MATHEMATICAL MODEL FOR RESHUFFLING OF CONTAINERS

B	the set of bay-locations
Q	the set of bay-locations capacities (i.e. 8, 12, 16, 20)
F	the set of the possible class configurations
W	the set of weight configurations
P	the set of weight limits
C	the set of containers
D	the set of ports of destination
H	the set of types/heights (Box, High Cube)
S	the set of sizes (20, 40 feet)
n_q	the number of 20'bay-locations having capacity q, $\forall q \in Q$
d_i	destination of container i, $\forall i \in C$
s_i	size of container i, $\forall i \in C$
h_i	height of container i, $\forall i \in C$
w_i	weight of container i, $\forall i \in C$
u_p	weight upper bound of weight limit $p, \forall p \in P$
l_p	weight lower bound of weight limit $p, \forall p \in P$
$\delta_{fw} \forall f \in F, \forall w \in W, \delta_{fw} = 1$	if the weight configuration w belong to class configuration f, 0 otherwise
$\gamma_{wp} \forall w \in W, \forall p \in P, \gamma_{wp} = 1$	if the weight limits p belong to configuration w, 0 otherwise
α	weight used in the objective function for penalising the empty slots in the bay-locations.

Let us introduce the following decision variables:

$x_{ij} \in \{0, 1\}, \forall i \in C, \forall j \in B$	$x_{ij} = 1$ if container i is stored in bay-locations j
$y_{jdhsq} \in \{0, 1\}, \forall j \in B, \forall d \in D, \forall h \in H, \forall s \in S, \forall q \in Q$	$y_{jdhsq} = 1$ if in bay-locations j, with capacity q, are stored containers for destination d, with height h and size s
$c_{pj} \in \{0, 1\}, \forall p \in P, \forall j \in B$	$c_{pj} = 1$ if weight limits p are assigned to bay-locations j
$a_f \in \{0, 1\}, \forall f \in F$	$a_f = 1$ if class configuration f is chosen for the yard storage
$b_w \in \{0, 1\}, \forall w \in W$	$b_w = 1$ if weight configuration w is chosen

$z_j \geq 0, \forall j \in B$ — number of empty slots in bay-locations j.

The resulting model is the following:

$$Min \sum_{j\in B}\sum_{d\in D}\sum_{h\in H}\sum_{s\in S}\sum_{q\in Q} y_{jdhsq} + \alpha \sum_{j\in B} z_j \qquad (1)$$

Subject to:

$$\sum_{j\in B} x_{ij} = 1 \quad \forall i \in C \qquad (2)$$

$$\sum_{d\in D}\sum_{h\in H}\sum_{s\in S}\sum_{q\in Q} y_{jdhsq} \leq 1 \quad \forall j \in B \qquad (3)$$

$$\sum_{i\in C} x_{ij} \leq \sum_{d\in D}\sum_{h\in H}\sum_{s\in S}\sum_{q\in Q} q\, y_{jdhsq} \quad \forall j \in B \qquad (4)$$

$$\sum_{j\in B}\sum_{d\in D}\sum_{h\in H}\sum_{s\in S:s_i=20} y_{jdhsq} + 2\sum_{j\in B}\sum_{d\in D}\sum_{h\in H}\sum_{s\in S:s_i=40} y_{jdhsq} \leq n_q \quad \forall q \in Q \qquad (5)$$

$$\sum_{f\in F} a_f = 1 \qquad (6)$$

$$\sum_{w\in W} b_w = 1; \qquad (7)$$

$$\sum_{w\in W} \delta_{wf} b_w \leq a_f \quad \forall f \in F \qquad (8)$$

$$\sum_{p\in P} c_{pj} \leq 1 \quad \forall j \in B \qquad (9)$$

$$\sum_{p\in P} \gamma_{pw} c_{pj} \leq b_w \quad \forall j \in B, \forall w \in W \qquad (10)$$

$$w_i x_{ij} \leq \sum_{p\in P} u_p c_{pj} \quad \forall i \in C, \forall j \in B \qquad (11)$$

$$w_i x_{ij} \geq \sum_{o\in P} l_p c_{pj} - M(1 - x_{ij}) \quad \forall i \in C, \forall j \in B \qquad (12)$$

$$\sum_{d\in D}\sum_{h\in H}\sum_{s\in S}\sum_{q\in Q} q\, y_{jdhsq} - \sum_{i\in C} x_{ij} \leq z_j \quad \forall j \in B \qquad (13)$$

THE HEURISTIC MODEL BASED ON TIME-INDEXING (OUT_TIME)

Indices

i, q=Job index, i, q=1, 2,..., n

j=Machine index, j=1, 2,..., m

k=Tool type index, k=1, 2,..., t

h_k=Replicate number of tool type k, h_k=1, 2,..., r_k

u=Time interval, u=1, 2,..., T

Parameters

p_{ij}=Processing time of job i on machine j

r_k=Number of tools of type k

$l(i)$=Set of tools required to process job i

Decision variables

$C_{\max}$=Production makespan

C_i=Production completion time of job i

$$X_{iju} = \begin{cases} 1 & \text{if job } i \text{ is scheduled to start processing on machine } j \text{ at the beginning of time interval } u \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{ih_k} = \begin{cases} 1 & \text{if replicate } h \text{ of tool } k \text{ is used to process job } i \\ 0 & \text{otherwise} \end{cases}$$

The mathematical model TIM is given below:

$$\text{Min } C_{\max} \tag{1}$$

subject to

$$C_{\max} \geq C_i \qquad \forall i \tag{2}$$

$$C_i = \sum_{j=1}^{m} \sum_{u=0}^{T} \left[\left(u + p_{ij} \right) X_{iju} \right] \qquad \forall i \tag{3}$$

$$\sum_{j=1}^{m} \sum_{u=0}^{T} X_{iju} = 1 \qquad \forall i \tag{4}$$

$$X_{iju} = 0 \qquad \forall i, j, u = T - p_{ij} + 1, \ldots, T \tag{5}$$

$$\sum_{i=1}^{n} \sum_{\mu=u-p_{ij}+1}^{u} X_{ij\mu} \leq 1 \qquad \forall j, u \tag{6}$$

$$\sum_{h_k=1}^{r_k} Z_{ih_k} = 1 \qquad \forall i, k \in l(i) \tag{7}$$

$$\sum_{j=1}^{m} \sum_{\mu=u-p_{ij}+1}^{u} X_{ij\mu} + \sum_{j=1}^{m} \sum_{\mu=u-p_{qj}+1}^{u} X_{qj\mu} \tag{8}$$
$$\leq 1-M\left(Z_{ih_k} + Z_{qh_k} - 2\right) \forall i < q, h, k \in l(i) \cap l(q), u$$

$$C_{\max} \geq 0 \tag{9}$$

$$C_i \geq 0 \qquad \forall i \tag{10}$$

$$X_{ij\mu} \in \{0, 1\} \qquad \forall i, j, u \tag{11}$$

$$Z_{ih_k} \in \{0, 1\} \qquad \forall i, k \in l(i) \tag{12}$$

EQUATIONS FOR INTEGER PROGRAMMING MODEL

The integer programming formulation for this workload allocation problem is given as follows:

P1:

$$\text{Minimize} \sum_{i \in \mathbf{I}} \sum_{j \in \mathbf{J}} \alpha_j^i \cdot U_{j,\Omega_j}^i + \sum_{t \in \mathbf{T}} \beta \cdot Z_t$$

Subject to

$$U_{j,t-1}^i \leq U_{j,t}^i \quad i \in \mathbf{I}, j \in \mathbf{J}, t = 1, \ldots, \Omega_j \tag{1}$$

$$\sum_{i \in \mathbf{I}} U_{j,t}^i = R_{j,t} \quad j \in \mathbf{J}, t \in \mathbf{T} \tag{2}$$

$$\sum_{j \in \mathbf{J}} U_{j,t}^i \leq K \cdot C \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{3}$$

$$U_{j,\Omega_j}^i \leq \hat{S}_1 \quad i \in \mathbf{I}, j \in \mathbf{J} \tag{4}$$

$$\sum_{j \in \mathbf{J}: \Omega_j = t} U_{j,t}^i \leq Z_t \quad i \in \mathbf{I}, t \in \mathbf{T} \tag{5}$$

$$U_{j,t}^i \in \{0, 1, 2, 3 \ldots\} \quad i \in \mathbf{I}, j \in \mathbf{J}, t \in \mathbf{T} \tag{6}$$

$$Z_t \geq 0 \quad t \in \mathbf{T} \tag{7}$$

GREEDY APPRAOCH

```
def container_reshuffling(container_stacks, export_time):
  """
  This function reshuffles the container stacks in order to minimize the number of
reshuffles needed to retrieve the containers with the earliest export time.
```

```
  Args:
    container_stacks: A list of lists, where each inner list represents a container stack.
    export_time: A list of integers, where each integer represents the export time of a
container.

  Returns:
    A list of lists, where each inner list represents the reshuffling of a container stack.
  """

  # Initialize the reshuffling of the container stacks.
  reshuffling = container_stacks

  # Iterate over the containers with the earliest export time.
  for i, container in enumerate(export_time):
    # Find the container stack that contains the container.
    container_stack = container_stacks[i]

    # Find the position of the container in the container stack.
    container_position = container_stack.index(container)
```