MATH 377: COST ANALYSIS OF A COAL TIPPLE

JAMES DELLA-GIUSTINA, CRAIG RILEY, MITCHELL MACADAMS

1. Introduction

We aim to solve a problem posed by COMAP Mathematical Contest in Modeling in 1993 [2].

1.1. **Problem.** "The Aspen-Boulder Coal Company runs a loading facility consisting of a large coal tipple. When the coal trains arrive, they are loaded from the tipple. The standard coal train takes 3 hours to load, and the tipple's capacity is 1.5 standard trainloads of coal. Each day, the railroad sends three standard trains to the loading facility, and they arrive at any time between 5 A.M. and 8 P.M. local time. Each of the trains has three engines. If a train arrives and sits idle while waiting to be loaded, the railroad charges a special fee, called a demurrage. The fee is $5,000 per engine per hour. In addition, a high-capacity train arrives once a week every Thursday between 11 A.M. and 1 P.M. This special train has five engines and holds twice as much coal as a standard train. An empty tipple can be loaded directly from the mine to its capacity in six hours by a single loading crew. This crew (and its associate equipment) costs $9,000 per hour. A second crew can be called out to increase the loading rate by conducting an additional tipple-loading operation at the cost of $12,000 per hour. Because of safety requirements, during tipple loading, no trains can be loaded. Whenever train loading is interrupted to load the tipple, demurrage charges are in effect."

We are asked to find the annual optimal cost of operation and the following questions are posed:

- How often should the second crew be called out?
- Can this tipple support a fourth standard train every day?
- What are the expected monthly demurrage costs?
- If the standard trains could be scheduled to arrive at precise times, what daily schedule would minimize loading costs?
- Would a third tipple-loading crew at $12,000 per hour reduce annual operations?

We fully admit that concepts and details formulated in a winning paper [7] were considered in our attempt to build a realistic simulation model. However, we also want to stress that certain ideas were intuitive and already present, such as establishing a lower bound and discretizing time into hours. Other assumptions such as interrupting a standard train loading process for a high capacity train were unrealistic in nature and we provide sound reasoning for abandoning such a conclusion below.

1.2. **Analysis.** Let's further break down this problem and make a number of assumptions to simplify our problem.

- Coal tipple holds at most 1.5 train-loads of coal.
- On every day besides Thursday, 3 trains arrive between 5 a.m. and 8 p.m. (15 hours).
- Loading a standard train with coal takes 3 hours.

- Tipple loading can occur before/after our train arrival time window.
- If a train has to wait for the tipple to be loaded, an hourly demurrage rate of $5,000/engine is imposed. Therefore a standard train with 3 engines incurs $15,000/hour in demurrage costs.
- The cost of one crew to load the tipple is $9,000 hour.
  – During tipple loading, no train can be loaded.
  – We assume that 'one' crew is composed of two shifts, each working in 12 hour spells with no time delay between shift changes. Therefore we have 24 hours in which we can load the tipple.
- It takes one crew 6 hours to load the tipple from empty to full (1.5 trainloads).
  – From the information above, one crew can load .25 train-loads of coal into the tipple in 1 hour.
- A second crew can be called in to load the tipple at $12,000 hour.
  – The rate of loading doubles with a second crew, therefore it only takes 1 hour to load .5 train loads of coal into the tipple.
  – This 'on call' second crew charges a higher rate, thus it is safe to assume that we can utilize them at any time we need to and that they are on site.
  – If a third crew at an additional $12,000 were to be called then all three crews together could load the tipple at a rate of .75 trainloads of coal per hour.
- On Thursdays between 11 a.m. and 1 p.m. there is a high-capacity train that has 5 engines and a capacity of 2 train-loads of coal.

Initially we want to make a few assumptions to simplify the problem, starting with discretizing time steps t into hours, with trains arriving exclusively on the hour. Eventually we will want to further reduce this time step of hours into something more realistic, such as minutes, for a more realistic and accurate model. Since we have chosen hours, let's first classify all relevant quantities needed to approach the problem.

Hourly Rates:

- Coal tipple is filled at `# crews * .25 * hours`.
- Coal tipple crew cost at `((# crews * 12,000)-3,000) * hours`.
- Trains are loaded at `.33 train-loads/hours`.
- Standard train demurrage rate at `$15,000 * # trains * hours`.
- High-capacity train demurrage rate at `$25,000 * hours`.

One assumption that we will not alter is that trains are loaded in the order that they arrive, a first in-first out approach. This is typical of a single tipple coal loading plant, which will presumably have only one track under the tipple for loading (otherwise we might have the ability to load multiple trains at a time). Empty freight trains can typically weigh around 11 million pounds [6] and a fully loaded coal train can weigh around 42 million pounds [5]. Additionally, the average length of a coal train is approximately 1.5 miles [1]. In light of these average lengths/weights of a typical coal train and ignoring the slow acceleration of a train from a stationary position, we are confident in this first in first out methodology.

1.3. **Bounds.** We first consider every day except Thursday, and so we only expect three standard trains to arrive in our 15 hour time window. Let's first create two schedules of best (minimized costs) and worst (maximized costs) case scenarios that we may encounter. This first scenario answers the question **If the standard trains could be scheduled to arrive at precise times, what daily schedule would minimize loading costs?** We can construct a time table as seen in Table 1 which shows the operation costs of an optimal day corresponding to three standard trains. Note that we do not have to ever call a second crew and that at no point do we incur any demurrage costs during the day since there is never any train waiting. This proposed optimal schedule results in a lower bound of $108,000 in crew costs, meaning we will always accrue *at least* this much in daily operations.

Schedule to Minimize Costs

Time	Tipple Capacity	Train Serviced	Loading Crew Cost $	Demurrage Cost $
5 AM	1.5	Train 1	0	0
6 AM	1.17	Train 1	0	0
7 AM	.84	Train 1	0	0
8 AM	.5		$ 9,000	0
9 AM	.75		$ 9,000	0
10 AM	1	Train 2	0	0
11 AM	.67	Train 2	0	0
12 PM	.33	Train 2	0	0
1 PM	0		$ 9,000	0
2 PM	.25		$ 9,000	0
3 PM	.5		$ 9,000	0
4 PM	.75		$ 9,000	0
5 PM	1	Train 3	0	0
6 PM	.67	Train 3	0	0
7 PM	.33	Train 3	0	0
8 PM	0		$ 9,000	0
9 PM	.25		$ 9,000	0
10 PM	.5		$ 9,000	0
11 PM	.75		$ 9,000	0
12 AM	1		$ 9,000	0
1 AM	1.25		$ 9,000	0
2 AM	1.5		0	0

Table 1. Coal train schedule to minimize operational costs during any day with three standard coal trains.

Now let's try to construct a similar time schedule (Table 3) to simulate a worst case scenario and thus an upper bound on operational costs. In this scenario we assume that all three trains arrive at 5 a.m and we have only one crew working to fill the tipple, incurring operational costs of $363,000. Therefore for any given day (excluding Thursday), we can expect $cost \in [\$108,000, \$363,000]$.

For the sake of completeness we have completed an optimal schedule for any given Thursday in Table 2 which gives a lower bound of $283,000 operational costs. However, we'd like to note that there is a significant amount of work that is running over from Thursday into Friday morning, and therefore it is bad form for us to consider this as a 'daily' cost. It should be abundantly clear that there is no scenario for any given Thursday in which we do not incur demurrage costs or tipple loading runs over into Friday morning.

Schedule to Minimize Costs on Thursday

Time	Tipple Capacity	Train Serviced	Loading Crew Cost $	Demurrage Cost $
5 AM	1.5	Train 1	0	0
6 AM	1.17	Train 1	0	0
7 AM	.84	Train 1	0	0
8 AM	.5		$9,000	0
9 AM	.75		$9,000	0
10 AM	1		$21,000	0
11 AM	1.5	HC Train	0	0
12 PM	1.17	HC Train	0	0
1 PM	.84	HC Train	0	0
2 PM	.5	HC Train	0	0
3 PM	.22	HC Train	0	0
4 PM	0		$ 21,000	$25,000
5 PM	.5	HC Train	0	0
6 PM	0		$21,000	0
7 PM	1	Train 2	$21,000	0
8 PM	.66	Train 2	0	$15,000
9 PM	.33	Train 2	0	$15,000
10 PM	0		$ 21,000	$15,000
11 PM	.5		$ 21,000	$15,000
12 AM	1	Train 3	0	0
1 AM	.66	Train 3	0	0
2 AM	.33	Train 3	0	0
3 AM	0		$9,000	0
4 AM	.33		$9,000	0
5 AM	.66		$9,000	0
6 AM	1		$9,000	0
7 AM	1.25		$9,000	0
8 AM	1.5		$9,000	0

Table 2. Coal train schedule to minimize operational costs during any Thursday with three standard coal trains and one high capacity train.

1.4. **Second & Third Crew.** Now let's turn our attention to another question posed: **How often should the second crew be called out?** Specifically this is the case when the demurrage costs exceed the combined costs of both crews at $21,000/hr. Let's break this down into each case:

- 2 standard trains waiting incurring a demurrage rate of $30,000/hr.
- 3 standard trains waiting incurring a demurrage rate of $45,000/hr.
- The high-capacity train is waiting at $25,000/hr.
  - 1 standard train in addition to the high capacity train at $40,000/hr.
  - 2 standard trains in addition to the high capacity train at $55,000/hr.

From the information above we can ascertain that a second crew needs to be called whenever we have one or more trains incurring demurrage costs.

An example will help illustrate this point; say we have a current tipple level of .5 train loads and a single train arrives. If we let only one crew load the tipple then we need 2 hours until the tipple can load the train, resulting in crew costs of \$18,000 and demurrage costs of \$30,000 for a total cost of \$48,000. Instead assume that as soon as the train arrives we allocate two crews to load the tipple in which we will only need 1 hour. Then crew costs would be \$21,000 but demurrage costs would be \$15,000 for a total cost of \$36,000, a 25% decrease.

Now let's move one and try to answer the question **Would a third tipple-loading crew at \$12,000 per hour reduce annual operations?** Let's use the same example from above to help illustrate such a choice but since three crews load the tipple at a rate of .75 train loads an hour, it is necessary to transition to speaking in terms of minutes. We have a total cost of 2 crews above for \$36,000 but 3 crews would load the tipple in 40 minutes. Therefore the total cost would be:

$$\frac{\$33,000/\text{hr} \cdot 40\text{min}}{60\text{min/hr}} + \frac{\$15,000/\text{hr} \cdot 40\text{min}}{60\text{min/hr}}$$

$$= \$22,000 + \$10,000 = \$32,000$$

With a total cost of \$32,000 for 3 crews loading the tipple. Since 1 is the minimum number of trains we could be waiting for, we are safe to assume that 3 crews will be optimal in reducing costs whenever we have *at least* 1 train accruing demurrage costs.

Schedule to Maximize Costs

Time	Tipple Capacity	Train Serviced	Loading Crew Cost $	Demurrage Cost $
5 AM	1.5	Train 1	0	$30,000
6 AM	1.17	Train 1	0	$30,000
7 AM	.84	Train 1	0	$30,000
8 AM	.5		$ 9,000	$30,000
9 AM	.75		$ 9,000	$30,000
10 AM	1	Train 2	0	$15,000
11 AM	.67	Train 2	0	$15,000
12 PM	.33	Train 2	0	$15,000
1 PM	0		$ 9,000	$15,000
2 PM	.25		$ 9,000	$15,000
3 PM	.5		$ 9,000	$15,000
4 PM	.75		$ 9,000	$15,000
5 PM	1	Train 3	0	0
6 PM	.67	Train 3	0	0
7 PM	.33	Train 3	0	0
8 PM	0		$ 9,000	0
9 PM	.25		$ 9,000	0
10 PM	.5		$ 9,000	0
11 PM	.75		$ 9,000	0
12 AM	1		$ 9,000	0
1 AM	1.25		$ 9,000	0
2 AM	1.5		0	0

Table 3. Coal train schedule to maximize operational costs during any day with three standard coal trains and one tipple loading crew.

2. Our Model

2.1. **Overview.** To begin we randomized a train arrival schedule for an entire week for three trains (4 with a high capacity train on Thursday) over a 15-hour and 2-hour time period with 2 crews maximum to serve as the base case scenario. We then repeated the process for 3 crews and 3 daily trains, 2 crews and 4 daily trains, and 3 crews and 4 daily trains. We then ran each of these schedules in our model for 6000 iterations (each consisting of a month) until we were getting monthly/weekly averages of total costs, demurrage costs, and crew costs within our bounds. Once we were confident that our model was yielding reasonable data, we extended our time steps t from hours into minutes to get a more accurate representation of the aforementioned average costs.

Satisfied with the results of setting our time steps t from hours to minutes, we sought to relax some assumptions in the problem that were unrealistic in nature. Specifically, these were the deterministic loading times of both the trains and tipple for which we cannot hope for some constant rate. Therefore for each process we multiplied by a random decimal value between .85 and 1.05 to reflect that variable nature of the time to complete each task. For the case of multiple crews, we wanted to reflect the fact that though the additional crews may be willing to come work on call for an increased rate it may not be realistic to assume that they are onsite and immediately available. Therefore we lowered the random decimal value multiplying the tipple loading rate to be in the range of .75 to 1.05 to give a reflection of the delay that it may take for the additional crews to arrive onsite.

2.2. **Algorithm.**

- Step 1: Generate a randomized train schedule for every day of the week, ensuring that the high capacity train comes on Thursday in the designated time-frame of 11 a.m. to 1 p.m.
- Once we have the randomized schedule, at each time step t:
- Step 2: Check to see if trains have arrived.
- Step 3: If a train has arrived we check the current capacity of the tipple.
  - If the current tipple capacity is greater than or equal to 1 train load of coal, we load the train.
  - If the current tipple capacity is less than 1, we call the maximum number of available crews in to load the tipple until it has at least 1 train load of coal.
- Step 4: If no train has arrived, we check the capacity of the tipple and if needed assign one crew to load the tipple to full capacity.

We chose to implement our algorithm in 'pure' `Python` which can be found in the Appendix. After exploring multiple discrete event simulation languages such as `Simul8, AnyLogic` and the `Python` simulation library `SimPy`, we decided that these approaches were somewhat of an overkill for our problem. We have only queue and one resource that needs to be replenished, and therefore did not require many of the complex processes that these languages have to offer.

3. Results

We chose to display the results of both average weekly and monthly total costs in Tables 4-7 in order to give some scope to how these small differences in weekly costs can quickly compound. In each table, we have time steps t classified into hour and minute intervals, and the maximum number of crews available to load the tipple. Average total, demurrage and crew costs are given as well as the standard deviation σ of each different scenario.

We chose the number of iterations to be 6,000 by observing the density histograms for each 1,000 iterations until the shape approximately resembled the Gaussian bell curve, indicating a normal distribution by the central limit theorem [4]. Such a histogram can be seen in Figure 1 for the base case of 3 standard daily trains, 2 crew maximum, and time steps t into minutes with a kernel density estimation drawn to show the approximately normal distribution. While increasing the number of simulations past 6,000 (see Figure 2) would have provided better convergence of the average and standard deviation, the computation time needed to run such calculations was not feasible with the deadline we were given. This can be illustrated by a similar histogram to Figure 1

The standard deviation is a quantity of particular interest, we can see that increasing both the time interval and the maximum number of crews decreases all associated costs and drives the standard deviation down. For example, Table 4 row 1 shows an average weekly total cost of roughly \$2.3 million with standard deviation $\sigma \approx$ \$1.3 million. In comparison row 3 has all the same quantities with only t measured in minutes instead of hours and gives an average \$1.6 million with $\sigma \approx$ \$287,000. Similarly for a 3 crew maximum we have

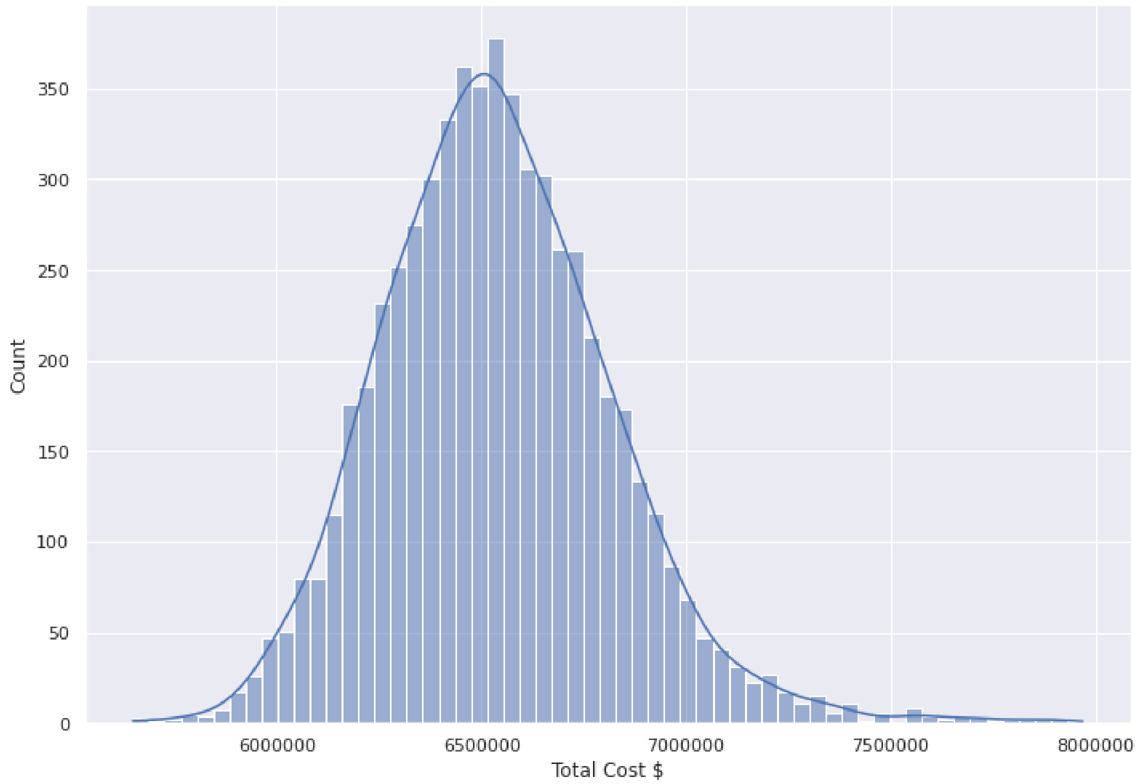

FIGURE 1. Total cost histogram for 6,000 simulations with 2 crews and 3 standard trains approximately follows a normal distribution.

\$1.9 total and $\sigma \approx$ \$862,000 versus \$1.4 total and $\sigma \approx$ \$219,000 for times t into hours and minutes respectively. This illustrates the level of control and efficiency that we can gain from choosing smaller time steps t, since the standard deviation for minutes is significantly lower. It also stresses a previously made statement, increasing the maximum number of crews available helps mitigate demurrage costs by roughly 40% and contributes to a lower standard deviation.

We are now able to answer another question posed: **What are expected monthly demurrage costs?** Table 6 lists results that are relevant to this answer, and we choose to specifically examine the more realistic time steps of $t = 1$ minute. For both cases of 2 and 3 crew maximums, we are incurring approximately \$1.8 to \$2.8 million dollars in monthly demurrage costs on average. Again we emphasize the importance of having a third crew available, as this reduced the average monthly demurrage cost by almost \$1 million dollars.

3.1. **Considering a Fourth Train.** Next we can look to Tables 7 & 5 to answer the question: **Can this tipple support a fourth standard train every day?** Successfully answering this question is difficult, since we are not given any operational budget constraints or guidelines to look to in making a decision. If the coal loading plants output was favored over operational costs, then we could easily say that a daily fourth standard train could be accommodated. However if we are to instead base all of our decisions on solely the operational costs, then it can easily be argued that a fourth daily standard train is not cost effective. Comparing all scenarios from the tables with 3 standard trains and those for 4

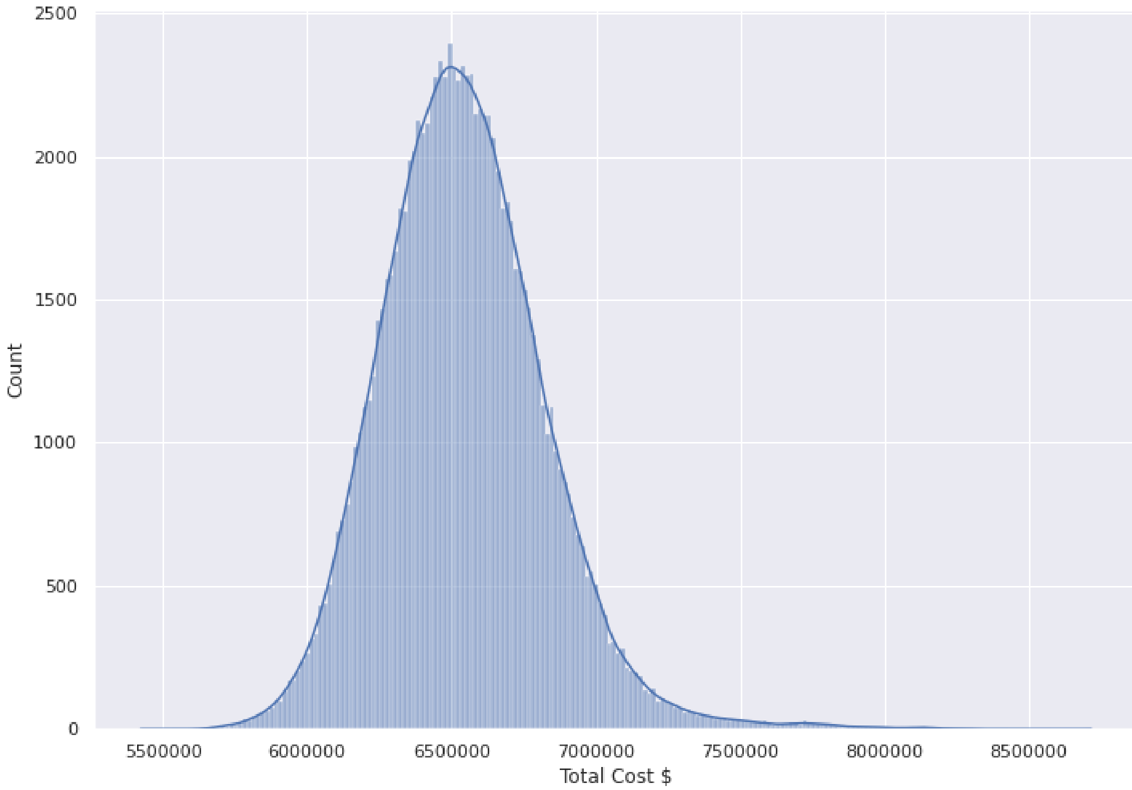

FIGURE 2. Total cost histogram for 100,000 simulations with 2 crews and 3 standard trains approximately follows a normal distribution.

standard trains, we see average cost increases of roughly %40-55 for a %33 output increase. Only the smallest cost increase ($t = 1$ minute and 3 crews) could be argued to be cost efficient, though we again underscore that we have no financial guidelines to reach such a conclusion.

Finally we can answer the main prompt of the problem; **Find the annual optimal cost of operation**. We are confident in our assessment that the annual operational costs of the tipple with 2 crews and 3 daily standard trains of being on average $78,538,933 with a standard deviation of $13,530,728. Again we highlight the importance of a third crew by displaying the same quantities; $68,315,796 on average with a standard deviation of $10,511,484 which indicates an average %14 decrease and thus gives us the minimized and therefore optimal annual costs.

4. Considerations & Conclusion

4.1. **Strengths.** This scenario has many moving, intricate pieces that touch on various classes of problems including scheduling and queuing. That is to say that simply running only a linear program would not provide adequate insight to answer the stated questions. With that in mind, our model gives more than reasonable results to a complex problem so that decisions can be made accordingly. From this fact alone, we would argue that this insight outweighs the undermentioned weaknesses.

Average Weekly Costs & Crews for 6,000 Simulations with t Time Steps

$t =$	# of Crews	Total \$	Demurrage \$	Crew \$	σ
1 hour	2	2,309,231.00	1,222,473.12	1,086,757.88	1,315,385.81
	3	1,914,342.67	785,616.67	1,128,726.00	861,702.09
1 minute	2	1,637,167.76	702,951.14	934,216.62	287,376.71
	3	1,423,245.86	467,318.09	955,927.77	218,989.41

Table 4. Average weekly total, demurrage, maximum # of crews and associated crew costs and standard deviation for 24,000 simulations using t time steps.

What About a 4^{th} Train? (Average Weekly Costs)

$t =$	# of Crews	Total \$	Demurrage \$	Crew \$	σ
1 hour	2	4,984,366.29	3,590,981.67	1,393,384.62	2,765,434.12
	3	3,513,343.79	1,944,959.79	1,568,384.00	1,802,256.98
1 minute	2	3,116,378.53	1,869,569.47	1,246,809.06	510,366.00
	3	2,309,220.75	994,769.43	1,314,451.32	319,602.45

Table 5. Considering an additional 4^{th} standard train.

Average Monthly Costs & Crews for 6,000 Simulations with t Time Steps

$t =$	# of Crews	Total \$	Demurrage \$	Crew \$	σ
1 hour	2	9,236,924.00	4,889,892.50	4,347,031.50	5,261,543.23
	3	7,657,370.67	3,142,466.67	4,514,904.00	3,446,808.34
1 minute	2	6,548,671.04	2,811,804.56	3,736,866.48	1,149,506.83
	3	5,692,983.43	1,869,272.35	3,823,711.08	875,957.62

Table 6. Average monthly total, demurrage, maximum # of crews and associated crew costs and standard deviation for 24,000 simulations using t time steps.

4.2. **Weaknesses.** The weaknesses of our model lie primarily in the assumptions we chose to make. We assumed that crews could be called to the job at a moments notice which is not necessarily a real-world actuality. Inherent in our model is the assumption that all of the loading and unloading moves seamlessly throughout time. In that same light, extraneous situations such as machinery failure were not taken into account.

We also chose to simulate weeks at a time, so that we could observe any possible late arrivals and tipple loading operations that might roll over from one day to the next. In light of this, we acknowledge that scenarios may happen on a Sunday night that truncate current loading processes when the clock hits midnight on Monday. However we'd like to stress that

What About a 4^{th} Train? (Average Monthly Costs)

$t =$	# of Crews	Total \$	Demurage \$	Crew \$	σ
1 hour	2	19,937,465.17	14,363,926.67	5,573,538.50	11,061,736.48
	3	14,053,375.17	7,779,839.17	6,273,536.00	7,209,027.92
1 minute	2	12,465,514.11	7,478,277.89	4,987,236.22	2,041,464.01
	3	9,236,883.01	3,979,077.72	5,257,805.28	1,278,409.81

Table 7. Considering an additional 4^{th} standard train.

any time model will have these types of edge cases, and that simulating months or years at a time may be computationally infeasible without significant memory and performance optimization.

4.3. **Conclusion.** In order to fully answer the given problem, we have simulated all possible scenarios for number of trains and maximum number of crews with differing time steps of hours and minutes. We also sought to implement additional measures such as variable train and tipple loading rates to inject a level of realism into the model. While we have tried to consider every edge case and handle appropriately, we are certain that some detail may have been overlooked and hope to continue to refine this model in the future.

5. Appendix

Below is the Python code we created for our base case of 2 crews, 3 daily standard trains. Note that all subsequent iterations of various scenarios used this code as a template.

```
import random
from random import randrange
import numpy as np
import matplotlib.pyplot as plt
import statistics
from statistics import *
import seaborn as sns

def getHourForTrain(n):
  return randrange(300+24*60*n, 1200+24*60*n)
def getHourForHCTrain(n):
  return randrange(660+24*60*n, 780+24*60*n)

def generateMonth():
  return {
    "week1": {
      "train1": getHourForTrain(0),
      "train2": getHourForTrain(0),
      "train3": getHourForTrain(0),

      "train5": getHourForTrain(1),
      "train6": getHourForTrain(1),
      "train7": getHourForTrain(1),

      "train9": getHourForTrain(2),
      "train10": getHourForTrain(2),
      "train11": getHourForTrain(2),

      "train13": getHourForTrain(3),
      "train14": getHourForTrain(3),
      "train15": getHourForTrain(3),
      "trainhc": getHourForHCTrain(3),

      "train17": getHourForTrain(4),
      "train18": getHourForTrain(4),
      "train19": getHourForTrain(4),

      "train21": getHourForTrain(5),
      "train22": getHourForTrain(5),
      "train23": getHourForTrain(5),

      "train25": getHourForTrain(6),
```

```
    "train26": getHourForTrain(6),
    "train27": getHourForTrain(6),
   },
...
 }

# checks to see if trains have arrived
def isArrived(mins, trains, arrived):
  for train, arriveTime in trains.items():
    if mins == arriveTime:
      if 'hc' in train:
        arrived.append(2) # a hc train takes two trainloads and so we
            increase the demurrage rate but treat as two separate trains
      else:
        arrived.append(0)
  return arrived

# takes currently arrived trains and calculates demurrage
def demmurage(arrived, isLoading):
  dem = 0
  totalCost = 0
  for train in arrived:
    if train == arrived[0] and isLoading:
      dem += 0
    else:
      if train > 0:
        dem += (25000/60)
      else:
        dem += (15000/60)
  return dem

def simMonth():
  month = generateMonth()
  dem = 0
  crewCost=0
  crewCost1=0
  crewCost2=0
  crewCost3=0
  totalCost=0
  for week in month:
    trains = month[week]
    mins = 0
    arrived = []
    tipple = 1.5
    while(mins < 10080):
    # if train has arrived
```

```
      arrived = isArrived(mins, trains, arrived)
    # if there is a train that needs loading
      if len(arrived) > 0:
      # if we have coal in the tipple and it's a regular train, fill it
        if tipple >= 1:
        #takes 3 hours to fill
          tipple -= 1
        #checks to make sure a different train didn't arrive
          for i in range(int(180/(random.uniform(.85,1.05)))): # variable
              train loading rate
            dem += demmurage(arrived, True)
            mins += 1
            arrived = isArrived(mins, trains, arrived)
          arrived[0] -= 1
          if arrived[0] <= 0:
            arrived.pop(0)
        else: # if we don't have coal to fill it, its cheaper to hire a
            second crew
          crews=2
          crewCost2 += ((12000*crews-3000)/60)
          dem += demmurage(arrived, False)
          tipple += (0.5/60)*random.uniform(.75,1.05) # variable tipple
              loading rate
          mins += 1
    # if there are no trains, fill up for an hour
      elif len(arrived) == 0 and tipple < 1.5:
        crews=1
        crewCost1 += ((12000*crews-3000)/60)
        tipple += (0.25/60)*random.uniform(.85,1.05) # variable tipple
            loading rate
        mins += 1
      else: # if we are full of tipple and no one has arrived, do nothing
        mins += 1
  crewCost += crewCost1+crewCost2
  totalCost = dem + crewCost
  #print('Crew Cost 3=',crewCost3)
  #print('Total Cost = %s, Demurrage Cost = %s, Crew Cost= %s' % (totalCost,
      dem,crewCost))
  return[totalCost, dem, crewCost]
totals = []
realTotalCost = 0
realDemCost = 0
realCrewCost = 0
n = 6000
for i in range(0,n):
  totals.append(simMonth())
```

```
  realTotalCost += totals[i][0]/4
  realDemCost += totals[i][1]/4
  realCrewCost += totals[i][2]/4
nptotals=np.asarray(totals)
print('# MINUTES, 2 CREW MAX, 3 TRAINS # MONTHS')
print("MONTHLY Avg Cost = %s, Demurrage Cost = %s, Crew Cost = %s" % ((4*
   realTotalCost) / n, (4*realDemCost) / n, (4*realCrewCost) / n))
print("{:,.2f} & {:,.2f} & {:,.2f} & {:,.2f}".format((4*realTotalCost / n),
   (4*realDemCost / n), (4*realCrewCost / n), 4*stdev(nptotals[:,0])))
print("WEEKLY Avg Cost = %s, Demurrage Cost = %s, Crew Cost = %s" % (
   realTotalCost / n, realDemCost / n, realCrewCost / n))
print("{:,.2f} & {:,.2f} & {:,.2f} & {:,.2f}".format((realTotalCost / n), (
   realDemCost / n), (realCrewCost / n), stdev(nptotals[:,0])))
```

References

[1] 2013. URL: http://www.coaltrainfacts.org/key-facts.

[2] *1993: The Coal-Tipple Operations Problem.* 1993. URL: http://www.mathmodels.org/Problems/1993/MCM-A/1993A.pdf.

[3] Steven B. Horton Frank R. Giordano William P. Fox. *A First Course in Mathematical Modeling.* 5th ed. Cengage Learning, 2013. ISBN: 1-285-05090-8,978-1-285-05090-4.

[4] Saeed Ghahramani. *Fundamentals of Probability, with Stochastic Processes, 3rd Edition.* 3rd ed. Prentice Hall, 2004. ISBN: 0131453408,9780131453401.

[5] Gary Goddel. *How Much Coal, In Tons, Can a Railroad Car Hold? - Quora.* URL: https://www.quora.com/How-much-coal-in-tons-can-a-railroad-car-hold.

[6] *How Much Does a Train Weigh?* 2021. URL: https://trainconductorhq.com/how-much-does-a-train-weigh/.

[7] John Petty, Ray Eason, and John Rufenacht. *Coal-Tipple Operations.* URL: https://www2.kenyon.edu/Depts/Math/Paquin/1993Smcm.pdf.