# Hazira Port Analysis: Report

Lucy Malmud

Table of Contents

[Hazira Port Analysis: Report 1](#_Toc199430710)

[Executive Summary 1](#_Toc199430711)

[Methodology 2](#_Toc199430712)

[Simulation Engine & Assumptions 2](#_Toc199430713)

[Cost Model & Assumptions 4](#_Toc199430714)

[Scenario Model & Assumptions 5](#_Toc199430715)

[Results 6](#_Toc199430716)

[Simulation Results 6](#_Toc199430717)

[Cost-Model Results 7](#_Toc199430718)

[Scenario Analysis 8](#_Toc199430719)

[Appendix 9](#_Toc199430720)

# Executive Summary

Background

Objective of Study

Scope and Limitations

# Methodology

## Simulation Engine & Assumptions

All simulations were written in Python from the ground up. Below, we will outline the process and assumptions for each simulation. Any note marked “assumed\*” was not given in the task sheet, and was created in order to specify how the simulation will be executed.

S2: simulate\_berth\_hazira.py 🡪 berth\_occupancy\_hazira.csv

* For each berth (MP1-MP4, CT1-2) generates daily occupancy data in decimal format that is normally distributed with given mean .78 and (standard deviation assumed\*) standard deviation .05
* Mean is increased by .09 for winter (December-February) and decreased by .14 for monsoon dip (July-September)
* The results from this pipeline were not used in any other simulation

S3: simulate\_vessels\_hazira.py 🡪 vessel\_turnaround.csv

* Given that arrivals are distributed Poisson(1200 arrivals/year = 1200/(356\*24) arrivals/hour), we generate the time *between* arrivals from Exponential(365\*24/1200)
* As a vessel arrives, it is docked at the berth that has the next earliest idle time (assignment strategy \*assumed)
* The (given) time it takes for a vessel to be processed once assigned to a berth is N(mean = 23hours, standard deviation = 4.5 hours)
  + There is a (given) 11% change that a vessel experiences an service time, in which case an amount of hours uniform(.5, 3) (distribution assumed\*) are added to the service time

S4: simulate\_containers\_hazira.py 🡪 container\_moves.py

* Reads input from vessel\_turnaround\_hazira.csv and generates one container per vessel (assumed\*)
  + The number of times a container is moved is Poisson(2.6) (given)
  + The number of TEU (twenty-foot equivalent units) is N(mean = 1400, standard deviation = 150) with a maximum capacity of 1500 TEU (given mean and maximum, standard deviation and interpretation of 1500 assumed\*)
* Each container is moved using a Yard (also known as RTG) Crane or a Quay Crane. As per Hazira Yard’s Layout, there are 6 Quay Cranes and 14 Yard Cranes. (p. 9 Appendix #1)
  + Processing time for a quay crane is N(mean = 90 seconds, standard deviation = 10 seconds) minimum 20 seconds (assumed\* with source Appendix #2)
  + Processing time for a yard crane is N(mean = 144 seconds, standard deviation.= 15 seconds) minimum 30 seconds (assumed\* with source Appendix #2)
  + Cranes are identified “Quay” or “Yard” followed by a number, starting at 0
* Each container generates a number of moves that do not need to be processed in succession. Containers are assigned to the resource that is available the soonest (assignment strategy assumed\*)

S5: simulate\_cranes\_hazira.py 🡪 crane\_uptime\_hazira.csv

* The simulation runs for one year and marks each time that a crane is down
* A quay crane is expected to be up for 19 hours a day, where downtimes are each 1.2 hours (given)
* A yard crane is expected to be up for 15 hours a day, where downtimes are each 1 hour (given)
* Time between arrivals is Weibull(k = 1.7), which is then scaled to have an expected value of 2 (assumed\* interpretation of the given parameters)
  + The scale, lambda, is equal to 12/gamma(1 + 1/1.7)

S6: simulate\_gate\_hazira.py 🡪 gate\_entries\_hazira.csv

* Simulation runs for one year, but will generate truck arrivals according to a parameter HOUR\_TIMESTEP, since trucks cannot arrive continuously
* Truck arrival is Poisson(160 trucks/day) or equivalently Poisson(160/24\*HOUR\_TIMESTEP trucks/ 1/HOUR\_TIMESTEP hours)
  + The rate is scaled by 1.28 if it is between hours 8-10 or 17-19 (given) to represent the peak hour surge
* There is only one gate to service all trucks (assumed\*)
  + Trucks enter a queue, as soon as one truck is processed, the next one waiting may be processed
* The time it takes for a truck to be serviced is N(mean = 11 minutes, standard deviation = 2.5 minutes)
* It is noted that the queue length becomes very long as the simulation progresses – the assumption that there is only one gate to service all trucks is likely incorrect for these rates.

S7: simulate\_energy\_hazira.py 🡪 energy\_consumption\_hazira.py

* Produces entirely deterministic hourly data for an entire year (energy draw is the same, it does not come from any distribution assumed\*)
* Base hourly energy draw is 6500, which is always increased by 6% for admin costs (given)
  + Scaled by 1.17 during June, July and August (summer) and .83 during December, January, and February (winter) (given)
  + Further scaled by 1.27 if during daily hours of 8-18 for peak

S8: simulate\_maintenance\_hazira.py 🡪 maintenance\_events\_hazira.csv

* Each crane (quay and yard) experiences weekly maintenance which lasts 3.5 hours (given) and these events are randomly inserted on a random day every week (assumed\* that each event is randomly scheduled each week)
* All other resources experience maintenance three times a month for 4.5 hours each (and such events are also randomly distributed throughout the month assumed\*)
* Conveyors and lights are treated as one pool resource (no documentation found fore the number present at the port) and this may be easily changed by changing global simulation parameters NUM\_CONVEY, NUM\_LIGHT (assumed\*)

## Cost Model & Assumptions

In order to calculate costs, we had to make some assumptions on how to convert the metrics produced in the simulation to values that may be multiplied by costs. All calculations will conducted in lakhs (1 rupee = 100,000 lakh = 10,000,000 crore). The following values were all given.

Unit Rates

|  |  |  |  |
| --- | --- | --- | --- |
| **metric** | **unit\_measure** | **unit\_rate** | **driver** |
| quay\_crane | hour of use | 0.51 | crane\_uptime\_hazira.csv |
| yard\_crane | hour of use | 0.036 | crane\_uptime\_hazira.csv |
| container\_moves | TEU-day in yard | 0.0055 | container\_moves\_hazira.csv |
| truck\_entry | num entries | 0.0038 | gate\_entries\_hazira.csv |
| energy | kWh consumed | 0.000078 | energy\_consumption\_hazira.csv |

Yearly Expenses

We will receive inputs from each of the relevant simulations in order to calculate the relevant costs. First, we will summarize how volume is calculated:

|  |  |
| --- | --- |
| Crane | Need to calculate the total hours of use. Simulation output only records hours of downtime. TOTAL\_DOWNTIME\_HOURS is calculated by summing the duration of downtime for each crane.  VOLUME = NUM\_CRANES \* 24 \* 365 – TOTAL\_DOWNTIME\_HORUS |
| Container Moves | Given unit rate is “TEU-day/yard.” This was interpreted to be the total number of TEU used, which is calculated by summing the TEU that each container holds.  VOLUME = SUM(TEU moved per container) |
| Truck Entry | Note that this was not calculated as the number of trucks that entered the gate, but rather the number of trucks that were processed (assumed\*)  VOLUME = Number of trucks processed |
| Energy | VOLUME = SUM(kWh used each hour) |

## Scenario Model & Assumptions

We were given three sets of parameters that define how the simulations will run, summarized below:

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario Name | Vessel Service Time | Crane Downtime | Gate Speed |
| Conservative | .95 | .96 | 1.03 |
| Moderate | .90 | .92 | 1.06 |
| Aggressive | .80 | .85 | 1.10 |

Note that originally parameters were given in the form “+5% vessel turnaround,” but these values were converted so that they could directly scale to previous simulation outputs. For example, the output of crane productivity was measured in the hours the crane was down, so we wanted to *decrease* the downtime which would increase the productivity.

The process to apply scenarios to vessel service time and crane downtime is straightforward: simply scale each value by the given scenario factor. For gate speed, we similarly increase the number of trucks processed at each hour by the given factor. There are additional sanity checks of ensuring that no more trucks are processed than the number of trucks currently in the queue. Furthermore, we allow for fractional amounts of trucks to be processed (otherwise there may not be a notable different in the simulation from the original). This is a reasonable assumption, as a fractional amount of a truck in a queue simply corresponds to that truck being partway through its service time.

# Results

## Simulation Results

First, we check the correctness of the vessel-to-berth assignment strategy. Here, each horizontal line indicates a time period that a berth is occupied by a vessel.

A graph with lines and numbers

AI-generated content may be incorrect.

We have also constructed a heatmap of the berth occupancy, based on the generated percentages. Note that these values are independent of the occupancy documented in the previous chart (a possible source of error- there is currently no link between the results in berth\_occupancy\_hazira.csv and vessel\_turnaround\_hazira.csv).

A graph showing a number of numbers

AI-generated content may be incorrect.

## Cost-Model Results

Using the provided unit costs and generated yearly simulation data, we have a total cost of 39,770 cr. Note that for the year ended March 2024, the total operating expenses were 31,958 cr (p. 19 Appendix #3).

|  |  |  |  |
| --- | --- | --- | --- |
| **metric** | **volume** | **unit\_cost** | **total\_cost** |
| quay\_crane | 52343.1 | 0.51 | 26694.981 |
| yard\_crane | 122215.458 | 0.036 | 4399.7565 |
| container\_moves | 1543655 | 0.0055 | 8490.1025 |
| truck\_entry | 47753 | 0.0038 | 181.4614 |
| energy | 47753 | 0.000078 | 3.724734 |

We summarize these results in the following chart.

## Scenario Analysis

After running each scenario, we can see percent saved in each subprocess. Note that the amounts saved for truck entries is negative- this is because the speedup in gate processing time will allow for more trucks to be processed, thus incurring additional cost.

Lastly, we perform a sensitivity analysis. We allow for a change in the unit rate and a performance adjustment. In order to implement these changes, we multiply the variable\_cost\_savings by the unit\_rate\_scalar and 1 + improvement\_adj\_pp, as a linear approximation to the simulated costs is sufficient. The results for the conservative scenario are below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sensitivity table** | low input | high input | NPV @ low | NPV @ high |
| unit rate scalar | 0.8 | 1.2 | -667.41286 | 31896.2798 |
| improvement adj pp | -0.05 | 0.05 | 11638.5631 | 19741.6498 |

The dark blue part of the bar indicates the adjustment incurred by a change in unit rates, where the orange part notes the change due to an improvement adjustment. We are not confident in the accuracy of these parts of the calculation, as a negative internal rate of return is also noted in all scenarios.

# Appendix

1. Official Hazira Port handbook: <https://www.adaniports.com/-/media/Project/Ports/PortsAndTerminals/Hazira-port-Documents/Documents/Port-Information-Book.pdf>
2. Information about container port automation: <https://www.itf-oecd.org/sites/default/files/docs/container-port-automation.pdf>
3. Fiscal Year 2024-2025: https://www.adaniports.com/-/media/Project/Ports/Investors/FY24-25/Adani-Hazira-Port-Ltd.pdf