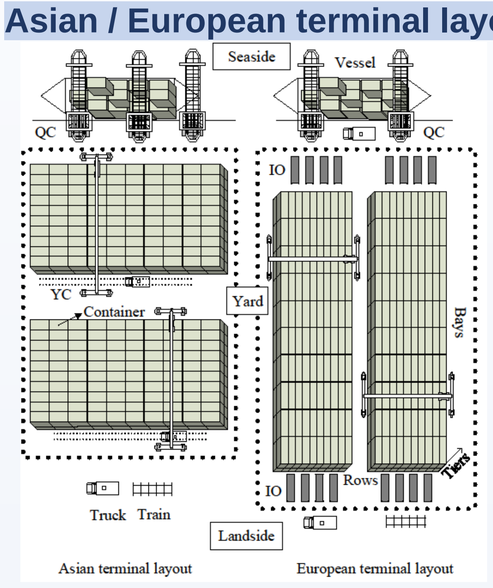
Container

Yard crane

Input/Output



Asian / European terminal layout



3- Mathematical Model

- The problem is actually a **yard crane scheduling problem** with **IO assignment** in which a list of jobs are sequenced with assigned IOs to minimize the total **weighted tardiness and congestion**.
- In the terminal, due to the arriving time of the container, AGVs or trucks, the containers are delivered with the release times. Ideally, the job are expected to be finished on the time, beyond which the container may be late for the vessels or trucks resulting in **tardiness**.
- The crane may not able to load the containers on time, making the congestion when all the IOs are occupied by the containers waiting to be loaded. In the operational level, the tardiness and congestion should to be penalized. Therefore, minimizing the weighted tardiness and **congestion** is the objective of the terminal

$$\min \sum_{c\in C} w_c^R(F_c - t_c) + \sum_{c\in C\setminus C_3} w_c^C(SO_c - t_c) + \sum_{c\in C_3} w_c^E E_c$$

$$\text{s.t.} \sum_{k\in K}\sum_{j\in O_c} X_{ckj} = 1 \qquad \forall c \in C$$

$$\sum_{c\in C}\sum_{j\in O_c} X_{ckj} = 1 \qquad \forall k \in K$$

$$SO_c \le SL_c \qquad \forall c \in C_1 \cup C_2$$

$$F_c = EL_c \qquad \forall c \in C_1 \cup C_2$$

$$SO_c = EL_c \qquad \forall c \in C_3 \cup C_4$$

$$F_c \ge SO_c \qquad \forall c \in C_3 \cup C_4$$

$$SO_c \ge t_c \qquad \forall c \in C \setminus C_3$$

$$E_c = t_c - SO_c + T_c \qquad \forall c \in C_3$$

$$F_c \ge t_c \qquad \forall c \in C$$

$$Loading - Constraints(12) - (19).$$

1. Obj: Minimize the lateness, congestion.
2. TSP- like (Traveling salesman problem) model
3. Variable time definition constraints
4. Loading constraints of variables

4-Iterated Greedy for GRASP

Step 1：start with a Initial sequence
Step 2: Solution destruction and reconstruction
Step 3: Local search to improving the solution quality
Step 4: IO heuristic for better IO assignment
Step 5: Solution selection by the acceptance criteria

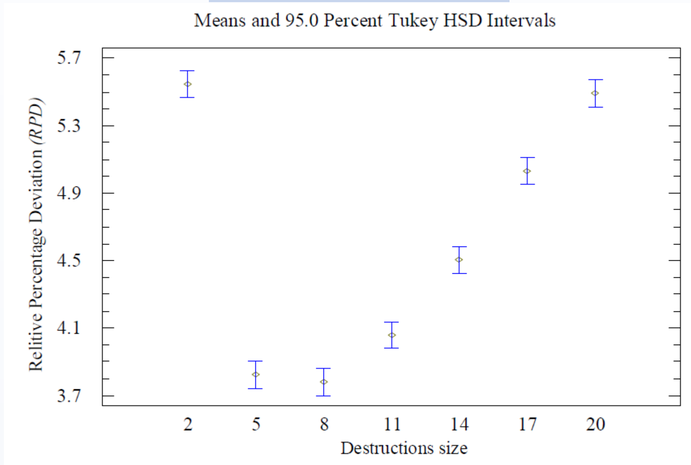
Main operators of the IG

Algorithm : Iterated Greedy (IG)

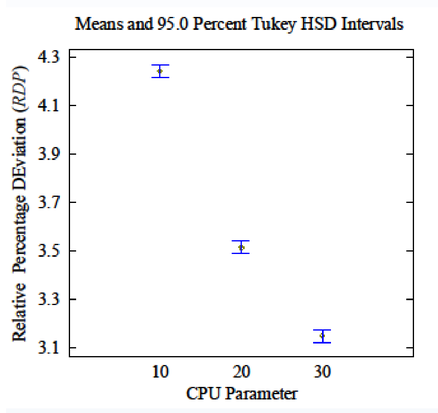
1: π_0 := Generated Initial TRP/MTRP/NCR Solution ;
2: π := Local Search(π_0);
3: **while** *termination criteria is not satisfied* **do**
4: *improvement*: =*True*;
5: π_1 := Destruction(π);
6: π_2 := Reconstruction(π_1);
7: π_3 := Local Search$(\pi_2)_{\pi_2<\pi}$;
8: π_4 := IO Heuristic(π_3);
9: π := Acceptance Criterion(π_4);
10: **end while**

5-Calibration of the parameters

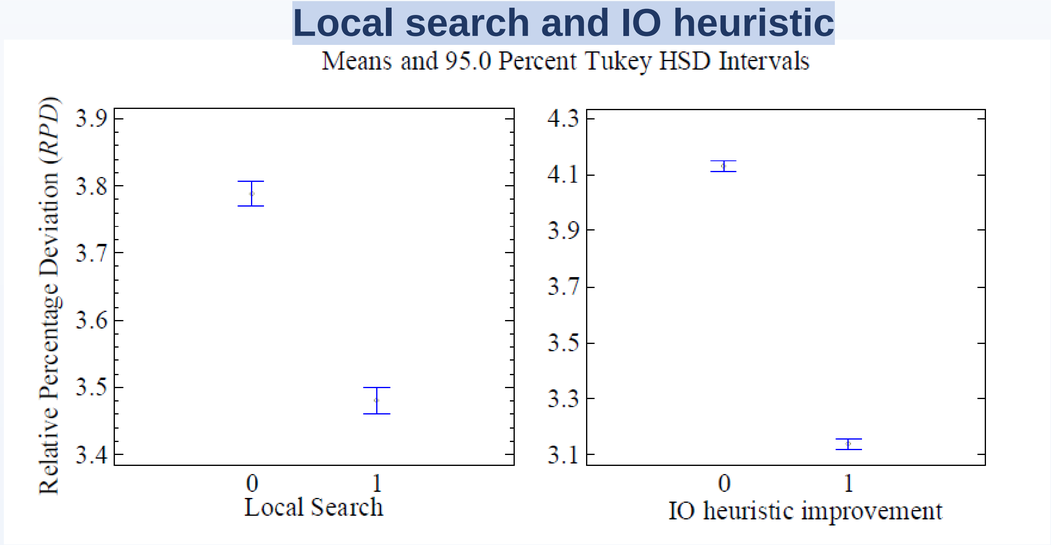
Destruction size



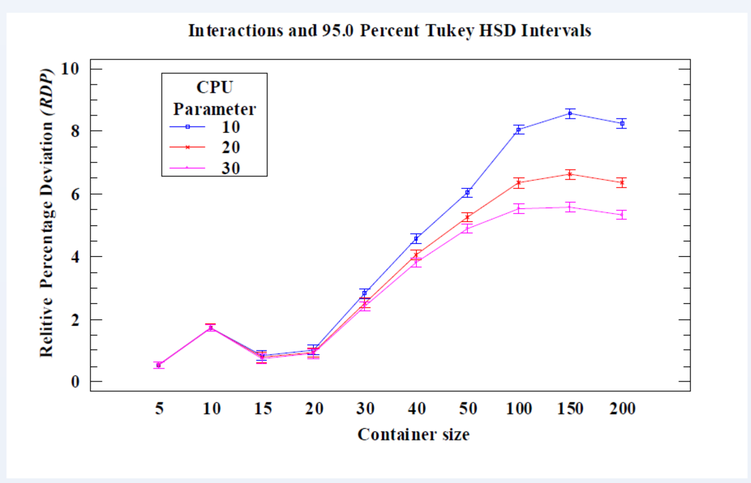
Cpu-time Parameter



Local search and IO heuristic



6-Performance of the method



Brief summary:

1. Relative percentage deviation(**RDP**)= $\frac{Current_s - Best_s}{Best_s} * 100\%$
2. Operators are **significant** in **ANOVA**
3. More CPU time, **lower RPD**
4. All RDPs are very **small (<10%)**

Appendix-Extension application: Valencia Port