Raayan Mohtashemi, Nicolas Whittle, Bradley Turcios, Peter Liu
April 19, 2021
IEOR 4405
Professor Clifford Stein

Port Truck Scheduling Problem

Introduction

The port is a very important part in the supply chain between sea and land transportation. It serves as the major entrance of goods from all over the world, as well as the exit door for domestic products to enter international markets. From an operational perspective, it is a very complex system with different subsystems that seeks to maximize the volume of transported goods within constraints of time and capital, or one that seeks to minimize the time of transporting given a certain amount of capital-related resources and a fixed amount of goods. In this report, we focus on finding an optimal schedule for trucks and the optimal number of trucks to minimize the maximum makespan to unload or reload a container ship. We model the operations at DP World Port in Lirquén, Chile as a job shop scheduling problem simplified from four elements (ships, cranes, containers, and trucks) to two elements (containers as jobs and trucks as machines) that are dependent on release dates determined by cranes with constant and/or normally distributed crane rates and determined by ships with constant workloads per crane. When ships come into the port, cranes unload containers onto the trucks, releasing the first container at time 0. The DP World Port Team requested we look into whether assigning trucks dynamically as an entire fleet to any available cranes would lead to a faster unloading/offloading process as opposed to the current operations of mini fleets of trucks assigned to work solely with individual cranes. By discussing the cases where trucks pick up containers from a ship as several smaller fleets and as one part of a larger fleet, we find that:

1. When trucks are assigned as small fleets, an increase in the number of trucks that takes part in the fleet assigned to each crane will decrease the maximum makespan of that particular crane until too many trucks are added, which begins to increase the idle times on trucks. An optimal schedule that guarantees the minimum idle time of the cranes in this case would be to assign the trucks to containers by their “true” release dates (the release dates determined by the crane being able to drop off containers without having to idle), if possible.
2. When trucks are assigned as part of a larger fleet and when modeling crane rates and truck travel times (processing times) as normally distributed random variables, we are able to achieve a similar makespan as in the first case by using fewer trucks in total.

Data Sources

Our main data source was the DP World port operations team. We spoke with several employees in a video conference and maintained frequent contact over email. The team explained the system components of the port and provided data about necessary elements. This information provided the basis for our model. The components included:

- How long it takes for trucks to take a container from the ship to the yard and then return to the ship.
    - 12 minutes

- Total number of trucks in full fleet
  - 24
- Number of cranes
  - 6
  - All cranes serve the same ship at the same time
- Average number of jobs each crane handles
  - 275
- Average crane rate (the number of containers moved by a crane per hour)
  - 16 movements per hour = 3.75 minutes per container movement

Problem Formulation

After our initial meetings and correspondence with the DP World team, we decided to split the problem into 2 problems. The DP World team was asking which onloading and offloading method would require fewer trucks overall:

1. Continuing their current practice of assigning trucks in smaller, fixed fleets to individual cranes
2. Assigning individual trucks to cranes from a single, larger fleet of trucks dynamically

The common restrictions for both problems are the parameters of operations of the port, where they currently own 24 trucks.

- The release date of each container is subject to the previous one being picked up. In other words, the crane cannot operate again until the truck assigned to the container has picked it up because there is no storage space in the dock.
- The processing time of the trucks are very similar for each one of the containers.
- The ships that dock at this port generally have between 250-300 containers to be unloaded or re-loaded per crane. We averaged this to 275 containers per crane per ship

We modeled two versions of each problem: deterministic and stochastic. In the Deterministic version of both problems we assumed that the parameters of operations were fixed. The crane rate was considered 3.75 minutes per container for all cranes and all containers and the processing time of each truck was 12 minutes for all trips.

For the stochastic version we modeled the parameters of operations not as fixed but as a normal distribution for each job. For each crane rate we considered a normal distribution with mean 3.75 and a standard deviation of 0.3. For the processing times, we assumed a normal distribution with mean 12 and standard deviation of 1.

Othar extensions of this problem are to calculate and minimize the idle time of machines as well as containers/cranes, and also get a better understanding of the port’s double cycling process. Eventually, we want to take into account that some cranes might need to handle more containers than others. We would eventually like to create a tool that allows the port operations staff to understand how to optimize the trucks for any individual ship with a specific number of containers.

Problem 1

The challenge of problem 1 is to find the optimum number of trucks to assign to each one of the 6 cranes they have. It looks like a P|r_j|Cmax problem for each one of the cranes, where P is the number of trucks to assign depicted as machines with same speeds. R_j is the release time of each one of the containers and we look to minimize the maximum makespan of the system. In other words, we are looking to find the max(Cmax) of all six cranes.

Considering all the parameters of operations mentioned before, and considering a uniform workload of the cranes, it is reasonable to predict that the maximum number of trucks to assign to each crane must be 24/6 = 4 or else some other crane will be left with fewer trucks, increasing the makespan of the system.

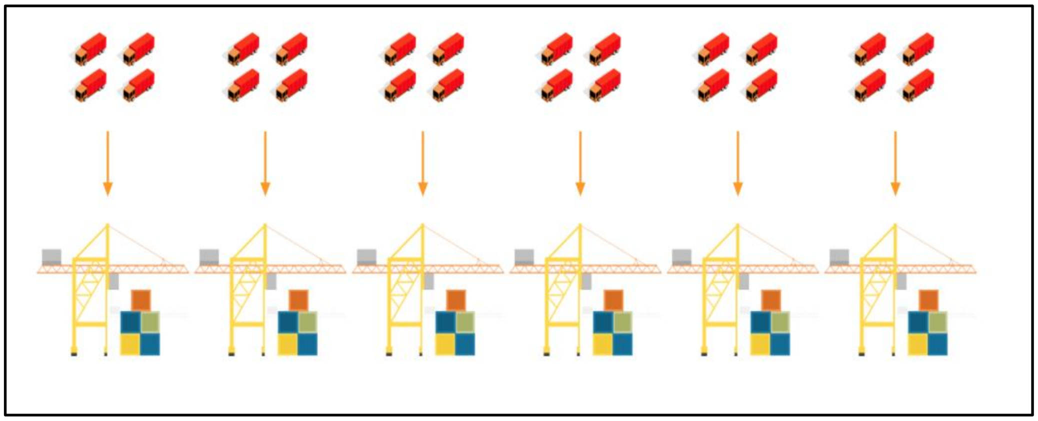


Figure 1: Problem 1 diagram

Problem 2

Problem 2 is modelled exactly the same as problem 1: P|r_j|Cmax, where P is the number of trucks in the whole system depicted as machines with same speeds. R_j is the release time of each one of the containers and we look to minimize the maximum makespan of the system.
The release times for this problem are not as trivial as the one before, where the data was given by the company and it was measured for only one crane. For this problem we had to come up with a way of finding the rate of operation of all 6 cranes combined. We found that in the long run, the rate of operations of the six cranes is approximately the rate of operation of one crane divided by the number of cranes, in this case 6.

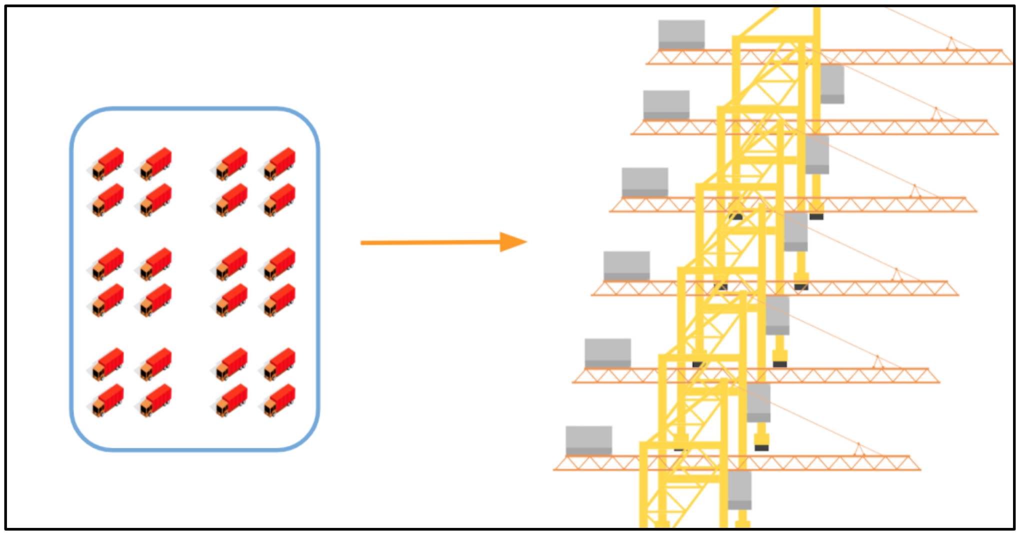
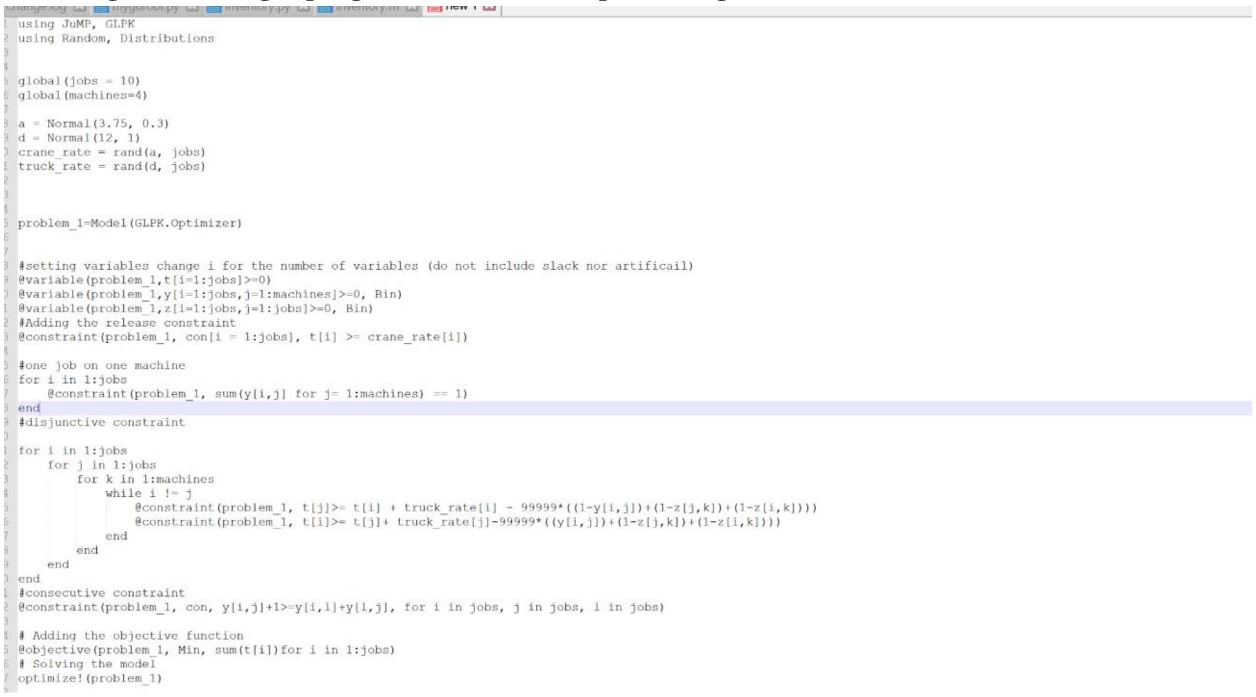


Figure 2: Problem 2 diagram

Methods

After researching job matching problems, we found an integer program formulation that we felt we could modify to schedule the minimum number of trucks to find the optimal Cmax. However, the program had a long run-time and would require significant work to simplify assumptions. We used excel and simpy for subsequent analysis and simulation, with the intent of returning to the integer program as a future step working further with the DP World Port Team.



```
using JuMP, GLPK
using Random, Distributions


global(jobs = 10)
global(machines=4)

a = Normal(3.75, 0.3)
d = Normal(12, 1)
crane_rate = rand(a, jobs)
truck_rate = rand(d, jobs)



problem_1=Model(GLPK.Optimizer)


#setting variables change i for the number of variables (do not include slack nor artificail)
@variable(problem_1,t[i=1:jobs]>=0)
@variable(problem_1,y[i=1:jobs,j=1:machines]>=0, Bin)
@variable(problem_1,z[i=1:jobs,j=1:jobs]>=0, Bin)
#Adding the release constraint
@constraint(problem_1, con[i = 1:jobs], t[i] >= crane_rate[i])

#one job on one machine
for i in 1:jobs
    @constraint(problem_1, sum(y[i,j] for j= 1:machines) == 1)
end
#disjunctive constraint

for i in 1:jobs
    for j in 1:jobs
        for k in 1:machines
            while i != j
                @constraint(problem_1, t[j]>= t[i] + truck_rate[i] - 99999*((1-y[i,j])+(1-z[j,k])+(1-z[i,k])))
                @constraint(problem_1, t[i]>= t[j]+ truck_rate[j]-99999*((y[i,j])+(1-z[j,k])+(1-z[i,k])))
            end
        end
    end
end
#consecutive constraint
@constraint(problem_1, con, y[i,j]+1>=y[i,l]+y[l,j], for i in jobs, j in jobs, l in jobs)

# Adding the objective function
@objective(problem_1, Min, sum(t[i])for i in 1:jobs)
# Solving the model
optimize!(problem_1)
```

Figure 3: Code Snippet from Integer Program. The run-time was too long to be useful, so we moved on to using excel and simpy.

Excel

We decided to start out by writing our assumptions into excel spreadsheets for the deterministic schedules. With deterministic schedules, many of our assumptions remained constant.

Crane Movements per Hour	16
Container Release Interval (r_j)	3.75
Number of Jobs Per Ship (n)	275
Round Trip Truck Time (p_j)	12

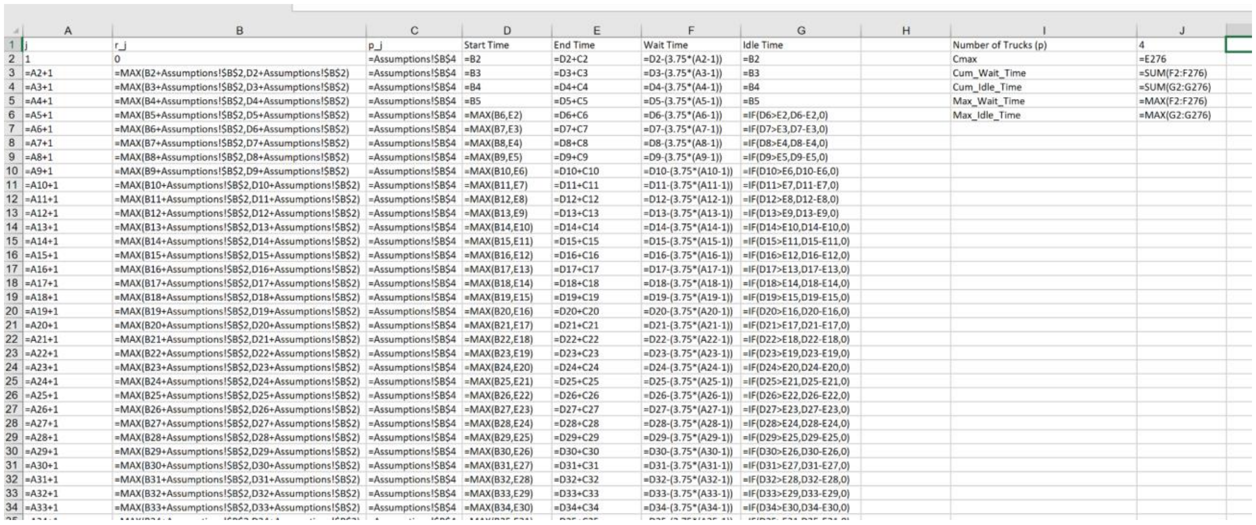
Figure 4: Excel Assumptions

Using these assumptions, we populated an excel spreadsheet with 275 jobs with constant processing times, and calculated their associated release dates, job start times, job end times, waiting time for jobs, and idle time for machines. For release dates, we assumed that only once a container is released, the crane movement for the next container starts. In other words, r_j = max(r_j-1 + 3.75, start time of previous job + 3.75). We created 6 separate sheets for the deterministic problem, representing 1-to-6 truck fleet sizes. In the deterministic case, earlier jobs always finish before later jobs, which translates to trucks always being ready for another container in the same order. Thus, our start time for new jobs is max(release date of job, and the end time of the job being processed by the same truck). End times are the start time plus processing time.

Idle time has to do with the amount of time that trucks are waiting for a container. We used an if/then comparison to classify idle time based on whether the truck accomplishing the job finished it's previous job before the release date of the current job.

Wait time is meant to calculate the amount of time a container waits to be put on a truck. In this framework, the wait time has two components. One more straightforward component is the amount of time the container spends hanging off the crane until it is placed onto a truck. The other component is the amount of time the container waits for a crane to pick it up in addition to the fastest schedule of release times. In other words, because the crane itself has to wait until a truck picks up the container it is currently holding, there is an added component of wait time for the containers still on the ship that wouldn't otherwise exist if a crane could drop off containers at will.

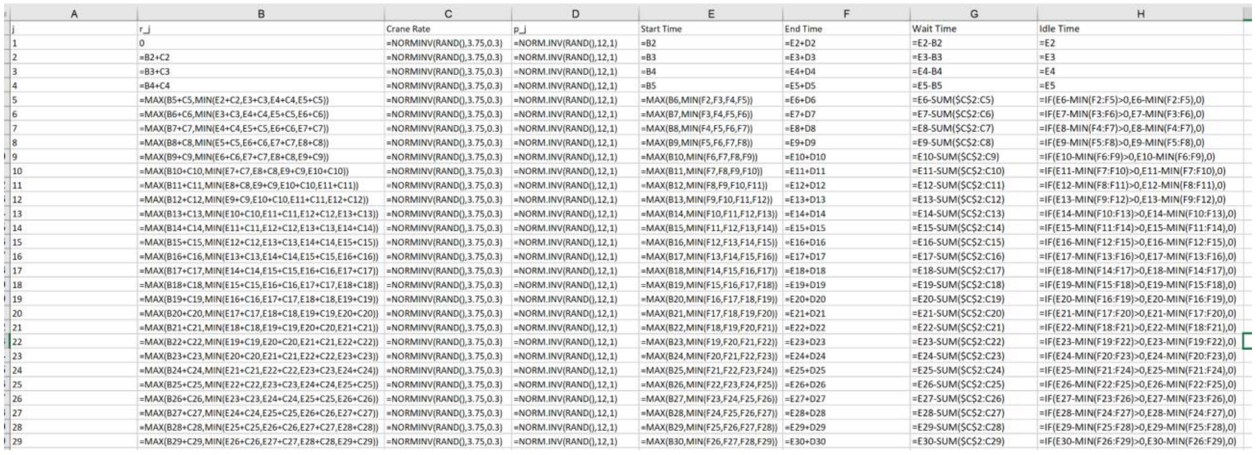
For the second problem deterministic case (trucks dynamically scheduled as part of one fleet), the results are trivially the same because the situation can be modeled as six jobs being released at the same time, with the same crane rates and processing times. Because processing times and crane rates are deterministic, the same trucks would service at the same intervals.



	A	B	C	D	E	F	G	H	I	J
1	j	r_j	p_j	Start Time	End Time	Wait Time	Idle Time		Number of Trucks (p)	4
2	1	0	=Assumptions!$B$4	=B2	=D2+C2	=D2-(3.75*(A2-1))	=B2		Cmax	=E276
3	=A2+1	=MAX(B2+Assumptions!$B$2,D2+Assumptions!$B$2)	=Assumptions!$B$4	=B3	=D3+C3	=D3-(3.75*(A3-1))	=B3		Cum_Wait_Time	=SUM(F2:F276)
4	=A3+1	=MAX(B3+Assumptions!$B$2,D3+Assumptions!$B$2)	=Assumptions!$B$4	=B4	=D4+C4	=D4-(3.75*(A4-1))	=B4		Cum_Idle_Time	=SUM(G2:G276)
5	=A4+1	=MAX(B4+Assumptions!$B$2,D4+Assumptions!$B$2)	=Assumptions!$B$4	=B5	=D5+C5	=D5-(3.75*(A5-1))	=B5		Max_Wait_Time	=MAX(F2:F276)
6	=A5+1	=MAX(B5+Assumptions!$B$2,D5+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B6,E2)	=D6+C6	=D6-(3.75*(A6-1))	=IF(D6>E2,D6-E2,0)		Max_Idle_Time	=MAX(G2:G276)
7	=A6+1	=MAX(B6+Assumptions!$B$2,D6+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B7,E3)	=D7+C7	=D7-(3.75*(A7-1))	=IF(D7>E3,D7-E3,0)			
8	=A7+1	=MAX(B7+Assumptions!$B$2,D7+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B8,E4)	=D8+C8	=D8-(3.75*(A8-1))	=IF(D8>E4,D8-E4,0)			
9	=A8+1	=MAX(B8+Assumptions!$B$2,D8+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B9,E5)	=D9+C9	=D9-(3.75*(A9-1))	=IF(D9>E5,D9-E5,0)			
10	=A9+1	=MAX(B9+Assumptions!$B$2,D9+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B10,E6)	=D10+C10	=D10-(3.75*(A10-1))	=IF(D10>E6,D10-E6,0)			
11	=A10+1	=MAX(B10+Assumptions!$B$2,D10+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B11,E7)	=D11+C11	=D11-(3.75*(A11-1))	=IF(D11>E7,D11-E7,0)			
12	=A11+1	=MAX(B11+Assumptions!$B$2,D11+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B12,E8)	=D12+C12	=D12-(3.75*(A12-1))	=IF(D12>E8,D12-E8,0)			
13	=A12+1	=MAX(B12+Assumptions!$B$2,D12+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B13,E9)	=D13+C13	=D13-(3.75*(A13-1))	=IF(D13>E9,D13-E9,0)			
14	=A13+1	=MAX(B13+Assumptions!$B$2,D13+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B14,E10)	=D14+C14	=D14-(3.75*(A14-1))	=IF(D14>E10,D14-E10,0)			
15	=A14+1	=MAX(B14+Assumptions!$B$2,D14+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B15,E11)	=D15+C15	=D15-(3.75*(A15-1))	=IF(D15>E11,D15-E11,0)			
16	=A15+1	=MAX(B15+Assumptions!$B$2,D15+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B16,E12)	=D16+C16	=D16-(3.75*(A16-1))	=IF(D16>E12,D16-E12,0)			
17	=A16+1	=MAX(B16+Assumptions!$B$2,D16+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B17,E13)	=D17+C17	=D17-(3.75*(A17-1))	=IF(D17>E13,D17-E13,0)			
18	=A17+1	=MAX(B17+Assumptions!$B$2,D17+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B18,E14)	=D18+C18	=D18-(3.75*(A18-1))	=IF(D18>E14,D18-E14,0)			
19	=A18+1	=MAX(B18+Assumptions!$B$2,D18+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B19,E15)	=D19+C19	=D19-(3.75*(A19-1))	=IF(D19>E15,D19-E15,0)			
20	=A19+1	=MAX(B19+Assumptions!$B$2,D19+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B20,E16)	=D20+C20	=D20-(3.75*(A20-1))	=IF(D20>E16,D20-E16,0)			
21	=A20+1	=MAX(B20+Assumptions!$B$2,D20+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B21,E17)	=D21+C21	=D21-(3.75*(A21-1))	=IF(D21>E17,D21-E17,0)			
22	=A21+1	=MAX(B21+Assumptions!$B$2,D21+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B22,E18)	=D22+C22	=D22-(3.75*(A22-1))	=IF(D22>E18,D22-E18,0)			
23	=A22+1	=MAX(B22+Assumptions!$B$2,D22+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B23,E19)	=D23+C23	=D23-(3.75*(A23-1))	=IF(D23>E19,D23-E19,0)			
24	=A23+1	=MAX(B23+Assumptions!$B$2,D23+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B24,E20)	=D24+C24	=D24-(3.75*(A24-1))	=IF(D24>E20,D24-E20,0)			
25	=A24+1	=MAX(B24+Assumptions!$B$2,D24+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B25,E21)	=D25+C25	=D25-(3.75*(A25-1))	=IF(D25>E21,D25-E21,0)			
26	=A25+1	=MAX(B25+Assumptions!$B$2,D25+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B26,E22)	=D26+C26	=D26-(3.75*(A26-1))	=IF(D26>E22,D26-E22,0)			
27	=A26+1	=MAX(B26+Assumptions!$B$2,D26+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B27,E23)	=D27+C27	=D27-(3.75*(A27-1))	=IF(D27>E23,D27-E23,0)			
28	=A27+1	=MAX(B27+Assumptions!$B$2,D27+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B28,E24)	=D28+C28	=D28-(3.75*(A28-1))	=IF(D28>E24,D28-E24,0)			
29	=A28+1	=MAX(B28+Assumptions!$B$2,D28+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B29,E25)	=D29+C29	=D29-(3.75*(A29-1))	=IF(D29>E25,D29-E25,0)			
30	=A29+1	=MAX(B29+Assumptions!$B$2,D29+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B30,E26)	=D30+C30	=D30-(3.75*(A30-1))	=IF(D30>E26,D30-E26,0)			
31	=A30+1	=MAX(B30+Assumptions!$B$2,D30+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B31,E27)	=D31+C31	=D31-(3.75*(A31-1))	=IF(D31>E27,D31-E27,0)			
32	=A31+1	=MAX(B31+Assumptions!$B$2,D31+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B32,E28)	=D32+C32	=D32-(3.75*(A32-1))	=IF(D32>E28,D32-E28,0)			
33	=A32+1	=MAX(B32+Assumptions!$B$2,D32+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B33,E29)	=D33+C33	=D33-(3.75*(A33-1))	=IF(D33>E29,D33-E29,0)			
34	=A33+1	=MAX(B33+Assumptions!$B$2,D33+Assumptions!$B$2)	=Assumptions!$B$4	=MAX(B34,E30)	=D34+C34	=D34-(3.75*(A34-1))	=IF(D34>E30,D34-E30,0)			

Figure 5: Formulas for the deterministic case - 4 truck example

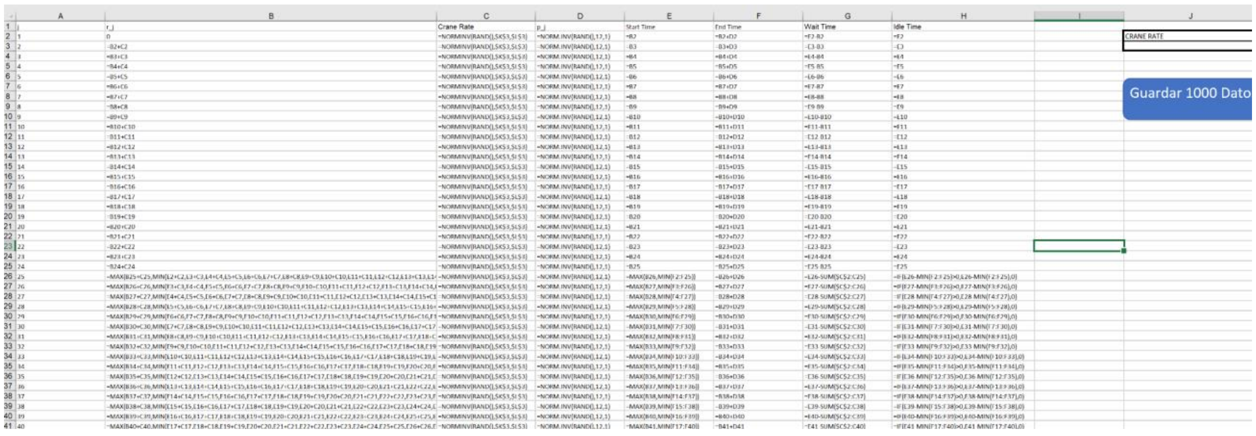
For the stochastic case, jobs that start earlier can theoretically end later. Therefore, the release date depends on the max of the previous release date plus the crane rate, and the earliest ending job of the active jobs. For simplifying assumptions in excel, we assumed that the active jobs would be the number of preceding jobs equivalent to the number of trucks being scheduled in that specific instance. We also created 6 sheets for both the stochastic case of problem 1 and problem 2, respectively. For the problem 2 stochastic formulation, we divided the crane rate by 6 and modeled crane rate as a normal distribution with a relatively small variance around this new, faster mean. Dividing the crane rate by 6 allowed us to model the situation with just one crane. We created a spreadsheet with 275*6 jobs and scheduled them based on the assumptions previously listed, adjusted slightly for a stochastic case. We then used excel macros to simulate 1000 instances of the random problem 2 and calculate average cmax. We later implemented simulations in simpy to do the same thing faster.



A	B	C	D	E	F	G	H
j	r_j	Crane Rate	p_j	Start Time	End Time	Wait Time	Idle Time
1	0	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=B2	=E2+D2	=E2-B2	=E2
2	=B2+C2	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=B3	=E3+D3	=E3-B3	=E3
3	=B3+C3	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=B4	=E4+D4	=E4-B4	=E4
4	=B4+C4	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=B5	=E5+D5	=E5-B5	=E5
5	=MAX(B5+C5,MIN(E2+C2,E3+C3,E4+C4,E5+C5))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B6,MIN(F2,F3,F4,F5))	=E6+D6	=E6-SUM($C$2:C5)	=IF(E6-MIN(F2:F5)>0,E6-MIN(F2:F5),0)
6	=MAX(B6+C6,MIN(E3+C3,E4+C4,E5+C5,E6+C6))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B7,MIN(F3,F4,F5,F6))	=E7+D7	=E7-SUM($C$2:C6)	=IF(E7-MIN(F3:F6)>0,E7-MIN(F3:F6),0)
7	=MAX(B7+C7,MIN(E4+C4,E5+C5,E6+C6,E7+C7))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B8,MIN(F4,F5,F6,F7))	=E8+D8	=E8-SUM($C$2:C7)	=IF(E8-MIN(F4:F7)>0,E8-MIN(F4:F7),0)
8	=MAX(B8+C8,MIN(E5+C5,E6+C6,E7+C7,E8+C8))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B9,MIN(F5,F6,F7,F8))	=E9+D9	=E9-SUM($C$2:C8)	=IF(E9-MIN(F5:F8)>0,E9-MIN(F5:F8),0)
9	=MAX(B9+C9,MIN(E6+C6,E7+C7,E8+C8,E9+C9))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B10,MIN(F6,F7,F8,F9))	=E10+D10	=E10-SUM($C$2:C9)	=IF(E10-MIN(F6:F9)>0,E10-MIN(F6:F9),0)
10	=MAX(B10+C10,MIN(E7+C7,E8+C8,E9+C9,E10+C10))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B11,MIN(F7,F8,F9,F10))	=E11+D11	=E11-SUM($C$2:C10)	=IF(E11-MIN(F7:F10)>0,E11-MIN(F7:F10),0)
11	=MAX(B11+C11,MIN(E8+C8,E9+C9,E10+C10,E11+C11))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B12,MIN(F8,F9,F10,F11))	=E12+D12	=E12-SUM($C$2:C11)	=IF(E12-MIN(F8:F11)>0,E12-MIN(F8:F11),0)
12	=MAX(B12+C12,MIN(E9+C9,E10+C10,E11+C11,E12+C12))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B13,MIN(F9,F10,F11,F12))	=E13+D13	=E13-SUM($C$2:C12)	=IF(E13-MIN(F9:F12)>0,E13-MIN(F9:F12),0)
13	=MAX(B13+C13,MIN(E10+C10,E11+C11,E12+C12,E13+C13))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B14,MIN(F10,F11,F12,F13))	=E14+D14	=E14-SUM($C$2:C13)	=IF(E14-MIN(F10:F13)>0,E14-MIN(F10:F13),0)
14	=MAX(B14+C14,MIN(E11+C11,E12+C12,E13+C13,E14+C14))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B15,MIN(F11,F12,F13,F14))	=E15+D15	=E15-SUM($C$2:C14)	=IF(E15-MIN(F11:F14)>0,E15-MIN(F11:F14),0)
15	=MAX(B15+C15,MIN(E12+C12,E13+C13,E14+C14,E15+C15))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B16,MIN(F12,F13,F14,F15))	=E16+D16	=E16-SUM($C$2:C15)	=IF(E16-MIN(F12:F15)>0,E16-MIN(F12:F15),0)
16	=MAX(B16+C16,MIN(E13+C13,E14+C14,E15+C15,E16+C16))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B17,MIN(F13,F14,F15,F16))	=E17+D17	=E17-SUM($C$2:C16)	=IF(E17-MIN(F13:F16)>0,E17-MIN(F13:F16),0)
17	=MAX(B17+C17,MIN(E14+C14,E15+C15,E16+C16,E17+C17))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B18,MIN(F14,F15,F16,F17))	=E18+D18	=E18-SUM($C$2:C17)	=IF(E18-MIN(F14:F17)>0,E18-MIN(F14:F17),0)
18	=MAX(B18+C18,MIN(E15+C15,E16+C16,E17+C17,E18+C18))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B19,MIN(F15,F16,F17,F18))	=E19+D19	=E19-SUM($C$2:C18)	=IF(E19-MIN(F15:F18)>0,E19-MIN(F15:F18),0)
19	=MAX(B19+C19,MIN(E16+C16,E17+C17,E18+C18,E19+C19))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B20,MIN(F16,F17,F18,F19))	=E20+D20	=E20-SUM($C$2:C19)	=IF(E20-MIN(F16:F19)>0,E20-MIN(F16:F19),0)
20	=MAX(B20+C20,MIN(E17+C17,E18+C18,E19+C19,E20+C20))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B21,MIN(F17,F18,F19,F20))	=E21+D21	=E21-SUM($C$2:C20)	=IF(E21-MIN(F17:F20)>0,E21-MIN(F17:F20),0)
21	=MAX(B21+C21,MIN(E18+C18,E19+C19,E20+C20,E21+C21))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B22,MIN(F18,F19,F20,F21))	=E22+D22	=E22-SUM($C$2:C21)	=IF(E22-MIN(F18:F21)>0,E22-MIN(F18:F21),0)
22	=MAX(B22+C22,MIN(E19+C19,E20+C20,E21+C21,E22+C22))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B23,MIN(F19,F20,F21,F22))	=E23+D23	=E23-SUM($C$2:C22)	=IF(E23-MIN(F19:F22)>0,E23-MIN(F19:F22),0)
23	=MAX(B23+C23,MIN(E20+C20,E21+C21,E22+C22,E23+C23))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B24,MIN(F20,F21,F22,F23))	=E24+D24	=E24-SUM($C$2:C23)	=IF(E24-MIN(F20:F23)>0,E24-MIN(F20:F23),0)
24	=MAX(B24+C24,MIN(E21+C21,E22+C22,E23+C23,E24+C24))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B25,MIN(F21,F22,F23,F24))	=E25+D25	=E25-SUM($C$2:C24)	=IF(E25-MIN(F21:F24)>0,E25-MIN(F21:F24),0)
25	=MAX(B25+C25,MIN(E22+C22,E23+C23,E24+C24,E25+C25))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B26,MIN(F22,F23,F24,F25))	=E26+D26	=E26-SUM($C$2:C25)	=IF(E26-MIN(F22:F25)>0,E26-MIN(F22:F25),0)
26	=MAX(B26+C26,MIN(E23+C23,E24+C24,E25+C25,E26+C26))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B27,MIN(F23,F24,F25,F26))	=E27+D27	=E27-SUM($C$2:C26)	=IF(E27-MIN(F23:F26)>0,E27-MIN(F23:F26),0)
27	=MAX(B27+C27,MIN(E24+C24,E25+C25,E26+C26,E27+C27))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B28,MIN(F24,F25,F26,F27))	=E28+D28	=E28-SUM($C$2:C27)	=IF(E28-MIN(F24:F27)>0,E28-MIN(F24:F27),0)
28	=MAX(B28+C28,MIN(E25+C25,E26+C26,E27+C27,E28+C28))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B29,MIN(F25,F26,F27,F28))	=E29+D29	=E29-SUM($C$2:C28)	=IF(E29-MIN(F25:F28)>0,E29-MIN(F25:F28),0)
29	=MAX(B29+C29,MIN(E26+C26,E27+C27,E28+C28,E29+C29))	=NORMINV(RAND(),3.75,0.3)	=NORM.INV(RAND(),12,1)	=MAX(B30,MIN(F26,F27,F28,F29))	=E30+D30	=E30-SUM($C$2:C29)	=IF(E30-MIN(F26:F29)>0,E30-MIN(F26:F29),0)

Figure 6: Formulas for random problem 1 - 4 truck example

Formulas for random problem 2 (example is for 24 trucks):



	A	B	C	D	E	F	G	H	I	J
1	j	r_j	Crane Rate	p_j	Start Time	End Time	Wait Time	Idle Time		
2	1	0	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B2	=B2+D2	=E2-B2	=E2		CRANE RATE
3	2	=B2+C2	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B3	=B3+D3	=E3-B3	=E3		
4	3	=B3+C3	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B4	=B4+D4	=E4-B4	=E4		
5	4	=B4+C4	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B5	=B5+D5	=E5-B5	=E5		
6	5	=B5+C5	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B6	=B6+D6	=E6-B6	=E6		
7	6	=B6+C6	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B7	=B7+D7	=E7-B7	=E7		
8	7	=B7+C7	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B8	=B8+D8	=E8-B8	=E8		
9	8	=B8+C8	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B9	=B9+D9	=E9-B9	=E9		
10	9	=B9+C9	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B10	=B10+D10	=E10-B10	=E10		
11	10	=B10+C10	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B11	=B11+D11	=E11-B11	=E11		
12	11	=B11+C11	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B12	=B12+D12	=E12-B12	=E12		
13	12	=B12+C12	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B13	=B13+D13	=E13-B13	=E13		
14	13	=B13+C13	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B14	=B14+D14	=E14-B14	=E14		
15	14	=B14+C14	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B15	=B15+D15	=E15-B15	=E15		
16	15	=B15+C15	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B16	=B16+D16	=E16-B16	=E16		
17	16	=B16+C16	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B17	=B17+D17	=E17-B17	=E17		
18	17	=B17+C17	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B18	=B18+D18	=E18-B18	=E18		
19	18	=B18+C18	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B19	=B19+D19	=E19-B19	=E19		
20	19	=B19+C19	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B20	=B20+D20	=E20-B20	=E20		
21	20	=B20+C20	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B21	=B21+D21	=E21-B21	=E21		
22	21	=B21+C21	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B22	=B22+D22	=E22-B22	=E22		
23	22	=B22+C22	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B23	=B23+D23	=E23-B23	=E23		
24	23	=B23+C23	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B24	=B24+D24	=E24-B24	=E24		
25	24	=B24+C24	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=B25	=B25+D25	=E25-B25	=E25		
26	25	=MAX(B25+C25,MIN(E2+C2,E3+C3,E4+C4,E5+C5,E6+C6,E7+C7,E8+C8,E9+C9,E10+C10,E11+C11,E12+C12,E13+C13,E1	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B26,MIN(F2:F25))	=B26+D26	=E26-SUM($C$2:C25)	=IF(E26-MIN(F2:F25)>0,E26-MIN(F2:F25),0)		
27	26	=MAX(B26+C26,MIN(E3+C3,E4+C4,E5+C5,E6+C6,E7+C7,E8+C8,E9+C9,E10+C10,E11+C11,E12+C12,E13+C13,E14+C14,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B27,MIN(F3:F26))	=B27+D27	=E27-SUM($C$2:C26)	=IF(E27-MIN(F3:F26)>0,E27-MIN(F3:F26),0)		
28	27	=MAX(B27+C27,MIN(E4+C4,E5+C5,E6+C6,E7+C7,E8+C8,E9+C9,E10+C10,E11+C11,E12+C12,E13+C13,E14+C14,E15+C1	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B28,MIN(F4:F27))	=B28+D28	=E28-SUM($C$2:C27)	=IF(E28-MIN(F4:F27)>0,E28-MIN(F4:F27),0)		
29	28	=MAX(B28+C28,MIN(E5+C5,E6+C6,E7+C7,E8+C8,E9+C9,E10+C10,E11+C11,E12+C12,E13+C13,E14+C14,E15+C15,E16+	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B29,MIN(F5:F28))	=B29+D29	=E29-SUM($C$2:C28)	=IF(E29-MIN(F5:F28)>0,E29-MIN(F5:F28),0)		
30	29	=MAX(B29+C29,MIN(E6+C6,E7+C7,E8+C8,E9+C9,E10+C10,E11+C11,E12+C12,E13+C13,E14+C14,E15+C15,E16+C16,E1	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B30,MIN(F6:F29))	=B30+D30	=E30-SUM($C$2:C29)	=IF(E30-MIN(F6:F29)>0,E30-MIN(F6:F29),0)		
31	30	=MAX(B30+C30,MIN(E7+C7,E8+C8,E9+C9,E10+C10,E11+C11,E12+C12,E13+C13,E14+C14,E15+C15,E16+C16,E17+C17	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B31,MIN(F7:F30))	=B31+D31	=E31-SUM($C$2:C30)	=IF(E31-MIN(F7:F30)>0,E31-MIN(F7:F30),0)		
32	31	=MAX(B31+C31,MIN(E8+C8,E9+C9,E10+C10,E11+C11,E12+C12,E13+C13,E14+C14,E15+C15,E16+C16,E17+C17,E18+C	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B32,MIN(F8:F31))	=B32+D32	=E32-SUM($C$2:C31)	=IF(E32-MIN(F8:F31)>0,E32-MIN(F8:F31),0)		
33	32	=MAX(B32+C32,MIN(E9+C9,E10+C10,E11+C11,E12+C12,E13+C13,E14+C14,E15+C15,E16+C16,E17+C17,E18+C18,E19	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B33,MIN(F9:F32))	=B33+D33	=E33-SUM($C$2:C32)	=IF(E33-MIN(F9:F32)>0,E33-MIN(F9:F32),0)		
34	33	=MAX(B33+C33,MIN(E10+C10,E11+C11,E12+C12,E13+C13,E14+C14,E15+C15,E16+C16,E17+C17,E18+C18,E19+C19,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B34,MIN(F10:F33))	=B34+D34	=E34-SUM($C$2:C33)	=IF(E34-MIN(F10:F33)>0,E34-MIN(F10:F33),0)		
35	34	=MAX(B34+C34,MIN(E11+C11,E12+C12,E13+C13,E14+C14,E15+C15,E16+C16,E17+C17,E18+C18,E19+C19,E20+C20,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B35,MIN(F11:F34))	=B35+D35	=E35-SUM($C$2:C34)	=IF(E35-MIN(F11:F34)>0,E35-MIN(F11:F34),0)		
36	35	=MAX(B35+C35,MIN(E12+C12,E13+C13,E14+C14,E15+C15,E16+C16,E17+C17,E18+C18,E19+C19,E20+C20,E21+C21,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B36,MIN(F12:F35))	=B36+D36	=E36-SUM($C$2:C35)	=IF(E36-MIN(F12:F35)>0,E36-MIN(F12:F35),0)		
37	36	=MAX(B36+C36,MIN(E13+C13,E14+C14,E15+C15,E16+C16,E17+C17,E18+C18,E19+C19,E20+C20,E21+C21,E22+C22,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B37,MIN(F13:F36))	=B37+D37	=E37-SUM($C$2:C36)	=IF(E37-MIN(F13:F36)>0,E37-MIN(F13:F36),0)		
38	37	=MAX(B37+C37,MIN(E14+C14,E15+C15,E16+C16,E17+C17,E18+C18,E19+C19,E20+C20,E21+C21,E22+C22,E23+C23,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B38,MIN(F14:F37))	=B38+D38	=E38-SUM($C$2:C37)	=IF(E38-MIN(F14:F37)>0,E38-MIN(F14:F37),0)		
39	38	=MAX(B38+C38,MIN(E15+C15,E16+C16,E17+C17,E18+C18,E19+C19,E20+C20,E21+C21,E22+C22,E23+C23,E24+C24,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B39,MIN(F15:F38))	=B39+D39	=E39-SUM($C$2:C38)	=IF(E39-MIN(F15:F38)>0,E39-MIN(F15:F38),0)		
40	39	=MAX(B39+C39,MIN(E16+C16,E17+C17,E18+C18,E19+C19,E20+C20,E21+C21,E22+C22,E23+C23,E24+C24,E25+C25,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B40,MIN(F16:F39))	=B40+D40	=E40-SUM($C$2:C39)	=IF(E40-MIN(F16:F39)>0,E40-MIN(F16:F39),0)		
41	40	=MAX(B40+C40,MIN(E17+C17,E18+C18,E19+C19,E20+C20,E21+C21,E22+C22,E23+C23,E24+C24,E25+C25,E26+C26,E	=NORMINV(RAND(),$K$3,$L$3)	=NORM.INV(RAND(),12,1)	=MAX(B41,MIN(F17:F40))	=B41+D41	=E41-SUM($C$2:C40)	=IF(E41-MIN(F17:F40)>0,E41-MIN(F17:F40),0)		

Figure 7: Formulas for random problem 2 - 24 truck example

Simpy

Once we decided to simulate the situation, we used simpy on python to recreate the situation. Simpy would let us easily create a process that is "processing" for a specified time before repeating. The difficulty came when we had to limit the cran processes depending on the amount of available trucks. If there were no trucks available, even if a crane was ready, the simulation had to wait until a truck became available before continuing to the next job. We first modeled the discrete case with one crane and 4 trucks. The numbers could be compared with our previous excel results verifying that our simulation was simulating the correct situation. Our first approach was to create a crane process for each truck. This crane would not start until the previous process was finished. Four virtual cranes were needed to simulate one real crane. This approach was not successful. There was a problem with starting the code, as a crane would not start until the previous was finished, a necessary constraint except at the start. We realized that this approach was very limited, as it could only simulate this specific one crane 4 trucks scenario, and would be difficult to generalize.

Our second approach made better use of the simpy environment. The trucks could be considered a "limited resource" which simpy knows how to handle. We then created a crane function. This function would wait 3.75 seconds, before "releasing" a new job. After releasing a job, the crane requests a truck. Here simpy could handle the case where no trucks were available by making the crane function wait until a truck becomes available before continuing to release another job. Once a truck becomes available, the crane calls a truck function to process the container. After processing, the truck is released and this job completes. Using simpy, we turn the crane function into a process to run.

This environment allows us to easily modify it to fit a specific scenario. If we wanted to simulate more trucks we just increase the amount of limited resources. If we wanted to simulate more cranes we can increase the amount of crane processes. We also modified the release date of the containers and the processing time of trucks to be stochastic, and more accurately model the real world.

Results

Problem 1 Results

The first problem involves equally sized mini fleets of trucks per crane from the total pool of 24 trucks. For the deterministic case, we found that the minimum Cmax of 1039.5 minutes was achieved with 4 trucks. Any additional trucks did not improve the Cmax. Instead, other measurements such as wait time and idle time increased.

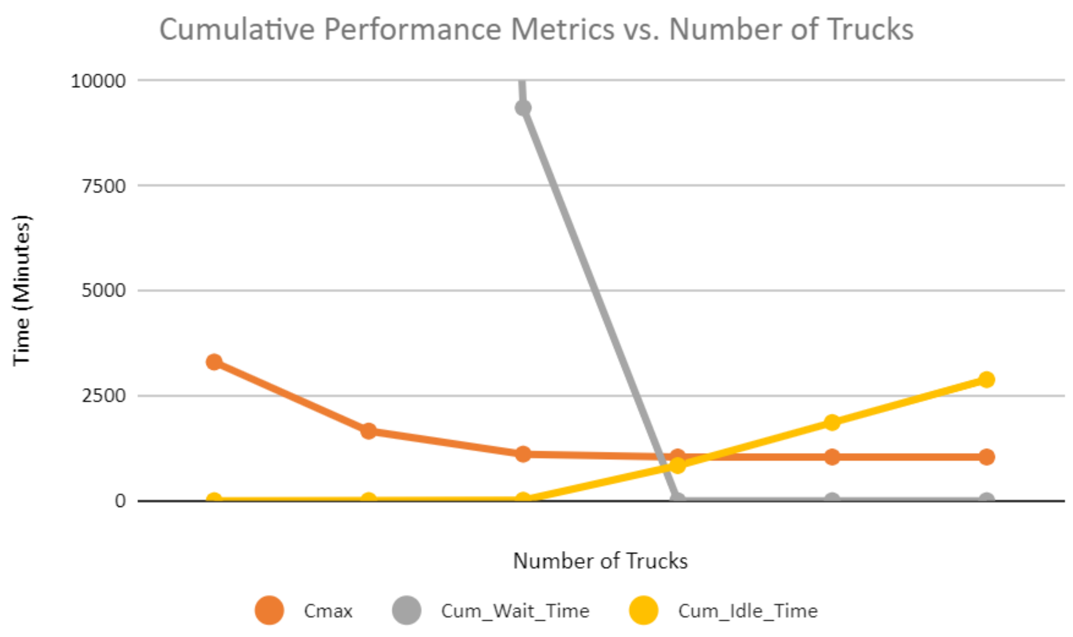


Figure 8: Problem 1 Cumulative Performance Metrics - for 1-to-6 truck fleets

The above graph illustrates the Cmax, cumulative wait time, and cumulative idle time for the different number of trucks. A 3-truck subfleet is worth exploring because there is only a small increase in Cmax, but a large drop in idle time. The large increase in wait time is less consequential in this model because the cmax is still close to optimal. That being said, we believe that 4 trucks is the best solution because cranes cannot work on other ships while one ship is in the port. Therefore, it is best to process an entire ship as quickly as possible to accommodate a new ship. This is because the costs of operating cranes are probably more expensive than operating trucks. However, in the case where operating more trucks is more expensive than a ship being at the dock for a slightly suboptimal cmax, a 3 truck formulation could be better.

Problem 2 Results

The second problem involves the pool of trucks with 6 cranes. We used simpy to simulate the results of implementing this scenario. We looked at makespan as well as wait time. The results are as follows.

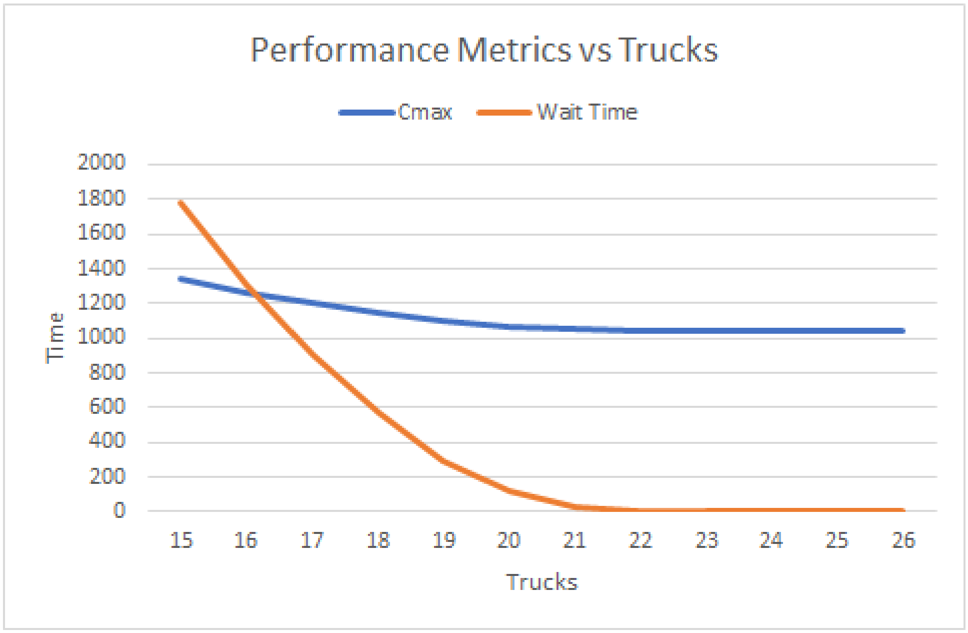


Figure 9: Problem 2 Results - Stochastic Example

When we compared the performance of the stochastic scenario with the deterministic scenario, there was no significant difference. At 24 trucks, the wait time had converged to 0, and the C_{max} had converged to 1400. We then modified the number of trucks and ran the simulation multiple times, creating the above graph. As the number of trucks increases, waiting time of the cranes decreases, as expected. This waiting time reaches zero at 22 trucks. The C_{max} converged to the optimal 1400 at 21 trucks. This is because adding more trucks will not significantly decrease the waiting time of the cranes. The cranes processing times are the limiting factor in the operation. Decreasing the number of trucks to 22 would decrease costs without affecting the operation time.

Conclusions

In conclusion, port operations is a very complex system, made up of various problems, restrictions and resources to make it work. This complexity opens up many instances to improve and optimize the whole system. In that sense, our project has the potential of providing enough evidence for DP World Lirquén to change its current model of assigning trucks not to reduce maximum makespan as we were expecting at the beginning, but as a way of reducing the total amount of trucks needed in order to achieve a similar outcome.

The first interesting observation is that both the deterministic and stochastic version of both problems yield similar decisions regarding the assignment of trucks in the port. In this matter there could be space for improvement by making on-field measurements in order to accommodate the correct probability distribution for both the crane rates and the processing times of the trucks.

From the results of problem 1 we can confirm the intuition that while you increase the number of trucks that take part in the “team” of each crane, the maximum makespan of the particular crane decreases. The makespan can only go as low as the rate of the crane allows it and that number is with 4 trucks. Any additional trucks increases the idle time of the trucks without reducing Cmax.

Since the release dates of the containers are subject to the one before being picked up, the optimal schedule is assigning trucks to containers by their release dates. This process guarantees that the idle time of the cranes will be minimized.

We can also observe how problem 1 with 4 trucks and problem 2 with 24 trucks yield a very similar makespan, but problem 2 allows us to use fewer trucks (<24) maintaining a similar makespan, opening a future analysis on cost reduction.

Finally, other costs should be taken into account. Real time truck assignment would require a dispatcher (increased labor costs) or new technologies to dynamically assign trucks (increased equipment cost). These costs were not modeled in our formulation. Furthermore, tradeoffs between idle time of wait time of cranes/containers, vs. wait time of trucks should be considered. These idle/wait times have different labor costs associated with them.