Optimize Switching in a Flat Rail Yard

1. Executive Summary

Switching wagons in large flat yards is arguably one of the most complex daily actions a railway performs. Doing it efficiently on a limited number of tracks of varying sizes while building many train-blocks takes skill and experience.

Little has been done to develop algorithms to assist engineers and conductors that have to figure out the sequence of steps to take to properly and efficiently sort a set of wagons[1].

The goal of this problem solving competition is to build open-source software and algorithms that can plan out the sequence of moves required to sort.

2. Introduction

There are several types of railway yards used for classification of freight wagons: Hump, gravity, and flat yards. This competition will focus only on flat yards and have wagons switched using locomotives. There are many more flat yards than hump or gravity yards. A general rule of thumb for North American railways is that flat yards are less expensive to operate when the number of wagons being switched is 1500 wagons per day or less. Hump yards are more efficient for larger number of wagons being switched per day.

2.1. Terminology

For North American freight railways, the terms 'car' or 'railcar' are typically used to represent the railway vehicles that carry cargo. However, railcar is not a universal term. For example, in Britain a railcar is a self-propelled vehicle for passengers that operates on railways. Other terms for railway cargo carrying vehicles is bogie, truck and wagon. Bogie and truck have different meanings in different parts of the world – for example, bogie in England represents the frame while in India it represents the entire vehicle.

Throughout this document we will use the term **wagon** for cargo carry railway vehicles. It is used in most countries.

A train typically consists of one or several locomotives and a set of wagons. The locomotives power the train. Within a railway yard, smaller locomotives are used for moving around wagons. These are locomotives in most of the world (but not North America) are called **shunters**. Henceforth, locomotives for yard operations will be called shunters.

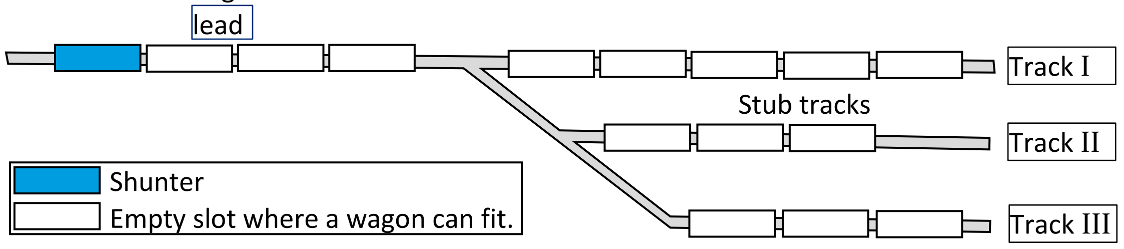
[1] The only commercial company the author is aware of that has a system for flat yard switching is https://www.cedarai.com/. Also, BNSF has announced some software on this, but little is known at this point (click here)

2.2. Toy Example

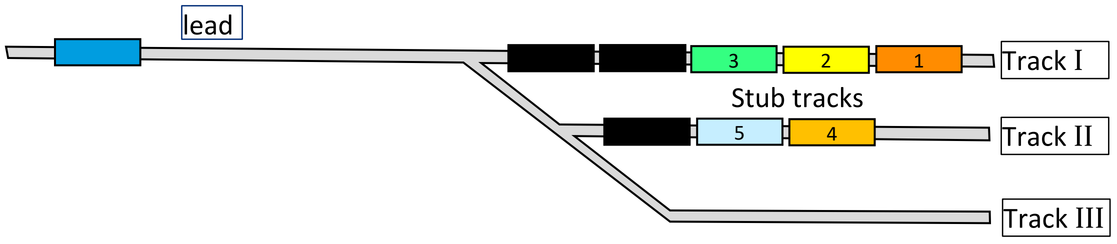
Switching wagons in a flat yard is challenging. It takes significant planning to do it efficiently.

As a toy example, we will take the 'Inglenook Sidings' puzzle that shows some of the complexities of real-life.

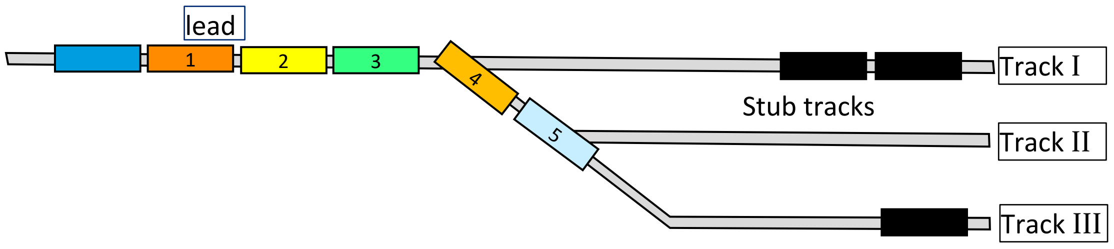
In this small puzzle, there is a 'lead' track and three stub-end tracks. The lead track has room to fit one shunter (in blue) and three wagons. Track I has room for 5 rolling stock. The other two stub tracks have room for three wagons each.



The puzzle uses 8 of the 14 slots for wagons (leaving 6 slots free to use to reposition wagons) and randomly chooses 5 of the 8 wagons to position into a train. Copying from https://www.wymann.info/ShuntingPuzzles/Inglenook/inglenook-rules.html, suppose we have 8 wagons of which the wagons numbered 1-5 need to be pulled onto the mainline to exit left, and they must be in order.



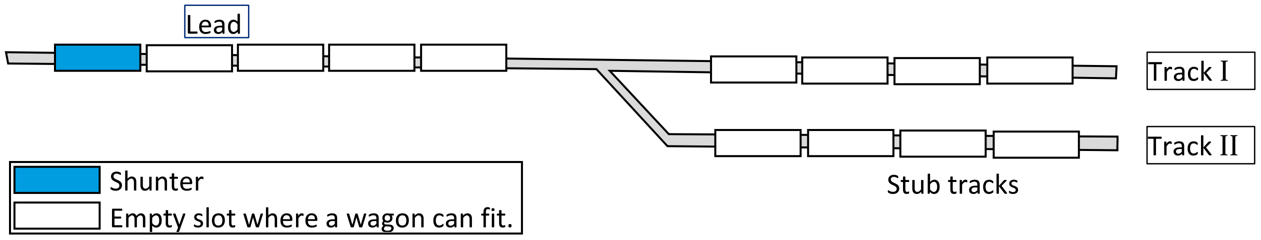
It takes about 17 back-and-forth movements of the engine to get to the solution where the shunter and the five numbered wagons (in order) can exit to the left.



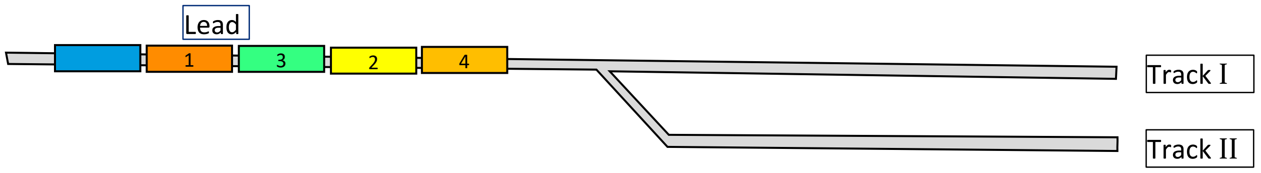
A recent reference to this puzzle with good citations is found in https://arxiv.org/pdf/1810.07970.pdf.

2.3. Another Toy Example to Show the Importance of Optimization

Consider the example in the unpublished manuscript by Xuesong Zhou “An Exact Approach to The Train Marshaling Problem with A Focus on Car Movement Cost”, where there is room for 4 wagons on both stub tracks. The two stub tracks are labeled I and II.

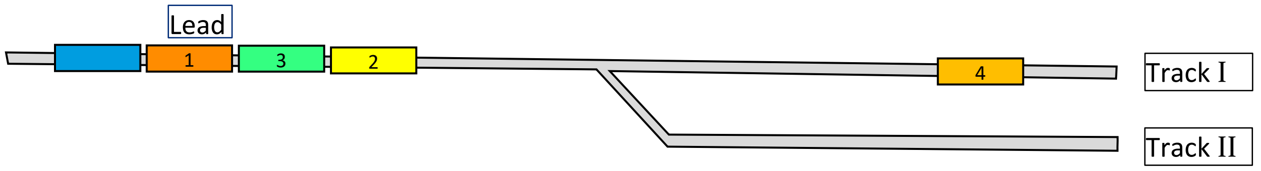


Suppose there are four wagons labeled 1-4 and in the order of 1-3-2-4 attached to the shunter in the lead. Suppose that we want to reorder them into the sequence 4-3-2-1



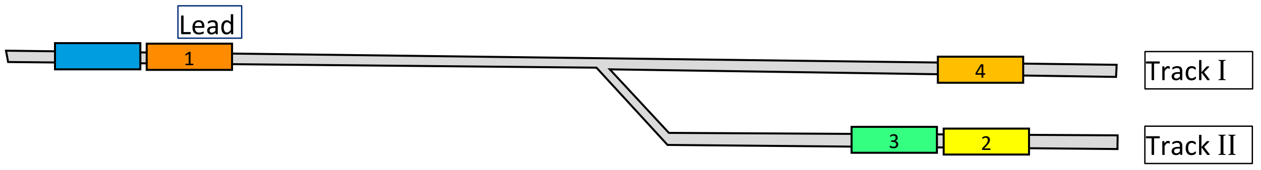
We proceed by having the shunter shove wagon 4 onto track I.

- Pushed 1 wagon



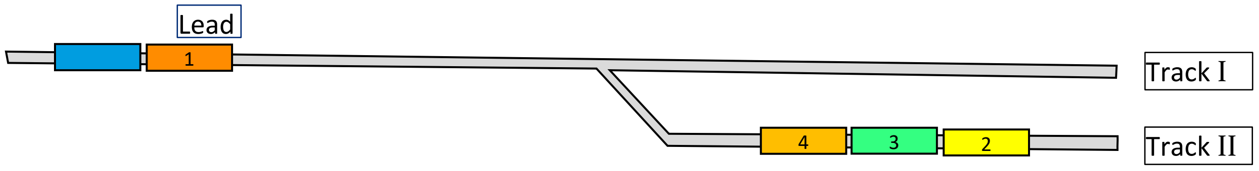
Next we place wagons 3, 2 on track II.

- Pushed 2 wagons



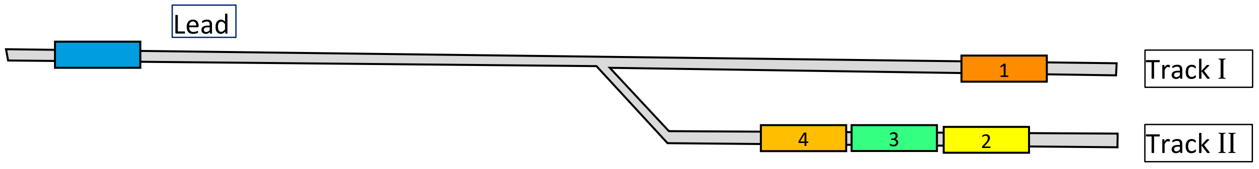
Then pick up wagon 4 and place it onto track II.

- Pulled 1 wagon
- Pushed 1 wagon



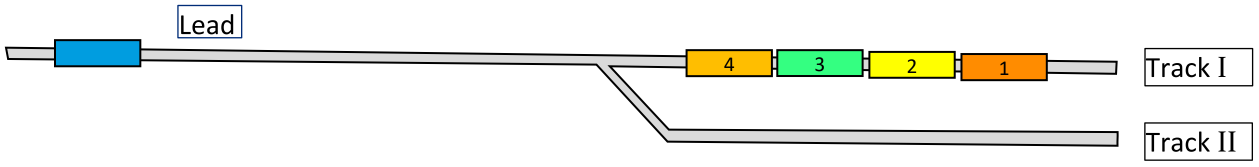
Place wagon 1 on track I.

- Pushed 1 wagon



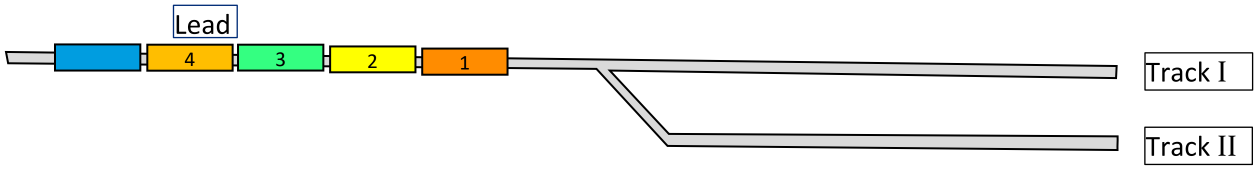
Pickup wagons 4, 3, 2 and place onto track I.

- Pulled back 3 wagons
- Pushed 3 wagons



Pull out all the wagons, and this is complete.

- Pulled back 4 wagons

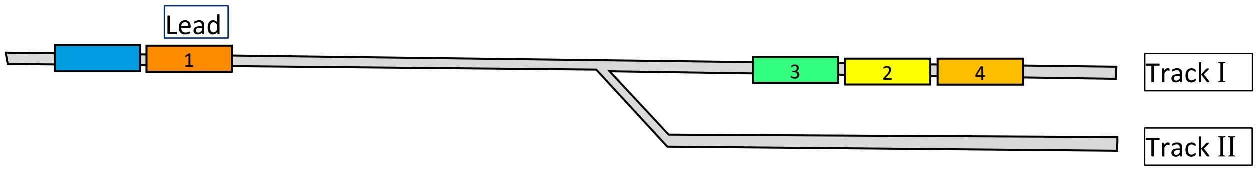


Count the operations:

- Pushed 5 times a total of 8 wagons
- Pulled 3 times a total of 8 wagons

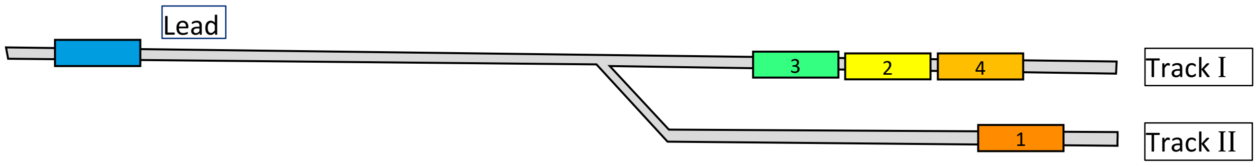
However, there is a better way. Starting over, place wagons 3, 2, 4 onto track I.

- Pushed 3 cars



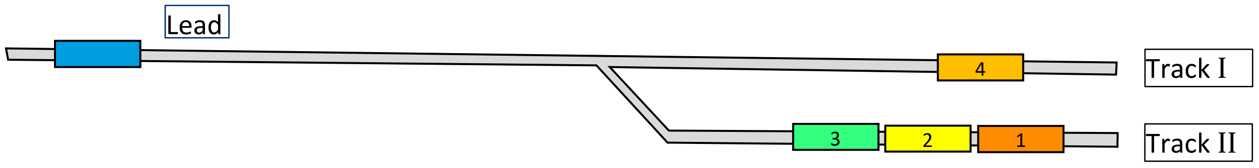
Place wagon 1 on track II.

- Pushed 1 car



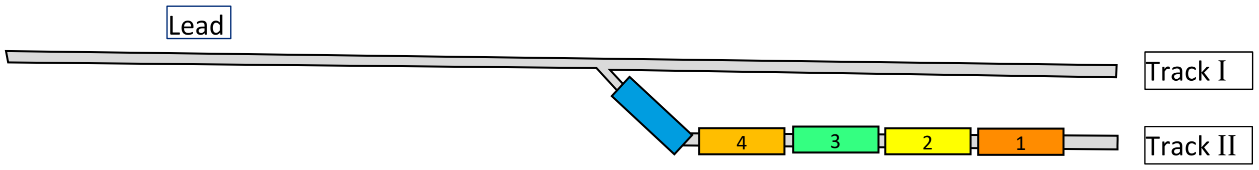
Take only wagons 3, 2 from the track I and put them onto track II:

- Pulled 2 wagons.
- Pushed 2 wagons.



Pull out wagon 4 from track I, then pick up all the wagons from track II, and then pull out.

- Pulled 1 wagon
- Pushed 1 wagon
- Pulled 4 wagons



Total count of operations:

- Pushed 4 times a total of 7 wagons.
- Pulled 3 times a total of 7 wagons.

This is an improvement. For realistic problems, the number of ways to switch wagons goes exponentially with the size of the problem.

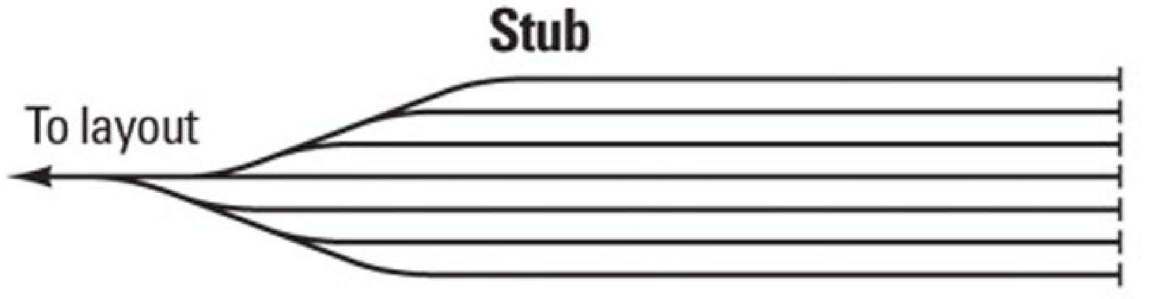
2.4. Typical Flat Yard Configuration

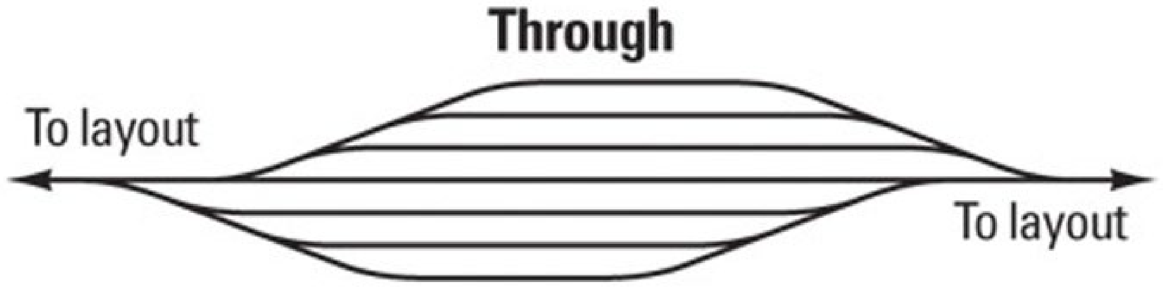
There are several flat yard configurations. The most popular are the *Stub* and *Through* configurations[2], as shown below in diagrams that were meant for model railroaders but do convey the concept. The Inglenook Puzzle discussed in the previous section is an example of a Stub configuration.

[2] https://www.trains.com/mrr/how-to/expert-tips/three-types-of-staging-yards/

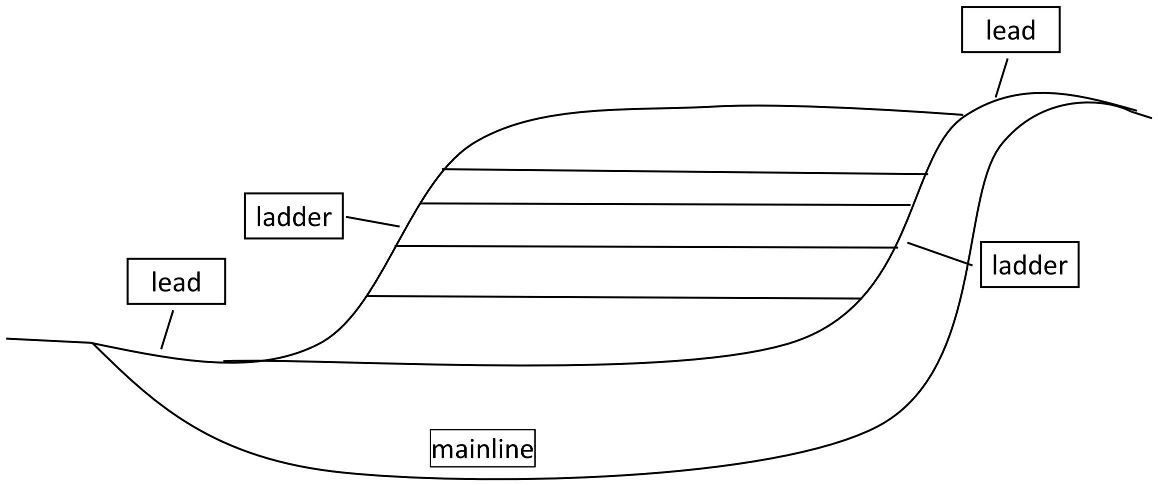
In either configuration, a shunter takes a set of cars and moves them onto one track (by setting the appropriate switch), uncouples some of the cars at the front, backs off the tracks and then shoves other cars onto other tracks.

The stub configuration is space-efficient, while the through configuration is more flexible. For example, if a train has both rear and front locomotives, a train could enter a through yard and easily detach both locomotives. Another use of the through configuration is that the shunter could 'run around' and enter the other side assuming there was an empty track (although this is uncommon).





Another example of a through yard but showing that the yard is typically attached to the mainline but not on the mainline is below. The 'lead' (also called the shunter neck some of the above images) is depicted as being short, but in real life it is quite long to accommodate the shunter and many wagons and not to interfere with the mainline.



In India, for example, the configuration may have a receiving (aka reception) and departure yard. Two configurations are below[3]. Trains enter from the mainline into the receiving tracks. The locomotives are detached and moved to the Loco yard. The shunter takes the wagons and in a series of back and forth moves uses the sorting lines to switch and classify the wagons and then puts the set of wagons that are in the correct order back in the departure yard. There the wagons are inspected, locomotive(s) is attached, and the train gets ready to be pulled out and run on the mainline.

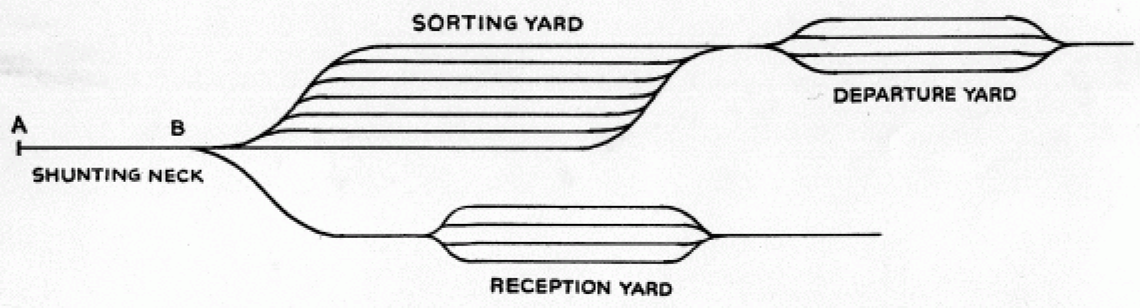


Fig. 1 FLAT YARD

[3] Irfca reference

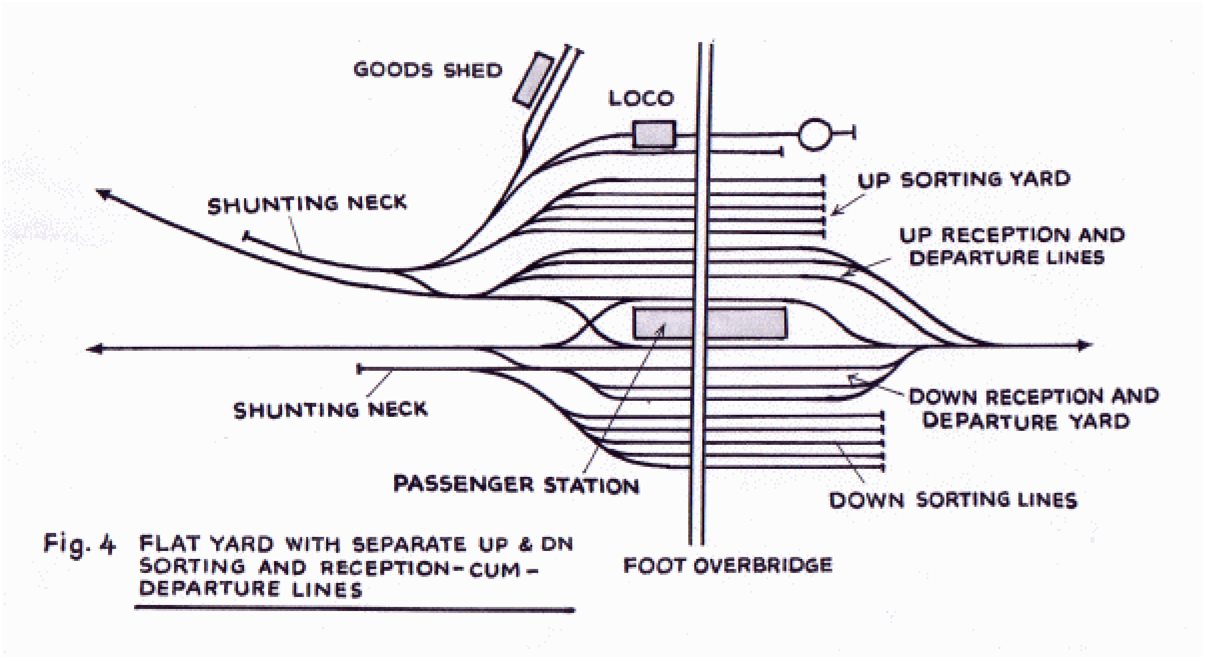


Fig. 4 FLAT YARD WITH SEPARATE UP & DN SORTING AND RECEPTION-CUM-DEPARTURE LINES

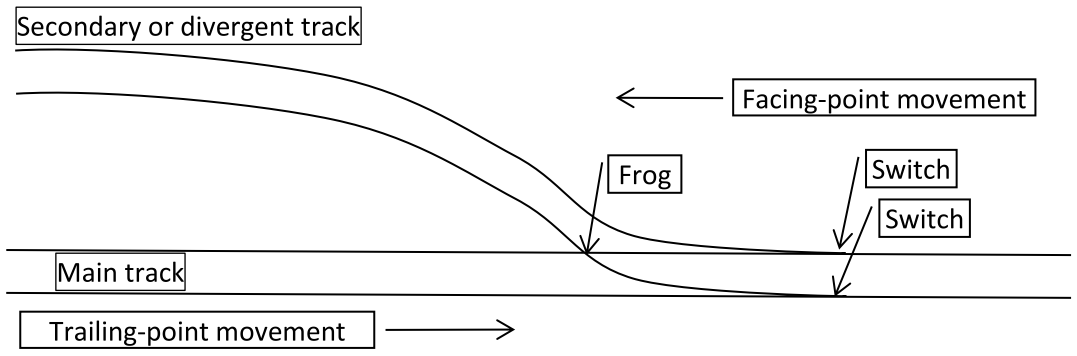
The tracks in a flat yard are not always the same size. Some tracks are short, some might be long. **The length of the track must be a consideration** in this competition.

Also, a typical flat yard is not exactly flat. Rather it has a 'bowl' shape. For through yards, the part of the track near the ladder is higher than the track in the center between the ladders. The reason is to keep wagons from rolling out of the yard and creating havoc.

This next picture, from https://virtualglobetrotting.com/map/decatur-yard-ns shows Norfolk Southern's Decatur Yard, the largest flat yard in North America. This yard seems to have two sections, with each having approximately 36 tracks for sorting. One section is probably for eastbound trains, the other for westbound trains trains. We note this is a through yard.

NS East Decatur Tower
NS Decatur Terminal
A E Staley Cooling Lake
Electric Substation
Primient Grain
ADM Randall Research Center
Grant Park
Torrence Park

Finally, we all some terminology about switches. In this competition, we only focus on a one type of switch used in a flat yard, sometimes called a facing-point switch. It looks like:



We occasionally refer to facing-point and trailing-point movements.

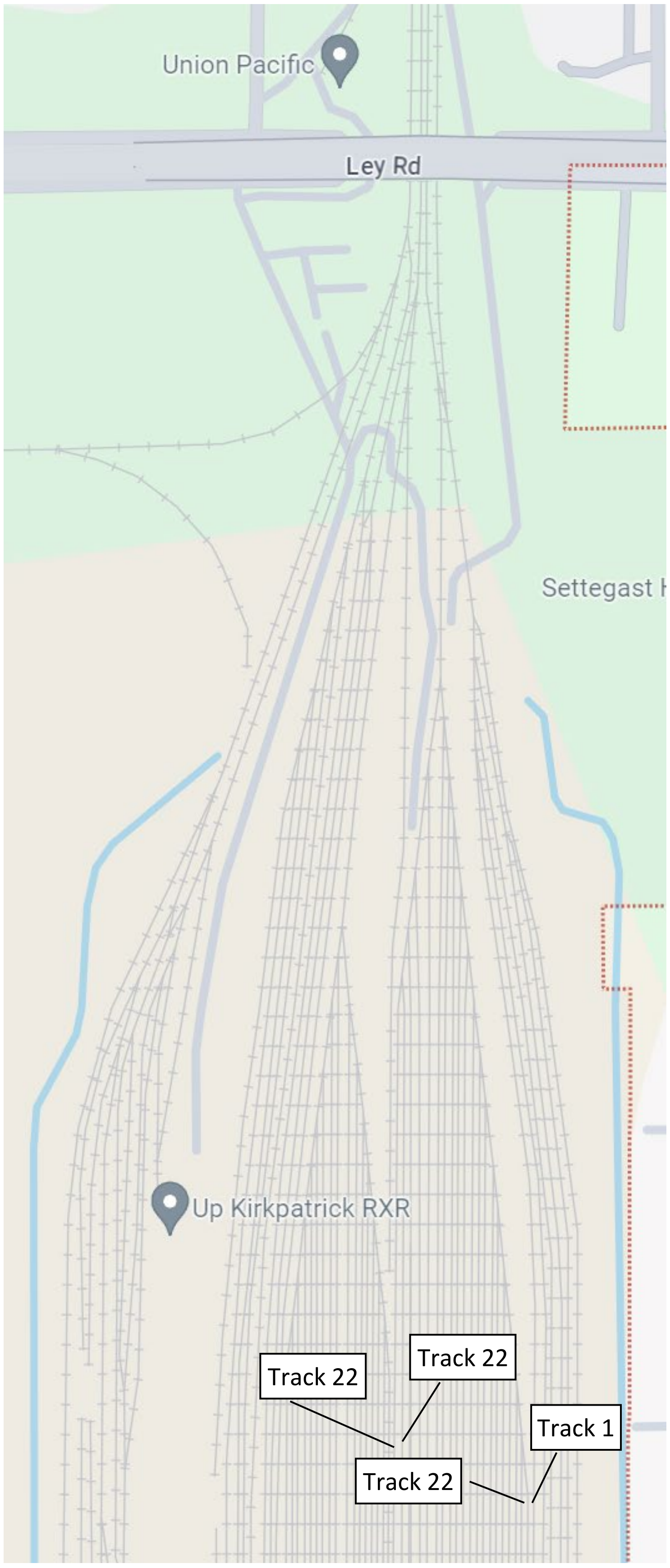
2.5. The Rail Yard Configurations for this Competition

We will fix how a stub yard and a through yard will look in this competition. Before doing so, we will show parts of a large flat yard in the North America to illustrate how complex some flat yards are.

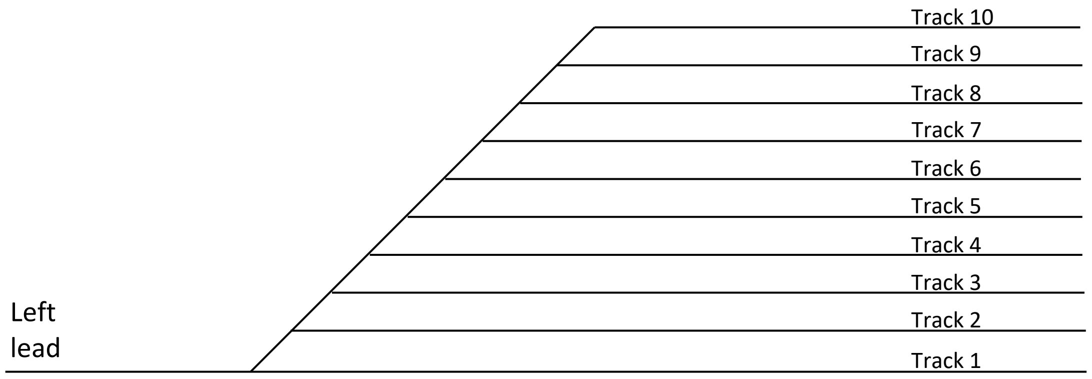
The image below is that of the north end of Settegast Yard, a flat yard owned by the Union Pacific in Houston, TX.

From the image, we see that a shunter can go into track 1, pull out some wagons, then go to track 2 and deposit some wagons over a short distance.

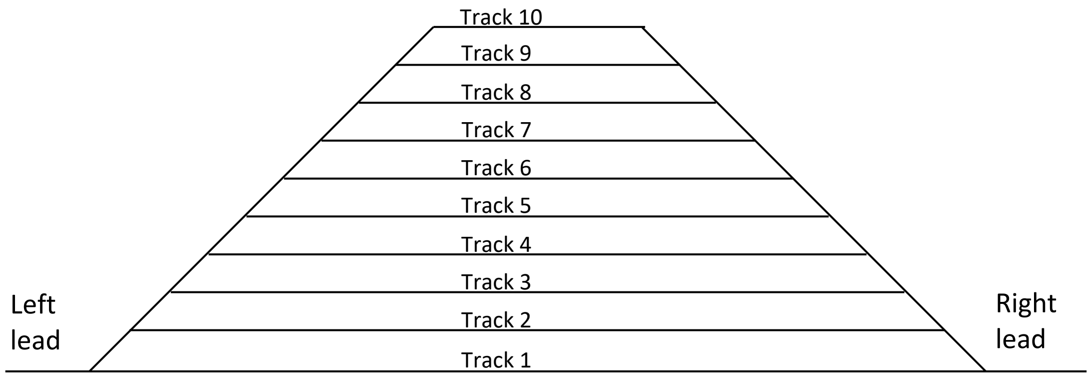
Not so between tracks 22 and 23. The shunter would need to back up on track 22 and the ladder around 0.85 km to get to the switch that would then enable it to get to the part of the yard where track 23 is. A total of around 1.7 km. Further, several switches would need to be thrown.


Union Pacific
Ley Rd
Settegast
Up Kirkpatrick RXR
Track 22
Track 22
Track 1
Track 22

To keep the problem statement simpler, and to support an objective function that is relatively easy to state, we will assume stub end yards look like:



For through yards, we will assume a configuration that looks pyramid-like:



When specifying the length of the leads, for stubs the right-lead will be ignored.

2.6. Typical Shunter Moves

There are two types of shunter moves: kicks and shoves. In this problem competition, we will require the team to model shoves. But for completeness we will describe both

Shoves

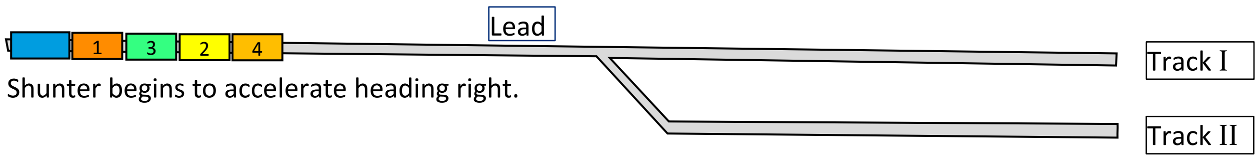
A shove is where a shunter takes a set of wagons from a track, backs out to where the entire train clears the switch. This is called a **pull**. A switchman throws switches so that the shunt moves forward and then cars enter another track. At some point, a set of wagons at the front is decoupled and stays on that track. This is called a **push**.

A shove is typically a pull then push pair of moves. It can take quite a while to perform depending on how far the pull/push travels, the time it takes to operate (typically manually) the switches, and the number of wagons attached to the shunter.

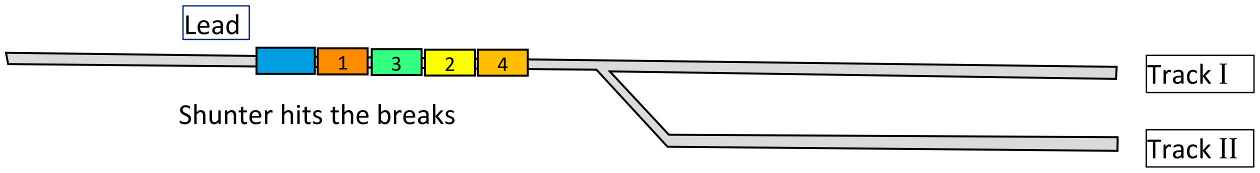
Kicks

The process of kicking wagons is not in the scope of this competition. We include it as background information.

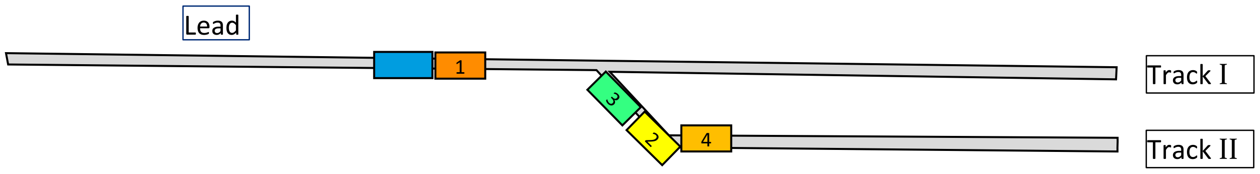
The following is one of many variations of the ‘kick’ process. Consider the example below with a shunter and 4 attached wagons on the lead. Suppose the desire is to place the three wagons at the end (wagons 3, 2, 4) onto track II. The shunter begins to accelerate and reaches, say 6.4 KPH



Before wagon 4 reaches the switch, the wagons are compressed due to the acceleration. One person unpins the wagons between wagon 1 and wagon 3, another person throws the switch so that the wagons would go onto track II.

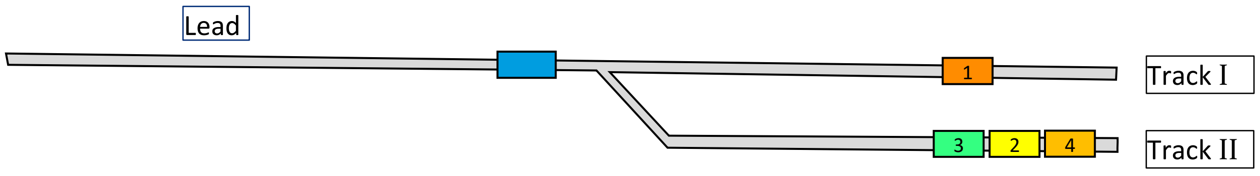


Then the engineer guiding the shunter applies breaks, and the shunter and wagon 1 slow down, while wagons 3, 2, 4 use their momentum to go onto track II. The wagons 3, 2, 4 would eventually stop, especially if the yard is a ‘bowl’ where the right side of the tracks have a slight upward grade. If this is not a bowl, there might be a employee on wagon 4 that applies a handbrake at an appropriate time to stop the wagons.



This is known as a **kick**. It is much faster than a shove, which would have the shunter enter track II, possibly for a long way down.

Sometimes, after the connection between wagons 1 and 3 are uncoupled and the wagons move to track II, there is enough room and enough time to uncouple wagon 1 from the shunter, accelerate the shunter a bit more and to throw the switch back, break the shunter to slow down, so that wagon 1 could separate and head into track I. The shunter might never have to enter track I.



This is called a **double kick**.

It takes skill and teamwork to effectively do a kick. The employees that unpin, throw switches, possibly operate hand brakes, and govern the shunter need to work in unison.

Kicking also takes more room on the lead, and there are some types of wagons and their cargo that are not allowed to be kicked.

To properly model this is way beyond the scope of this competition and will not be consider here. It is discussed here only to give the reader a better sense of the complexities of flat yard switching.

3. Our objective function

A realistic objective function is to minimize the time it takes to switch the wagons.

However, modeling that is too onerous because estimating the time required has too many details that will make this problem-solving competition overly complex. For example, the following contributes to the switching time:

- Characteristics of the locomotive such as acceleration, power, tractive effort.
- Weight of wagons being shoved, which is related to the number of loaded versus empty wagons being shoved.
- Length of track that needs to be transversed, including the length of the ladder between the track that wagons are pulled from and the track the wagons are pushed to.
- Kicking versus shoving. While kicking is not considered in this competition, it is significantly faster than shoving.
- Skill of the crews.

In this competition, we estimate the time each push and pull takes based on:

- The average length of the switching track
- The track number where the shunter pulls wagons and the track number where the shunter pushes wagon
- The number of wagons being pushed or pulled

There are several constants, and each problem set will define these constants:

Data	Description	Example
s_m	Max speed (km / s, not kph)	16/3600 kps
α_a	Constant term in acceleration	6750
β_a	Wagon increment term in acceleration	292.5
α_d	Constant term in deceleration	6750
β_d	Wagon increment term in deceleration	225
l	Speed on ladder (km / s)	10/3600 kps
r	Distance on ladder between tracks (m, not km)	3 m

The time it takes to pull or push n wagons on a track of length D is

$$\frac{D}{2s_m} + \frac{s_m}{2}\left((\alpha_a + \alpha_d) + n(\beta_a + \beta_d)\right)$$

Also, the time it takes to move on the ladder from track i to track j is

$$|i - j|\frac{r}{1000l}$$

Details are in **Appendix** 1: **Derivation of Objective Function**.

As an example, suppose a shunter is on the lead but close to the ladder, and needs to pick up 20 wagons on track 1 and, move back enough to clear the switch, then move to track 5, deposit 3 wagons there, then move back to clear the switch. Using the example number above and assuming the length of track 1 is 2 km, we have:

$$\gamma_1 = \frac{s_m}{2}\left((\alpha_a + \alpha_d) + n(\beta_a + \beta_d)\right) = \frac{16/3600}{2}\left((6750 + 6750) + (292.5 + 225)n\right)$$
$$= 30 + 1.15n$$

Time (Seconds)	Action	Calculation
255	Shunter with no wagons goes from lead to middle of track 1. Assume that Shunter begins next to the switch	$\frac{2km}{2(16/3600)} + (30 + 1.15 \times 0)$
278	Shunter pulls 20 wagons back to clear the switch	$\frac{2km}{2(16/3600)} + (30 + 1.15 \times 20)$
4	Shunter transverses up the ladder from track 1 to track 5	$\lvert 1 - 5\rvert \frac{3}{1000 \times 10/3600}$

The Excel spreadsheet "Example 1.xlsx" has an example of the calculations. Please understand that spreadsheet.

4. Problem (Input Data) Format

We will use JSON to describe the format of each problem.

As an example, consider the following:

```
{
  "problem_name": "Example 1",
  "yard_type": "stub",
  "parameters": {
      "alpha_a": 6750,
      "alpha_d": 6750,
      "beta_a": 292.5,
      "beta_d": 225,
      "track_speed_kph": 16,
      "ladder_speed_kph": 10,
      "wagon_length_m": 15,
      },
  "track_lengths_m": [2000, 1900, 1800, 1700, 500],
  "left_lead_m": 2000,
  "right_lead_m":   0,
  "track_occupancies": {
      "1": [1, 1, 2, 2, 0, 4],
      "2": [2, 0, 4, 0, 1]
      },
  "desired_block_order": [0, 2, 1, 4]
}
```

This says that "Example 1" is for a stub-type flat yard. It has various parameters relating to the objective defined (alpha_a, etc.). Also within parameters it says the wagons are 15 m long. We will also assume the shunter is the same length as a wagon.

There are 5 tracks with lengths from 2000 m to 500 m. If a track is 2000 m, it means that it can hold $\lfloor 2000m/15m \rfloor = 133$ wagons.

The left lead is 2000 m long. Because this is a stub yard, there is no right lead so it is ignored (and set to 0 in this case).

Two tracks have wagons in them. Track 1 (note, some 'JSON validators' want the track number to be quoted) has 6 wagons. The first two wagons belong to block 1, the next two belong to block 2, the fifth wagon belongs to block 0 and the sixth wagons belong to block 4. And on track 2, the first wagon belongs to block 2. Etc. Note:

- Tracks 3, 4, 5 are empty.
- Block codes are not necessarily consecutive. Note in this example, there is no block "3". Block codes do not have to start with 0.
- Tracks will always be numbered consecutively starting from 1 (and not 0 as is common in most programming languages)
- All fields above are required.

The desired block order field says that we will need to have the wagons rearranged so that the first block is 0, the next block is 2, etc.

There are two types of valid solutions. The first type is where all the wagons are together on a single track. Any track will do. In this example, the wagons should be in the order 0, 0, 2, 2, 2, 1, 1, 1, 4, 4 on a single track.

However, a second acceptable solution is where the wagons are on separate tracks but can be combined in a straightforward manner, such as

- 0, 0, 2, 2
- 2, 1, 1, 1, 4, 4

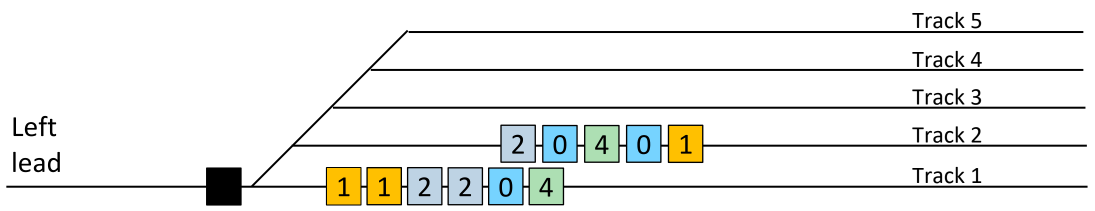
Even with block 2 being split between the two tracks, it is straightforward to pull the wagons

5. Solution (Output) Format

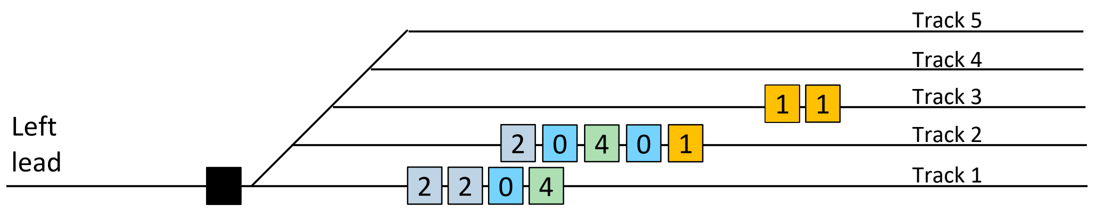
Keeping with the example in Section 4 above, this section will work through one (possibly sub-optimal) solution and demonstrate how to code up the solution.

Initially, we have wagons on track 1 with block codes 1, 1, 2, 2, 0, 4, and wagons on track 2 with block codes 2, 0, 4, 0, 1. Tracks 3, 4 and 5 are empty. The shunter (represented by the black rectangle) is just on the lead side of the switch for track 1. Obtaining a solution (not necessary the optimal one) is easy in this case because there are 5 tracks and only 4 blocks.

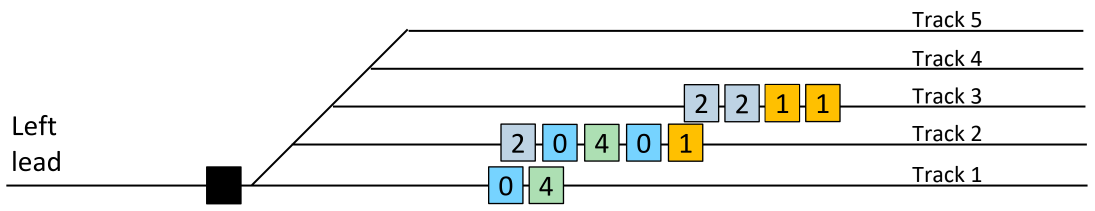
The desired outcome is to have the wagons rearranges so that the order is 0, 0, 0, 2, 2, 2, 1, 1, 1, 4, 4.



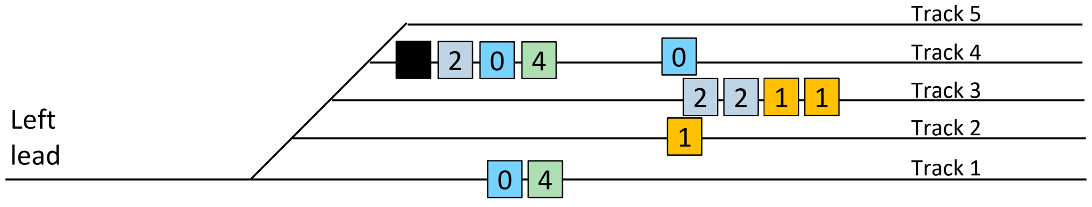
Pull 2 wagons from track 1, and push them to track 3



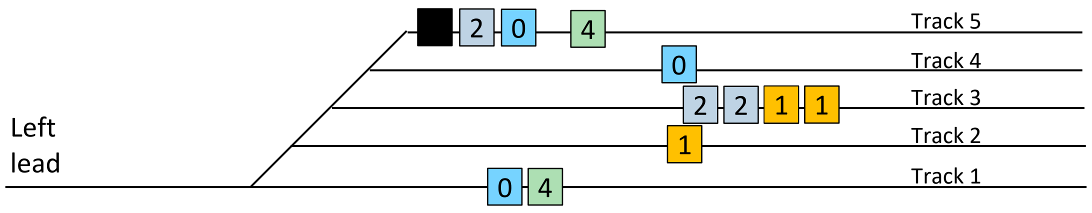
Again, pull 2 wagons from track 1, and push the to track 3



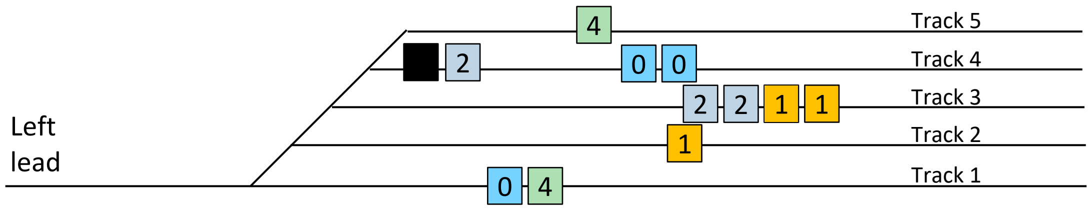
Pull 4 from track 2, push 1 onto track 4



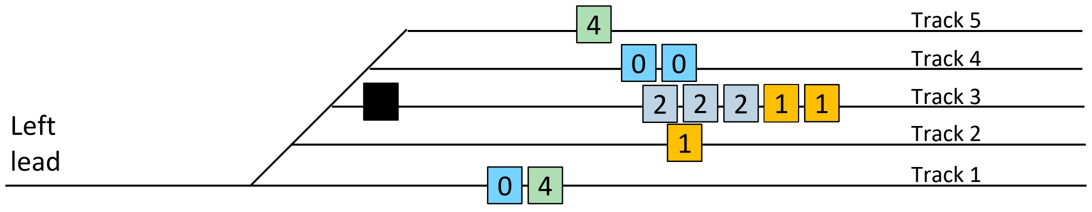
Push 1 onto track 5



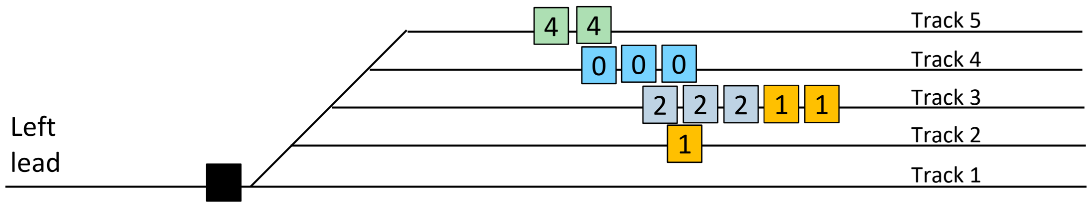
Push 1 onto track 4



Push 1 onto track 3



Pull 2 from track 1, push 1 to track 5, push 1 to track 4, have shunter go back to the lead.



This would be considered complete. A locomotive (not necessarily the shunter) can pull from track 1, then track 4, then track 3, then track 2, then track 4 and have a the outbound train formed.

The solution will be encoded in JSON format as follows: Note that the last one is a 'pull 0 wagons' which will mean return the shunter to the lead.

```
[
{"movement": "pull", "num_wagons": 2, "track": 1},
{"movement": "push", "num_wagons": 2, "track": 3},
{"movement": "pull", "num_wagons": 2, "track": 1},
{"movement": "push", "num_wagons": 2, "track": 3},
{"movement": "pull", "num_wagons": 4, "track": 2},
{"movement": "push", "num_wagons": 1, "track": 4},
{"movement": "push", "num_wagons": 1, "track": 5},
{"movement": "push", "num_wagons": 1, "track": 4},
{"movement": "push", "num_wagons": 1, "track": 3},
{"movement": "pull", "num_wagons": 2, "track": 1},
{"movement": "push", "num_wagons": 1, "track": 5},
{"movement": "push", "num_wagons": 1, "track": 4},
{"movement": "pull", "num_wagons": 0, "track": 0}
]
```

The last step, pull 0 wagons to track 0 means to have the shunter return the lead without any wagons.

Besides the obvious meaning of the movements "push" and "pull", we will allow one more movement type called "switch_sides". It can only be used:

- For through-yards. This is illegal for stub-yards..
- With 0 wagons. The shunter should not have an wagons attached
- On a track that is empty. The shunter will be positioned on the ladder, just clear of the switch for the track.

Switching sides within a through-yard is not a typical move and may not be worth the time it takes to go from one side to another. However, we wanted to leave this as a possibility.

This example is also in "Example 1.xlsx", an Excel spreadsheet that list all the moves, as well as the wagons on the track after each move, and the cost (time) for each move.

- The time is calculated to include time to go in and go out of a track. For "pull" moves, the movement time is the time after wagons are attached and the shunter pulls the wagons. The reverse movement time is the time when the shunter goes into the track. Note for pull moves, the pull will always have move wagons than the reverse move.
- Likewise, for "push" moves, the movement is the time when the shunter goes onto the track and leaves wagons behind. The reverse movement time is the shunter then leaving the track. The push will always have more wagons than the reverse movement.

6. Feasibility Checker and Objective Calculator

Sometime in May there will be made available a Jupyter Notebook coded in Python that will accept the input problem data and a solution file and check feasibility, whether the solution meets the desired block order, and the objective (total time). It will be a more complete version of the "Example 1.xlsx" spreadsheet. This will be used during judging as one of the criteria.

7. Rules of the Competition

There is a GitHub repository (https://github.com/MarcMeketonVillanova/2024_RAS_PSC) that has this document, the problems to be solved, the JSON schemas that will be used to validate inputs and your outputs, and soon the feasibility checker.

We expect that your source code will be placed in a GitHub repository. It could be a private repository, however if you become a finalist a member of the judging team will need to have access to your repository. We expect to compile your code and run it for ourselves as part of the judging process, including running it on new problems.

This implie5s that you should use a mainstream computer language such as Java, C#, Python, C++, Rust. If you use libraries, they should be open source, except for linear/integer programming solvers.

Evaluation Criteria: The criteria that judges will use to evaluate a solution include the following:

- Feasibility of the proposed solutions. Please use the feasibility checker that will be provided to ensure your solution is feasible.
- The quality of the proposed solutions for the given objective function values. Please use the feasibility checker to compare your objective to the one calculated in the feasibility checker.
- The tractability of the solution approach adopted.
- The usability/reproducibility of the solution approach.
- The performance of the solution approach, e.g., its computational time.
- The quality and clarity of the paper describing the solution approach considered for the problem.
- Submissions must comply with the specified format for solution files (Section 5).
- The quality of the presentation, to be given by three finalist teams at the Rail Applications Section Meeting for the 2024 INFORMS conference. This presentation might be video recording if travel restrictions are in place.

The finalists will make a presentation at the 2024 INFORMS Annual Meeting (or a video recording if travel restrictions are in place).

Note that being among the finalists and presenting at the Annual Meeting does not guarantee a finalist will receive first, second or third place. The decision of the judges is final.

Awards:

First Prize: $2,000

Second Prize: $1,000

Third Prize: $750

In addition to the cash prizes, the first prize winners' contribution to this competition will also be considered for publication in the journal Networks. The paper still needs to go through the journal's refereeing process; however, it will receive an expedited refereeing and publication process.

Eligibility: Practitioners of operations research and management science who are interested in solving problems in the railroad domain using Operations Research and Analytics tools are welcome to participate. Registration is open to all with the exception of RAS officers and organizing committee members. Likewise, members of the organizing committee may NOT help nor guide any participating team. Teams of up to five members can participate. At least one member of each finalist team must be available in-person or virtually for questions/answers at the 2024 INFORMS Annual Meeting. Winners will be announced after the session.

Registration: Participation in the RAS Problem Solving Competition requires registration by June 15th. Every team must register by the due date to participate in the contest. To register, please send the following information to railwayapplicationssection@gmail.com by the deadline.

- For each team member: Name, Email, Organization, Position.
- Brief statement describing what motivated you to participate.

After submitting your registration email, you will receive an email confirming your team's successful registration and eligibility, and you will be added to the competition's channel in Slack.

Can I publish? Yes, you can. In fact, RAS encourages you to do so. Anyone can use the RAS competition problem and provided datasets in their publication. References to year-specific problem competitions are given in the URL, and as such you can reference the year-specific competition URL which will not be changed.

Important dates:

- Release of Full Problem and Datasets: April 1st, 2024
- Registration Deadline: June 15th, 2024
- Q&A Period: Open from May 1st, until August 15th 2024.
- Solution Submission Deadline (Paper + Results): August 15, 2024. Submissions must be sent to rasproblemsolving2024@gmail.com
- Announcement of Finalists: August 31 st, 2023
- Finalists' Presentations: October 20-23, 2024, at INFORMS Annual Meeting, Seattle, WA.
- Winner Announced: October 20-23, 2024, at INFORMS Annual Meeting, Seattle, WA.

Competition chairs:

- Marc Meketon (marc.meketon@oliverwyman.com)
- Xuesong Zhou (xzhou74@asu.edu)

Appendix 1: Derivation of Objective Function

The objective function represents the time it takes to pull and push cars. It was largely derived from what is scientifically called handwaving. Some of the below was inspired by this paper which, among other things, suggests that the time to accelerate is linear to the weight being pulled/pushed.

- Max speed is 16 kph
- Typical wagon is 65 (metric) tons. This is an average of loaded and empty wagons. This was loosely used to develop the 1.3s acceleration factor below, inspired by the paper cited above.
- It takes 30s + N * 1.3s to accelerate to 16 kph (N is the number of wagons)
- The shunter has to travel 1 km
- It takes 30s + N * 1.0s to decelerate to 0 kph
- While on the ladder and going though switches, the train travels 10 kph.

The distance traveled to accelerate, with an average sped of 8 kph, is (8/3600) * (30 + N * 1.3)km. The distance traveled to stop is (8/3600) * (30 + N * 1.0)km. The total distance traveled while accelerating and decelerating is (8/3600) * (60 + N*2.3)km. If this is less than 1 km, then the distance at 'max' speed is (1 - (8/3600) * (60 + N*2.3)) km and the time it takes to transverse that distance is (1 - (8/3600) * (60 + N*2.3)) / (16/3600) s = 225 – (30 + 1.15N) = 195 – 1.15*N

Let's assume N = 20 wagons.

Algebraically,

Variable	Description	Example
n	Number of wagons	20
s_m	Max speed (km / s, not kph)	16/3600 kps
α_a	Constant Term in acceleration	6750
β_a	Wagon increment term in acceleration	292.5
α_d	Constant term in deceleration	6750
β_d	Wagon increment term in deceleration	225
a_n	Acceleration (km / s^2) with n wagons	$\frac{1\ km/s^2}{\alpha_a + \beta_a n}$
d_n	deceleration (km / s^2) with n wagons	$\frac{1\ km/s^2}{\alpha_d + \beta_d n}$
D	Total distance to move	1 km
l	Speed on ladder (km / s)	10/3600 kps
r	Distance on ladder between tracks (m, not km)	3 m

Split up the total time into 4 parts $T = t_1 + t_2 + t_3 + t_4$:

Variable	Description	Formula	Example

t_1	Time (seconds) it takes to go from 0 kph to s_m kph	$= \frac{s_m}{a_n}$	$= \frac{16/3600}{1/(6750 + 292.5n)}$ $= 30 + 1.3n = 56\ s$
t_3	Time (seconds) it takes to go from s_m kph to 0 kph	$= \frac{s_m}{d_n}$	$\frac{16/3600}{1/(6750 + 225n)}$ $= 30 + n = 50\ s$
t_2	Time (seconds) traveling at s_m kps	$= \frac{D}{s_m} - \frac{t_1}{2} - \frac{t_3}{2}$	$= \frac{1}{16/3600} - \frac{56}{2} - \frac{50}{2}$ $= 172\ s$
t_4	Time (seconds) it takes on the ladder, where the shunter is going between track i and j.	$= \lvert i - j \rvert \frac{r}{1000l}$	$= \lvert i - j \rvert \frac{3}{1000 \times 10/3600}$ $= \lvert i - j \rvert 1.08$
$t_1 + t_2 + t_3$	Time it takes to pull or push a distance of $D\ km$	$\frac{D}{s_m} + \frac{t_1}{2} + \frac{t_3}{2}$	$\frac{1}{16/3600} + \frac{56}{2} + \frac{50}{2} = 278\ s$

Some of this was pulled from:
https://www.researchgate.net/publication/348007661_Recommendations_for_the_selection_of_parameters_for_shunting_locomotives