# TRUCK SCHEDULING AT CONTAINER TERMINAL

## MIE562F Project – Preliminary Results

**Team:** Dylan Camus, Ryan Do, Fan Jia, Matheus Magalhaes, Sugumar Prabhakaran
**Date:** 08 Nov 2020

## INTRODUCTION

This project considers the scheduling of movement of ocean containers from the container terminal (seaport) after discharge from a container ship to a ***transloading facility***. The transloading (cross-docking) facility transfers goods from the ocean containers to domestic containers owned by a distributor, and then loads the container on rail or road towards its next destination (typically a distribution center). To avoid financial penalties at the terminal and account for the varying processing times at the transloading facility and variance of ship arrivals, containers can be sent to a third external location called ***the stack*** for temporary storage at a fixed cost of entry but with much lower daily cost. (See Figure 1 for a diagram of locations).

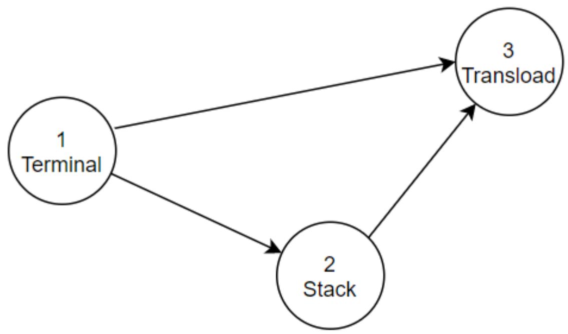



***Figure 1***: *Directed graph showing movement of containers between locations. Chassis may move bidirectionally between any two nodes.*

Containers are moved between locations using a container bed called a *chassis* attached to a truck. An external crane can pick up a container and place it on a chassis for transport. When a container is transported to the stack, it is removed from the chassis, and placed in a container pile. However, when a container is being processed at the transloading facility, the chassis is required to sit at the dock until the ocean container is empty because the crane at the transloading facility is unable to lift an ocean container full of goods. Therefore, the chassis must remain at the transloading facility until the container is processed. Chassis may move bidirectionally between any of the terminal, stack and transload.

## PROJECT OBJECTIVE

The objective of this project is to evaluate the effectiveness of two different scheduling methods to determine the optimal schedule of container transport (defined by minimizing total cost). The scheduling decision is deciding when and where to send each container after it arrives at the terminal. Containers may either be transported directly from the terminal to the transload facility or be sent to the stack for a certain amount of time, and then onto the transloading facility.

This scheduling problem can be considered as a variant of the resource-constrained Job Shop Scheduling Problem. Alongside the regular JSP constraints (release dates, non-overlap on resource, etc.), there are additional components of: assigning containers to legs, chassis travel time between jobs, the priority of the job must be considered, and the objective is to minimize cost based on the tardiness of soft deadlines from when a container is discharged (at Terminal and Transload), rather than the total makespan.

## SCHEDULING FACTORS

Several factors influence the optimal schedule:

- **Container Arrival Dates (Terminal).** We consider containers *released* when they are discharged from a container ship at the Terminal. Ships arrive on random dates and with a random quantity of ocean containers (Ex. Ship 1 arrives at time = 0 with 80 containers, Ship 2 arrives on time = 3 with 50 containers, etc.). We assume containers from a ship are all discharged on the same day.

- **Processing Capacity (Transloading Facility)**. Containers stay docked to chassis at the transloading facility. There is a processing time associated with processing a container which is typically 1-5 days. After processing, the chassis with the container is returned to the terminal.

- **Demurrage Costs (Tardiness at Terminal)**. Depending on the international shipping carrier, ocean containers have a few free storage days at the terminal after being discharged from a ship. After this period, there is a *Demurrage* cost incurred daily (Ex. Day 1-5: $0, Day 6+: $100/container/day).

- **Detention Costs ('Soft' Container Due Date)**. Depending on the international shipping carrier, ocean containers have a few weeks to be unloaded after being discharged from the ship. After this period, there is a *Detention* cost incurred daily until the container is returned to the terminal ($200/container/day). After processing, containers are returned to the carrier without need for chassis.

- **Container Priority**. Each container also has a priority level assigned by the company, which could be to prevent stock-outs of certain products further up the supply chain. These containers can have a priority of low, medium or high priority, defined by the intended distributor. Potential problem instances can consider a mix of priorities, such as a batch of high priority containers arriving on the same ship, or in quick succession. A cost penalty is assigned to each container at the time of departure from terminal which is scaled by the priority and time since container arrival (release date).

- **Storage Costs (Stack)**. At the stack, a fixed cost is charged for loading and unloading into the yard with a cheap cost incurred daily (Ex. $200 + [day 1→$15, day 2 → $30, etc..]).

- **Chassis Resource Constraints**. There is also a limited number of available chassis to move containers. Since a chassis is occupied for the duration of the transloading process, it cannot be used to move other containers from the terminal to the transloading facility or the stack. However, at the stack, this is no issue since the container is removed from the chassis. When the transloading process is finished, the chassis (with ocean container) are picked back up by a random truck and the ocean container is returned to the terminal. Between transporting successive containers, chassis need to travel between the first container's drop off point to the next container's pickup point.

## ASSUMPTIONS

- Chassis take the shortest path between start of jobs, with each leg constituting one job.
- A truck is available for whenever a chassis is scheduled to be transported.
- There are unlimited spots at the transloading facility for chassis to dock and be unloaded from.
- There is unlimited space at stack for containers.

## METHODOLOGY

For this problem, we used two approaches to obtain feasible or optimal solutions:

- Mixed Integer Programming; and
- Constraint Programming.

The formulation for each method and the preliminary results are outlined in the following two sections.

## MIXED-INTEGER PROGRAMMING (MIP) MODEL

*The MIP formulation for this truck scheduling problem.*

*Sets*:

| | |
|---|---|
| $C$ | $Set\ of\ chassis, c \epsilon C$ |
| $K$ | $Set\ of\ containers, k \epsilon K$ |
| $J$ | $Set\ of\ carriers, j \in J$ |
| $L$ = {1, 2, 3} | $Set\ of\ travel\ legs, l \epsilon L.$ $1: terminal\ to\ transload, process, and\ back\ to\ terminal,$ <br> $2: terminal\ to\ stack,$ <br> $3: stack\ to\ transload, process, and\ back\ to\ terminal$ |

*Parameters*:

| | |
|---|---|
| $\Phi_l$ | $Time\ duration\ of\ leg\ l\ [days]\ l\epsilon L$ |
| $D_{ll'}$ | $Delay\ required\ for\ a\ chassis\ [days] to\ start\ leg\ l'\ after\ completing\ l\ (includes\ travel)$ |
| $M$ | $some\ large\ number$ |
| $R_k$ | $Release\ time\ of\ container\ k\ [days]$ |
| $\Gamma_j$ | $Demurrage\ cost\ at\ terminal\ per\ unit\ time\ for\ carrier\ j\ after\ free\ period\ [^{\$}/_{container * days}]$ |
| $\Gamma"_j$ | $Free\ period\ before\ demurrage\ cost\ for\ carrier\ j\ at\ terminal\ [days]$ |
| $S$ | $fixed\ cost\ at\ stack\ [^{\$}/_{container}]$ |
| $S'$ | $unit\ cost\ at\ stack\ per\ unit\ time\ [^{\$}/_{container * days}]$ |
| $\delta"_j$ | $Free\ period\ before\ detention\ cost\ for\ some\ container\ of\ carrier\ j\ [days]$ |
| $\delta_j$ | $Detention\ cost\ per\ unit\ time\ for\ some\ container\ of\ carrier\ j\ [^{\$}/_{container * days}]$ |
| $Z_{jk}$ | $1\ if\ container\ k\ belongs\ to\ carrier\ j$ |
| $\rho_k$ | $Priority\ factor\ of\ container\ k, higher\ value\ is\ higher\ priority$ |

*Variables*:

| | |
|---|---|
| $x_{klc}$ | $1\ if\ container\ k\ is\ transported\ across\ leg\ l\ using\ chassis\ c, 0\ otherwise$ |
| $s_{klc}$ | $Start\ time\ of\ transporting\ container\ k\ across\ leg\ l\ using\ chassis\ c.$ |
| $z_{klk'l'c}$ | $1\ if\ the\ transport\ job\ of\ container\ k'\ for\ chassis\ c\ along\ leg\ l' is\ scheduled$ <br> $after\ the\ transport\ job\ of\ container\ k\ for\ chassis\ c\ along\ leg\ l.$ |

| | |
|---|---|
| $min \sum_{k\epsilon K} \sum_{c\epsilon C} \sum_{l\epsilon\{1,2\}} \sum_{j\epsilon J} \Gamma_j x_{klc} Z_{jk} * \max\{(s_{klc} - R_k - \Gamma_j^{"}), 0\}$ | Demurrage cost |
| $+ \sum_{k\epsilon K} \sum_{c\epsilon C} \sum_{l\epsilon\{1,3\}} \sum_{j\epsilon J} \delta_j x_{klc} Z_{jk} * \max\{(s_{klc} + \phi_{kl} - (R_k + \delta"_j)), 0\}$ | Detention cost |
| $+ \sum_{k\epsilon K} \sum_{c\epsilon C} x_{k2c} (S + S'(s_{k3c} - (s_{k2c} + \phi_{k2})))$ | Stack cost |
| $+ \sum_{k\epsilon K} \sum_{c\epsilon C} \sum_{l\epsilon\{1,2\}} \sum_{j\epsilon J} \rho_k x_{klc}(s_{klc} - R_k)$ | Priority cost |

| # | ***subject to***: | | ***description*** |
|---|---|---|---|
| 1 | $M(1 - x_{klc}) + M(1 - x_{k'l'c}) + (s_{klc} - s_{k'l'c}) \geq \Phi_{k'l'} + D_{l'l} - M * z_{klk'l'c}$ | $for\ all\ c \epsilon C,$<br>$for\ all\ k, k' \epsilon K$<br>$for\ all\ l, l' \epsilon L$<br>$s.t.$<br>$NOT\ AND$<br>$(k = k', l = l')$<br>$AND\ (k \leq k')$ | For each chassis, jobs can't overlap, and chassis may have to travel between jobs. For klc scheduled after k'l'c. |
| 1 | $M(1 - x_{klc}) + M(1 - x_{k'l'c}) + (s_{k'l'c} - s_{klc}) \geq \Phi_{kl} + D_{ll'} - M * (1 - z_{klk'l'c})$ | $for\ all\ c \epsilon C,$<br>$for\ all\ k, k' \epsilon K$<br>$for\ all\ l, l' \epsilon L$<br>$s.t.$<br>$NOT\ AND$<br>$(k = k', l = l')$<br>$AND\ (k \leq k')$ | For each chassis, jobs can't overlap, and chassis may have to travel between jobs. For k'l'c scheduled after klc. |
| 2 | $s_{klc} \geq R_k x_{klc}$ | $for\ all\ k \epsilon K,$<br>$for\ all\ l \epsilon L,$<br>$for\ all\ c \epsilon C$ | Release time constraint |
| 3 | $\sum_{c\epsilon C} x_{k2c} = \sum_{c\epsilon C} x_{k3c}$ | $for\ all\ k \epsilon K$ | Containers going through leg 2 must also go through leg 3 |
| 4 | $\sum_{c\epsilon C} x_{k2c} + \sum_{c\epsilon C} x_{k1c} = 1$ | $for\ all\ k \epsilon K$ | Containers go through leg 1 OR leg 2/3 |
| 5 | $M \sum_{c\epsilon C} x_{k1c} + \sum_{c\epsilon C} s_{k3c} \geq \sum_{c\epsilon C} s_{k2c} + \Phi_{k2}$ | $for\ all\ k \epsilon K$ | leg 2 to 3 precedence constraint |
| 6 | $M x_{klc} \geq s_{klc}$ | $for\ all\ k \epsilon K,$<br>$for\ all\ l \epsilon L,$<br>$for\ all\ c \epsilon C$ | When x is 0, s is 0. Necessary for the precedence constraint |
| 7 | $s_{klc} \geq 0$ | $for\ all\ k \in K$<br>$for\ all\ l \in L$<br>$for\ all\ c \in C$ | Domain of s |
| 8 | $x_{klc} = [0,1]$ | $for\ all\ k \in K$<br>$for\ all\ l \in L$<br>$for\ all\ c \in C$ | Domain of x |

The MIP model described aims to solve a scheduling-assignment hybrid problem that is to determine optimal start times for each container transport job while simultaneously assigning containers to chassis (resources) and determining which route to take for each chassis-container assignment (leg 1 vs. leg 2/3). The objective function for the optimization is to minimize demurrage costs (time containers spend at terminal), detention costs (time containers spend before being emptied and sent back to terminal), stack cost (time containers spend at stack), and priority cost (time containers spend at terminal scaled by priority factor).To re-iterate, at the transload facility chassis must stay with containers before returning them to the carrier at the terminal. As such, the legs are defined as the following (below, left):

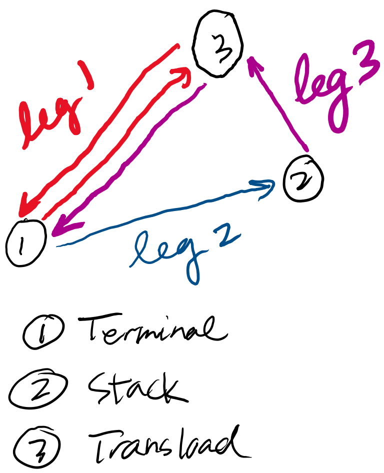



*Diagram 1. Transport leg definitions.*

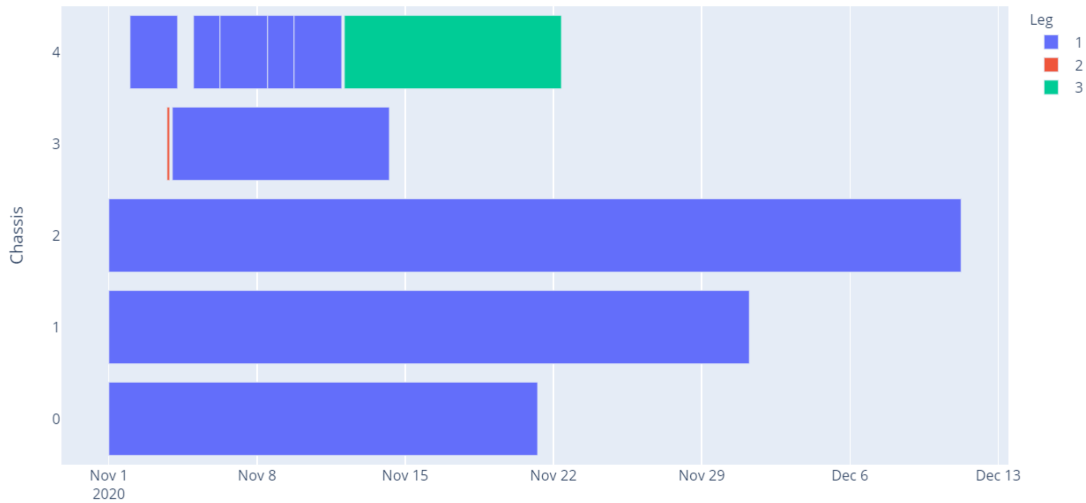
The decision variables include the start times of each container-chassis-leg combination. If a container-chassis-leg combination is not in the set of assignments $x_{klc} = 0$. In this case, $s_{klc}$ is also constrained to be 0 (Constraint 6, MIP Model). This makes Constraint 5 to be valid.

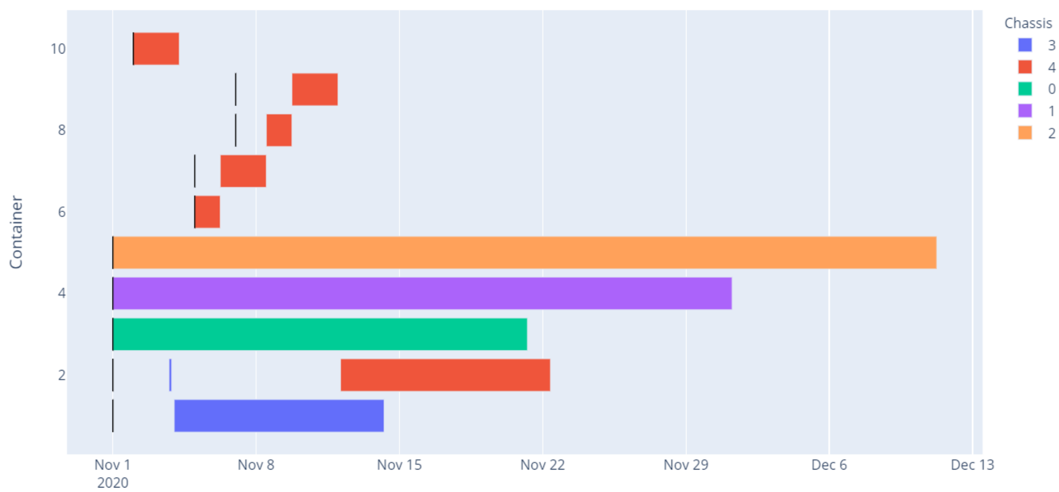
Containers assigned to a leg 2 transport job must also be assigned to a leg 3 transport job. (Constraint 3, MIP Model). It follows that the start time of each container leg 2 transport job must be at least $\Phi_{k2}$ time (travel time from (1) to (2), Diagram 1) before its leg 3 transport job start time (Constraint 5, MIP Model). The other option is to be assigned a leg 1 transport job (Constraint 4, MIP Model). Containers also have a release time at which they are released for transport from their respective carriers. Start times of all transport jobs must be at or after the respective release times of each container (Constraint 2, MIP Model).

Last but not least, for each individual chassis, they can only take 1 container at a time and must complete the leg-container combination before traveling to the next job. As such, this resource constraint is formulated as Constraint 1 where D represents the travel time between completion of leg l and start of leg l'. This inter-leg travel time is not modeled in the constraint programming model.

**MIP MODEL PRELIMINARY RESULTS**

## Test Instance 1 (base)

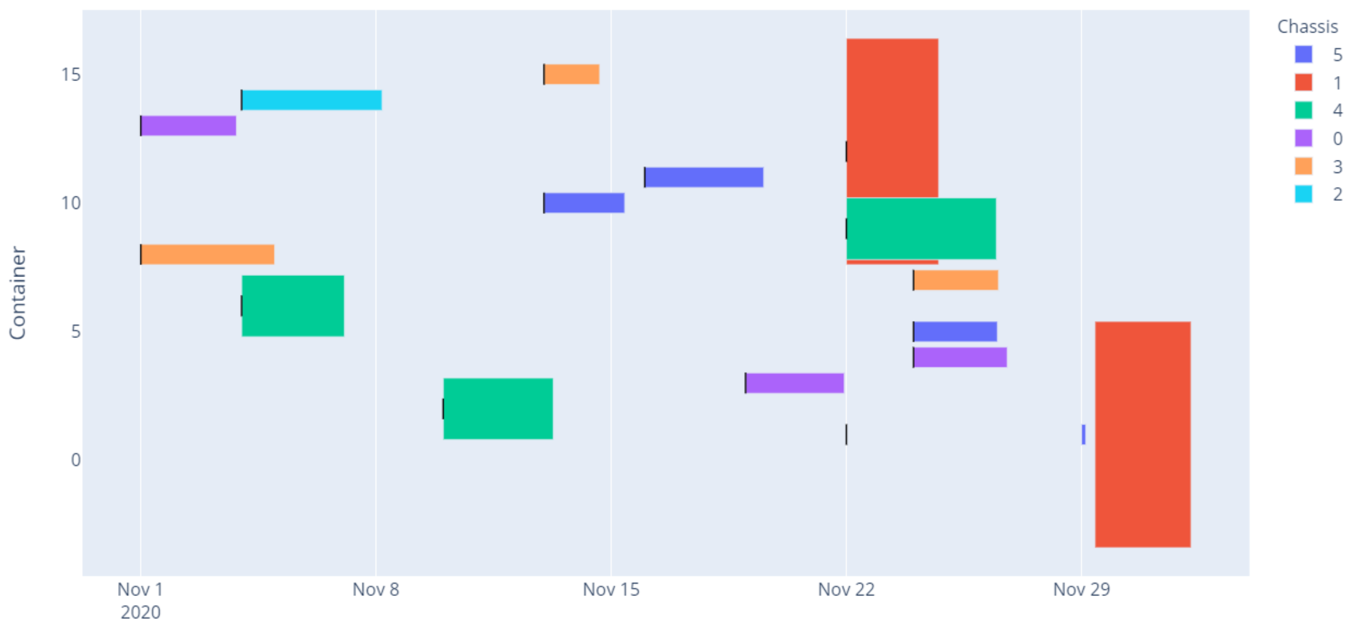



Black bars are release dates

**Test instance 2: 15_instance-- 15 containers -- 5 carriers -- 6 chassis -- 30-day release date horizon**

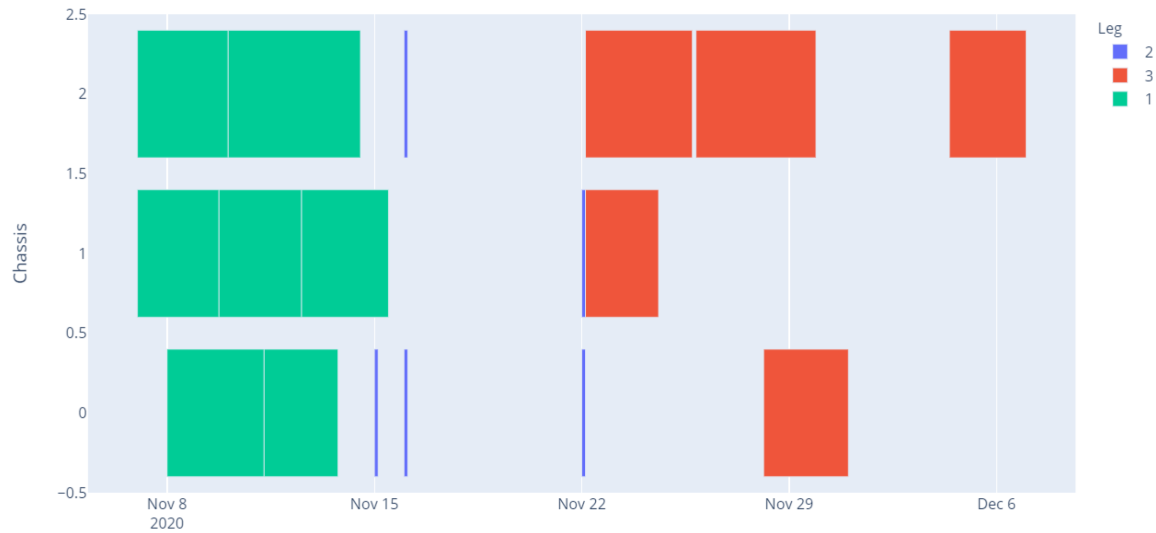
`13.32s, optimal.`

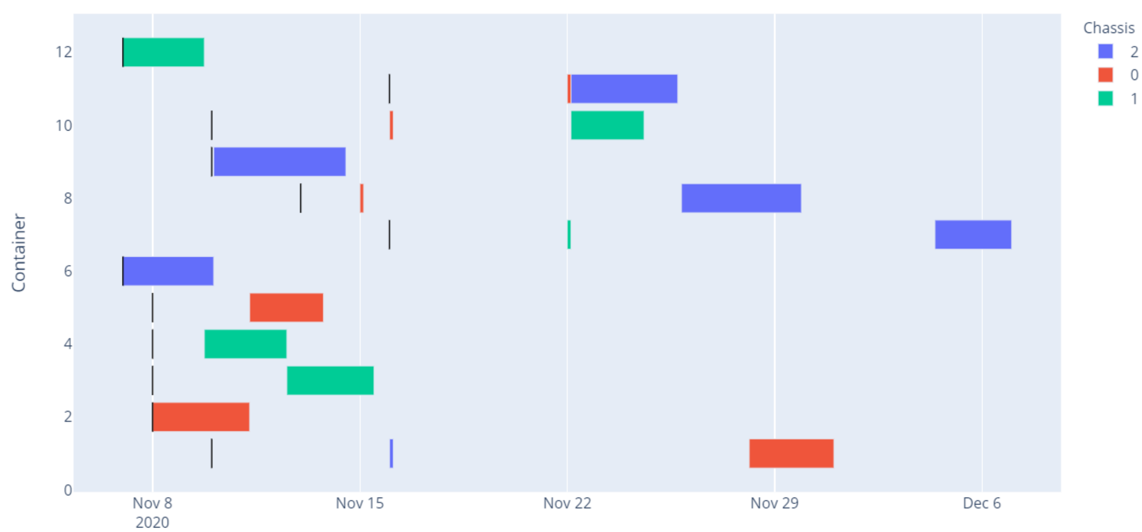




Black bars are release dates

**Test instance 3: 12_instance -- 12 containers -- 3 carriers -- 3 chassis -- 21-day release date horizon**

`11.77s, optimal.`

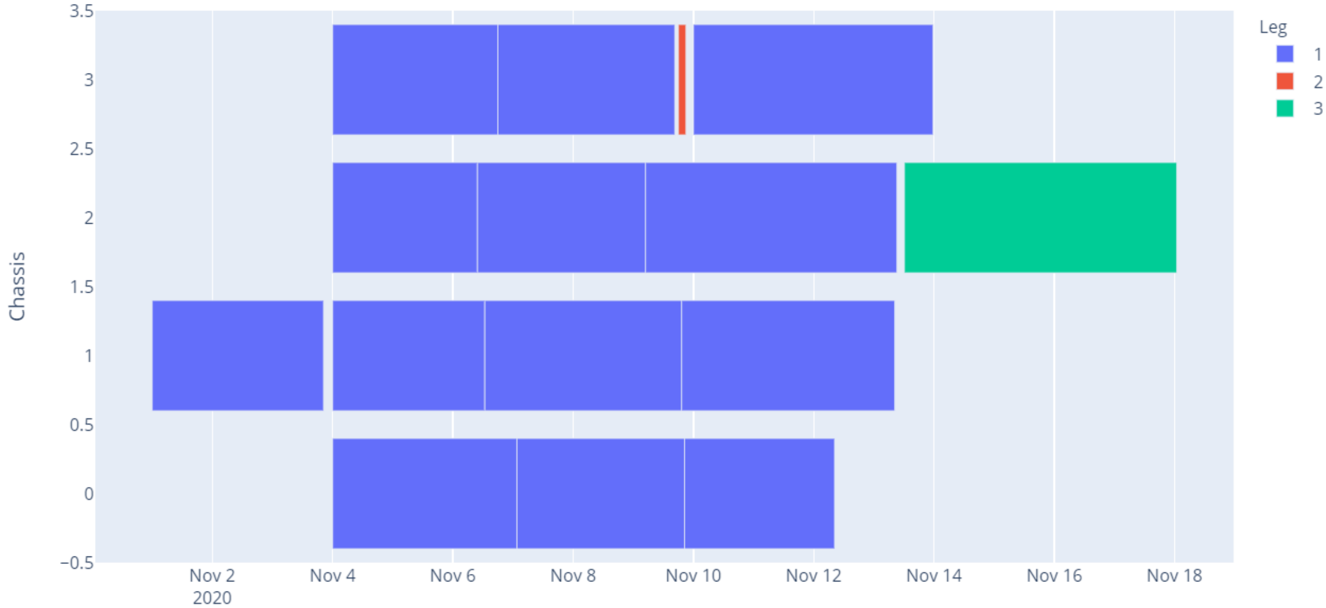


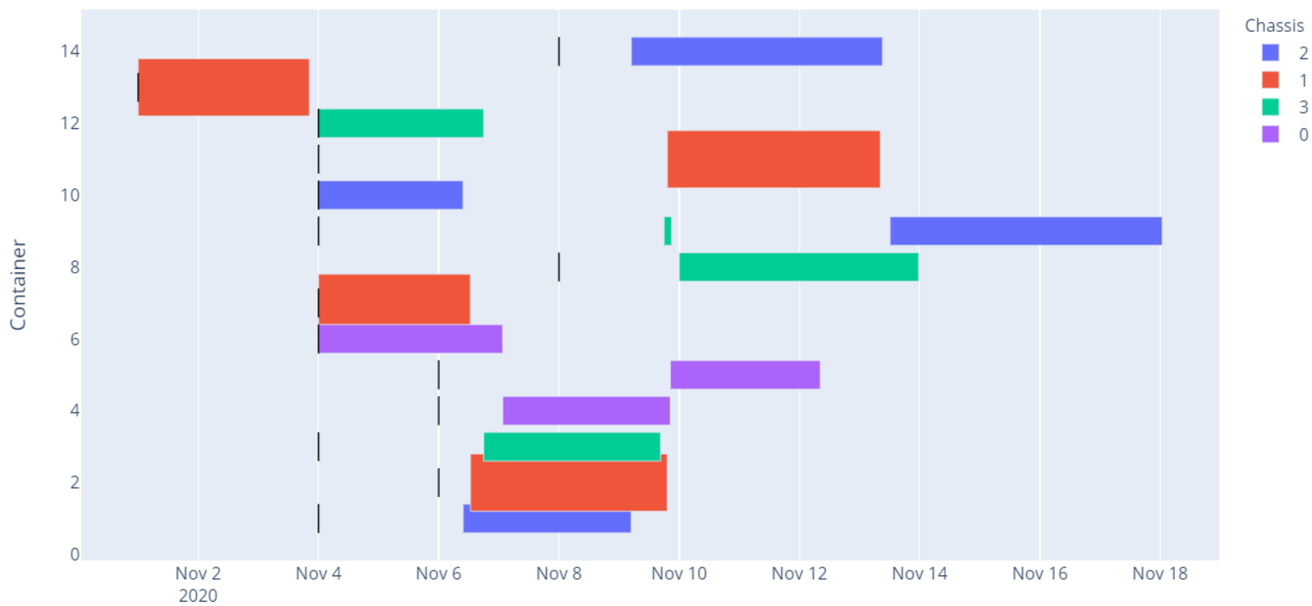




Black bars are release dates

**Test instance 4:  14_instance-- 14 containers -- 3 carriers -- 4 chassis -- 10-day release date horizon**

Feasible, run-time limit = 60 seconds

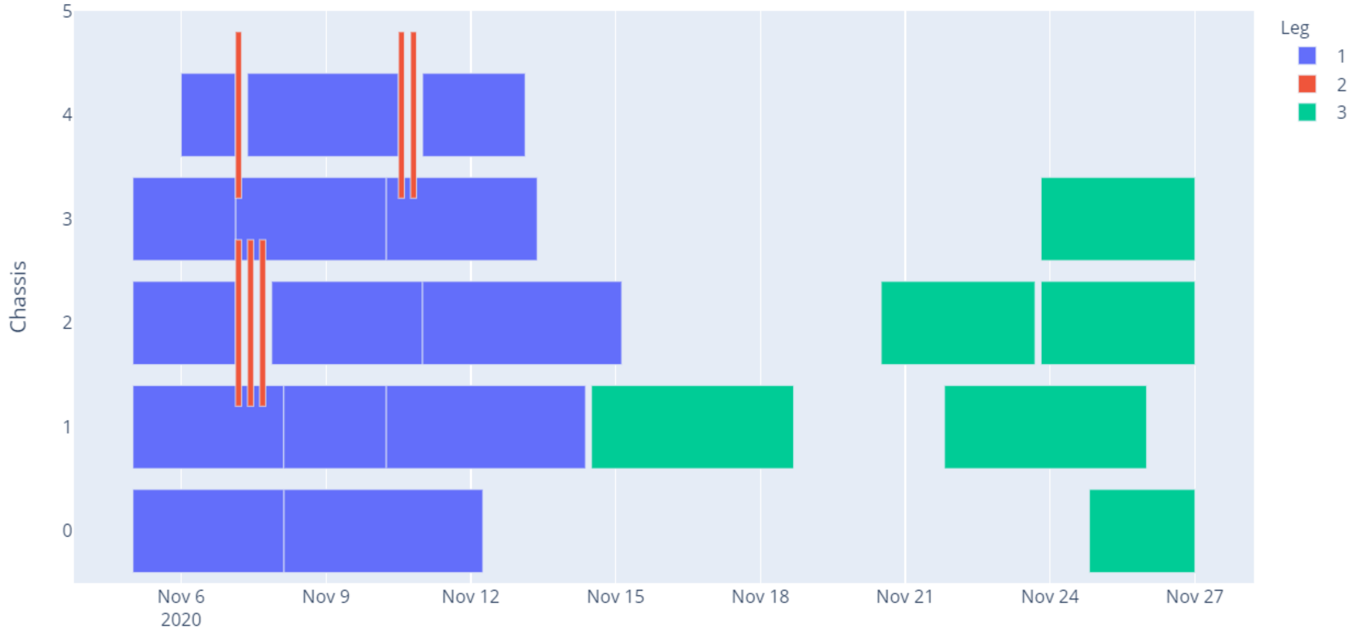


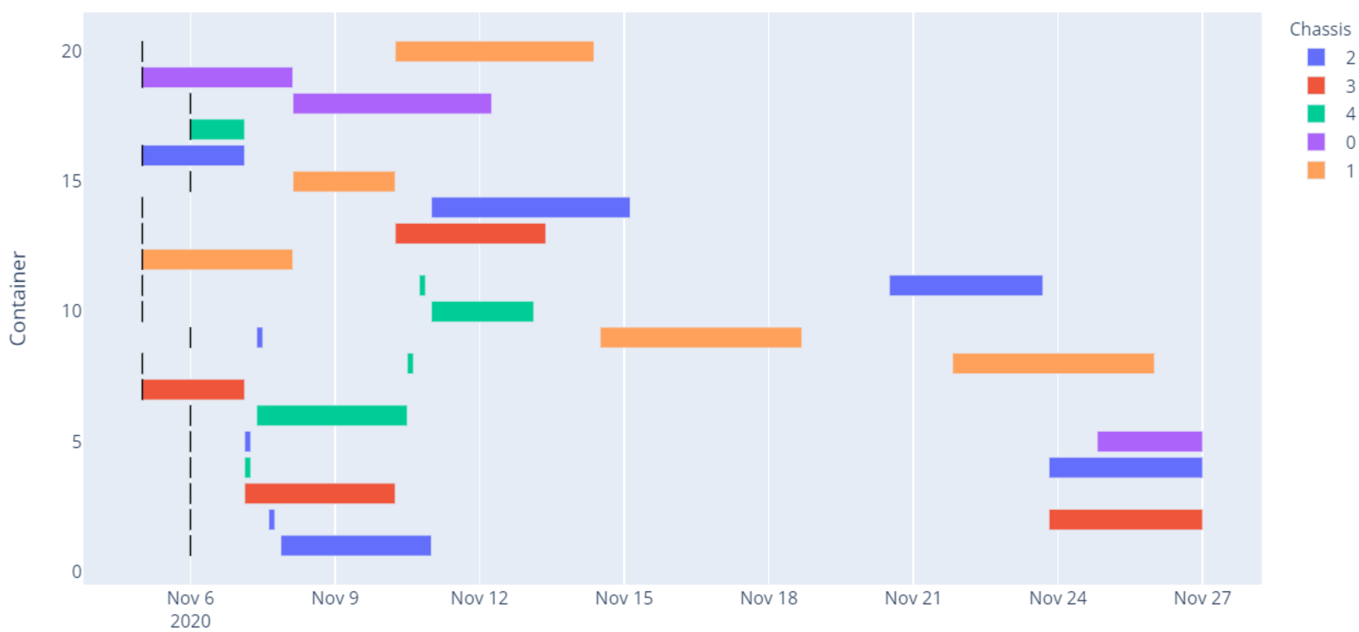




Black bars are release dates

**Test instance 5: 20_instance-- 20 containers -- 5 carriers -- 25 chassis -- 7-day release date horizon**

Feasible, run-time limit = 60 seconds




Black bars are release dates

## CONSTRAINT PROGRAMMING MODEL

*Sets*:

| | |
|---|---|
| $K$ | $Set\ of\ containers, k \in K$ |
| $C$ | $Set\ of\ chassis, c \in C$ |
| $J$ | $Set\ of\ carriers, j \in J$ |

*Parameters*:

| | |
|---|---|
| $R_k$ | $Release\ time\ of\ container\ k\ [days]$ |
| $\Gamma_j$ | $Demurrage\ cost\ at\ terminal\ per\ unit\ time\ for\ carrier\ j\ after\ free\ period\ [^{\$}/_{container * days}]$ |
| $\Gamma''_j$ | $Free\ period\ before\ demurrage\ cost\ for\ carrier\ j\ at\ terminal\ [days]$ |
| $S$ | $fixed\ cost\ at\ stack\ [^{\$}/_{container}]$ |
| $S'$ | $unit\ cost\ at\ stack\ per\ unit\ time\ [^{\$}/_{container * days}]$ |
| $\delta''_j$ | $Free\ period\ before\ detention\ cost\ for\ some\ container\ of\ carrier\ j\ [days]$ |
| $\delta_j$ | $Detention\ cost\ per\ unit\ time\ for\ some\ container\ of\ carrier\ j\ [^{\$}/_{container * days}]$ |
| $L$ | $Duration\ of\ legs\ to\ both\ stack\ and\ transload$ |
| $\rho\ _k$ | $Priority\ factor\ of\ container\ k, higher\ value\ is\ higher\ priority$ |

*Variables*:

| | |
|---|---|
| $transload_k$ | *Interval variable to represent time spent in stack of container k* |
| $trip\_leg_k$ | *Optional Interval variable that represents time spent travelling to either stack or transload* |
| $stack_k$ | *Optional Interval variable that represents time spent in stack of container k* |
| $chassis\_usage$ | $Elementary\ cumulative\ function\ variable, to\ represent\ the\ pooled\ chassis\ utilization$ |

*[CP Model]*

| | | | |
|---|---|---|---|
| $$min \sum_{k \epsilon K} \sum_{j \epsilon J} \Gamma_j * \max\{(Start\_of(trip\_leg_k) - R_k - \Gamma_j''), 0\}$$ | | | Demurrage Costs |
| $$+ \sum_{k \epsilon K} \sum_{j \epsilon J} \delta_j * \max\{(End\_of(transload_k) + Length\_of(stack_k) - R_k - \delta''_j), 0\}$$ | | | Detention Costs |
| $$+ \sum_{k \epsilon K} S * Presence\_of(stack_k) + S' * Length\_of(stack_k)$$ | | | Stack Costs |
| $$+ \sum_{k \epsilon K} \sum_{c \epsilon C} \rho_k * End\_of(transload_k)$$ | | | Priority Penalty Costs |
| **#** | ***subject to***: | | ***Description*** |
| 1 | $Length\_of(transload_k) = P_k$ | $for\ all\ k\ \epsilon\ K$ | For each container, the length of the transload interval is P_k |

| | | | |
|---|---|---|---|
| 2 | $Start\_of(trip\_leg_k) \geq R_k$ | $for\ all\ k \epsilon K$ | Release time constraint denoting arrival time of ships |
| 3 | $Length\_of(trip\_leg_k) = L$ | $for\ all\ k \epsilon K$ | Length of trips are predefined as L |
| 4 | $End\_Before\_Start(trip\_leg_k, stack_k)$<br>$End\_Before\_Start(trip\_leg_k, transload_k)$<br>$End\_Before\_Start(stack_k, transload_k)$ | $for\ all\ k \epsilon K$ | End_Before_Start global constraints to respect precedences on container k |
| 5 | $Chassis\_usage += Pulse(transload_k, 1)$ | $for\ all\ k \epsilon K$ | Cumulation resource expression, where transload_k seizes 1 chassis resource for the interval duration |
| 6 | $Chassis\_usage += Pulse(trip\_leg_k, 1)$ | $for\ all\ k \epsilon K$ | Seize 1 chassis resource for trip length |
| 7 | $Chassis\_usage \leq \|C\|$ | | Chassis usage must always be below the number of total available chassis |

**CP Resource Modelling**

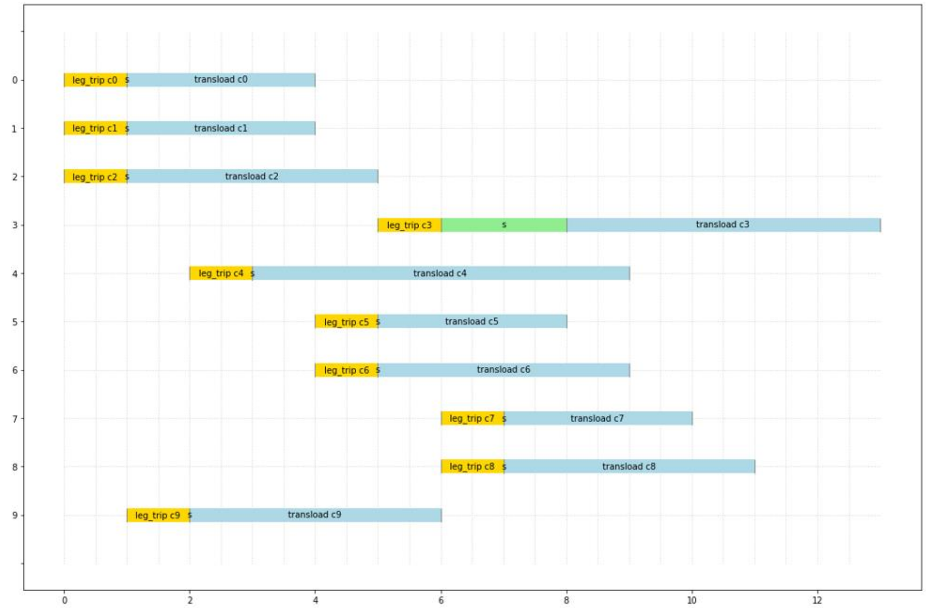
As a concept, interval variables in Constraint Programming are useful for modeling scheduling problems, having a start date, end date, and length of time which an activity was performed. If specified, interval variables can be excluded from a model. As such, the main idea behind the CP model was to represent the time in the transload facility as an interval that *must* be included, and an *optional* interval of time to represent the time spent in the stack, if needed.

The truck scheduling problem of the chassis resource is very similar to the unrelated parallel machine scheduling problem. There are 2 methods to model resources in scheduling, using Cumulative or Unary resources. In the MIP model, the chassis resources are modeled as Unary resources, with the movement between each task for a given resource all being tracked sequentially, much like a sequence of jobs on a machine in JSP. One of the simplifications made compared to the MIP, is that the CP model the resources are modeled as a pool of combined resources using a elementary cumulative resource function called Pulse (5)(6), which seizes a specified number of resources from a pool (Chassis_usage) for the length of the interval variable, and automatically released once completed. There is a single additional constraint which caps the number of actively used chassis below a specified threshold (7). The number of legs considered has reduced to the distance between terminal-transload and terminal-stack, which both seize a chassis resource for the container to be moved.

The objective function is very similar to the MIP version: summing over all demurrage penalty costs for tardiness of leaving the shipping terminal, the detention costs to minimize the tardiness of transload completion for all containers, the optional stack costs simplified by the interval variable presence values included in the optimal solution, and the priority costs being a function of completion time at transload. Constraints (1)-(3) specify the interval lengths at transload using P_k, start times greater than the arrival time of the ships arriving in port, and specifying the length of the leg trip. The End_Before_Start global constraints in (4) ensure that precedence constraints, regardless of the path that a container decides to take, are met.

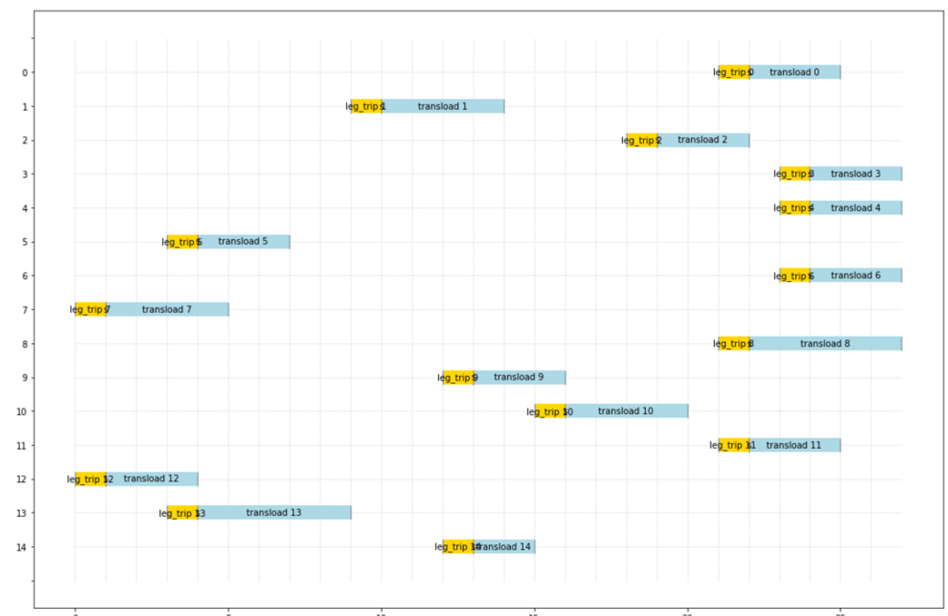
<u>**CP MODEL PRELIMINARY RESULTS**</u>

**Test instance 1: test_instance**



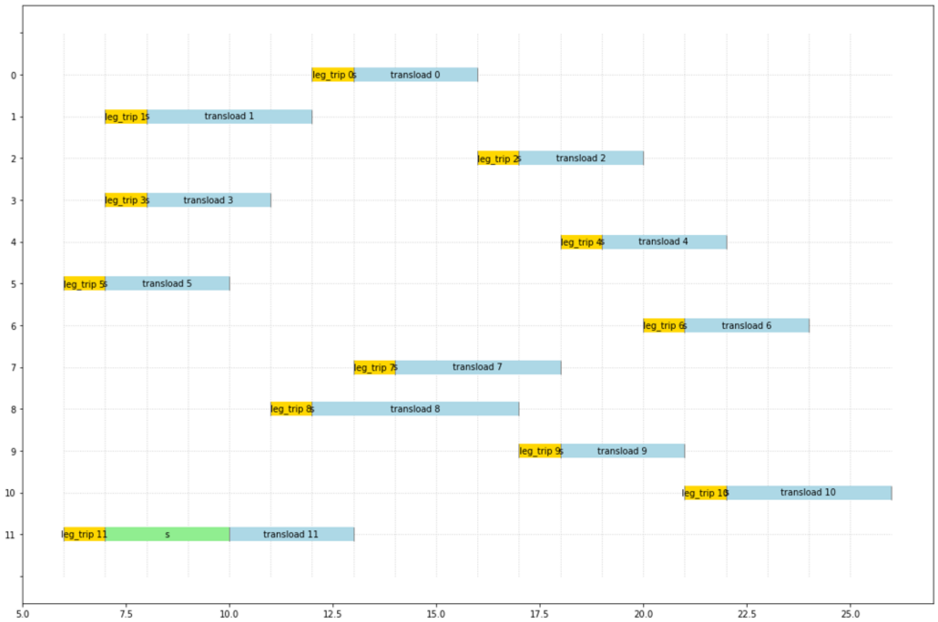

**Test instance 2: 15_instance-- 15 containers -- 5 carriers -- 6 chassis -- 30-day horizon**

Optimal, 0.02 sec



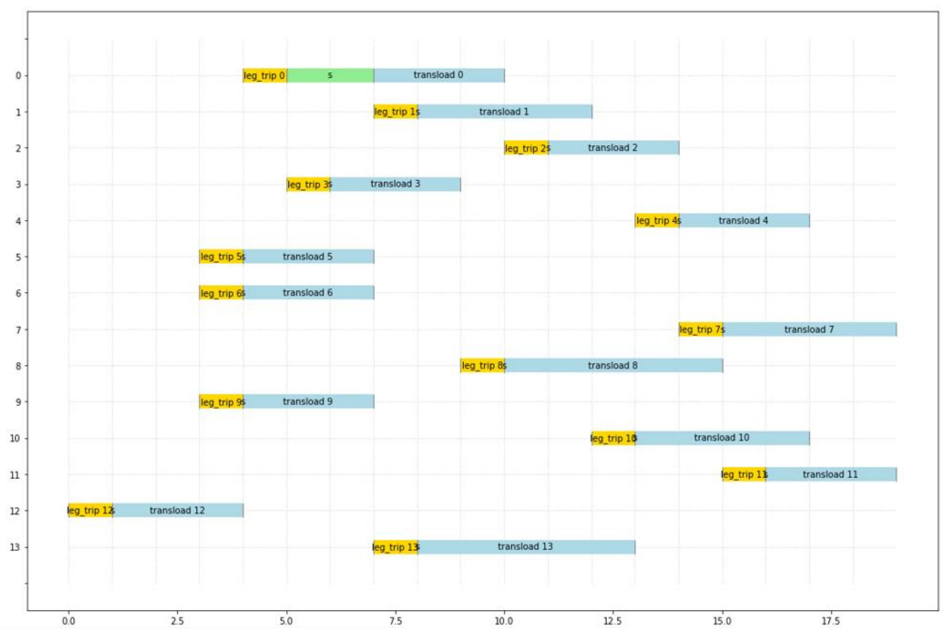

**Test instance 3: 12_instance -- 12 containers -- 3 carriers -- 3 chassis -- 21-day horizon**

Optimal, 6.69 sec



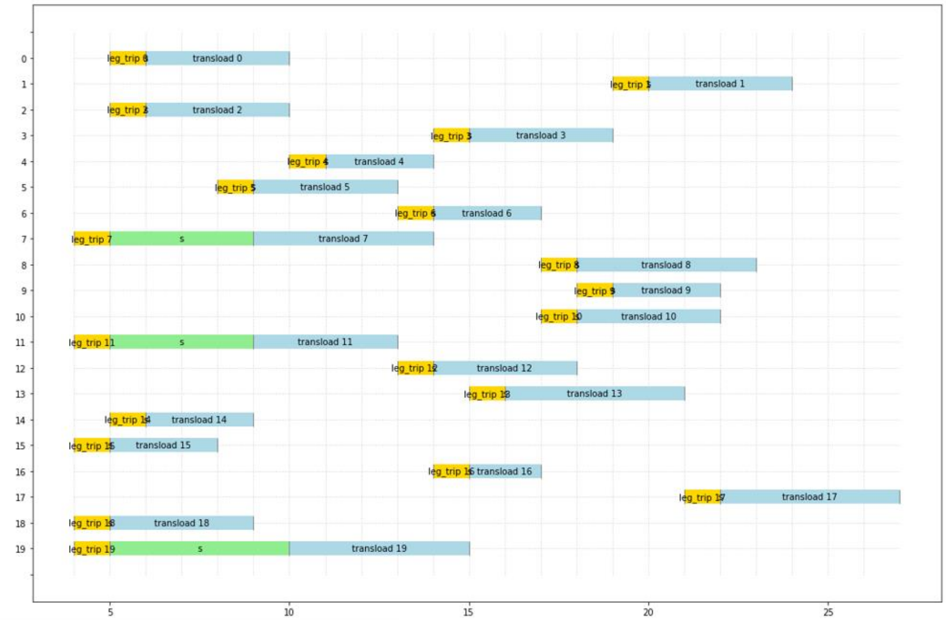

**Test instance 4: 14_instance-- 14 containers -- 3 carriers -- 4 chassis -- 10-day horizon**

Optimal, 22.74 sec




**Test instance 5: 20_instance-- 20 containers -- 5 carriers -- 25 chassis -- 7-day horizon**

Feasible, run time limit = 60 seconds



leg_trip 0
transload 0
leg_trip 1
transload 1
leg_trip 2
transload 2
leg_trip 3
transload 3
leg_trip 4
transload 4
leg_trip 5
transload 5
leg_trip 6
transload 6
leg_trip 7
s
transload 7
leg_trip 8
transload 8
leg_trip 9
transload 9
leg_trip 10
transload 10
leg_trip 11
s
transload 11
leg_trip 12
transload 12
leg_trip 13
transload 13
leg_trip 14
transload 14
leg_trip 15
transload 15
leg_trip 16
transload 16
leg_trip 17
transload 17
leg_trip 18
transload 18
leg_trip 19
s
transload 19
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
5
10
15
20
25