Mixed Model Assembly Line with Bicycle Manufacturing

Etienne Batiste

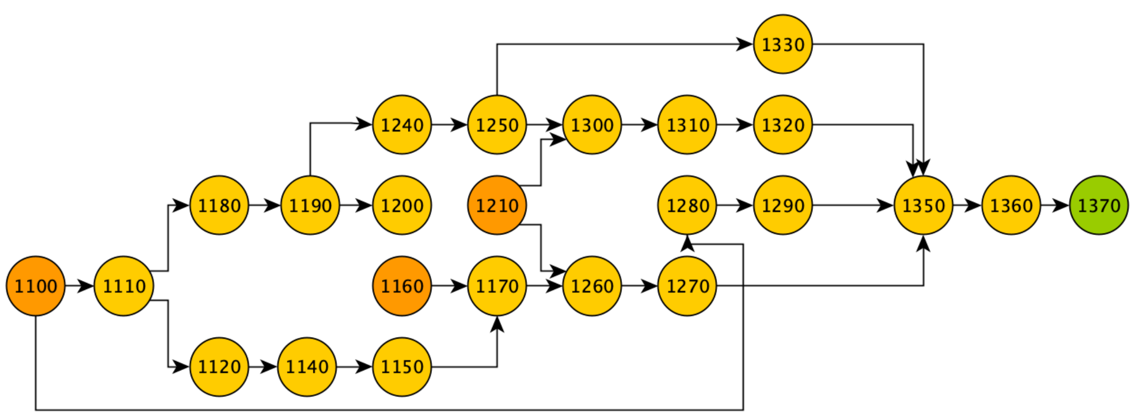
August 20, 2021

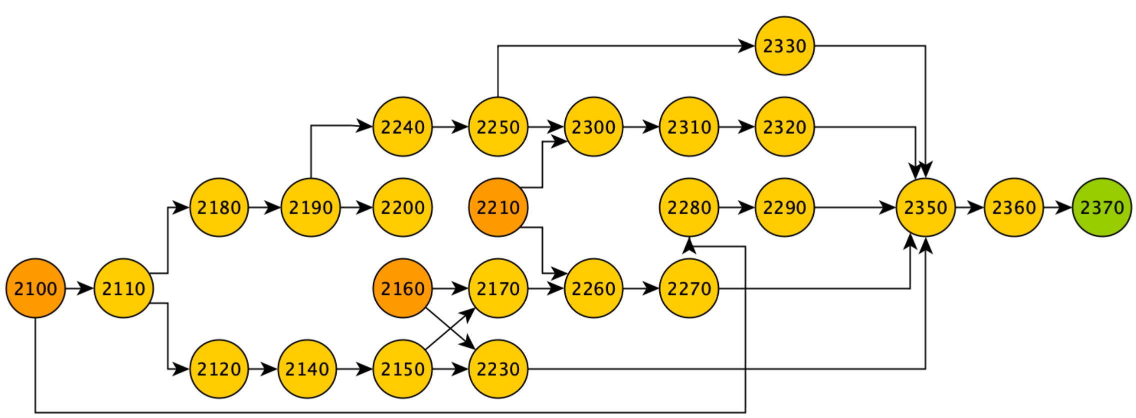
Abstract

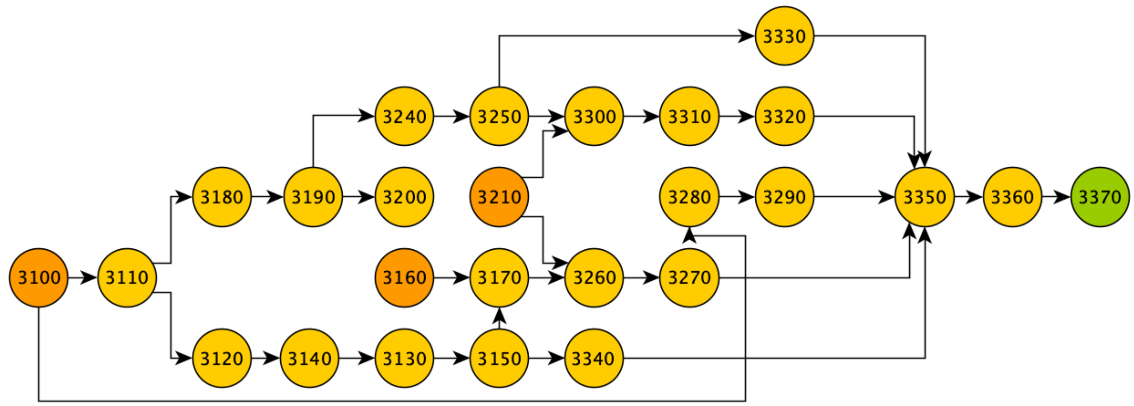
This paper discusses the theory behind the mixed model assembly optimization that was used for creating a bicycle manufacturing facility with three main bike types. There are three key components to this problem, assembly line balancing optimization, sequencing optimization and simulation. To put this theory into real results python with Gurobi Optimization was used. Gurobi is a mathematical programming solver which utilized the simplex method of optimization for this Mixed Integer Programming (MIP) problem. Our results are an optimized factory layout that can then be transferred to a simulation model where its data can be analyzed. The sequencing can be volatile and may need to be reworked later after assessing its validity with the simulation.

1 Introduction

To begin the formulation of our assembly line we must first map out our task precedence graphs for each bike type. Our precedence graphs will give a visual schedule of the tasks but will not tell where each task will be performed along the assembly line. For that we will need to balance. Each bike task starts with 1, 2 or 3 for bike types 1, 2, and 3 respectively. If the last three digits of the task number are the same that implies that these tasks are equivalent.







2 Assembly Line Balancing Problem

In this part of our problem we are looking to assign each task from each bike type to a specific station along our assembly line. Before generating this integer programming model we must calculate some ideals based on given parameters.

2.1 Calculating Our Ideals

Given Parameters:

Number of Units/day	250
Time in 12hr workday	43,200 s

- Takt Time (C): Our ideal processing time at each station

$$\frac{43,200s}{250} = 172.8s$$

- Minimum Number of Stations: Total Processing Time (TP_m) for each model m / Takt Time

$$\lceil\frac{2350}{172.8}\rceil = 14 \text{ (Race bike model)}$$

$$\lceil\frac{2490}{172.8}\rceil = 15 \text{ (Mountain bike model)}$$

$$\lceil\frac{2670}{172.8}\rceil = 16 \text{ (E bike model)}$$

2.2 Optimization for Single Model Assembly Line

2.2.1 Parameters

S := Set of all stations
T := Set of all tasks
C := Takt or Cycle time

2.2.2 Indices

j := Each station $j \in$ S
i := Each task $i \in$ T
t_i := Processing time for each task $i \in$ T
P_i:= Set of all predecessors for task $i \in$ T

2.2.3 Variables

1. Binary Decision Variable

$$x_{ij} = \begin{cases} 1; & \text{if task i is performed at station j} \\ 0; & \text{otherwise} \end{cases}$$

2.2.4 Constraints

1. To ensure each task is assigned to only one station

$$\sum_{j=1}^{S} x_{ij} = 1 \ \ \forall i \in T$$

2. To ensure the total processing time at each station is less than or equal to the Takt Time

$$\sum_{i=1}^{T} t_i \times x_{ij} \leq C \ \ \forall j \in S$$

3. To ensure all predecessors h for task i are performed at the same or a previous station as task i

$$\sum_{j=1}^{k} x_{hj} \geq x_{ik} \ \ \forall h \in P_i \ ; \ \forall i \in T$$

This is the general formulation for an optimization problem with a single model. We will use this basic problem and extend it to fit our multi model scenario.

2.3 Extending to Multi Model

2.3.1 Parameters

S := Set of all Stations
T_m := Set of all tasks for model m
M := Set of all models
C := Takt or Cycle time

2.3.2 Indices

j := Each station $j \in$ S
i := Each task $i \in T_m$
m := Each model $m \in$ M
t_{mi} := Processing time for task $i \in T_m$ for model m
P_{mi} := Set of all predecessors for task $i \in T_m$ for model m

2.3.3 Variables

1. Binary Decision Variable

$$x_{mij} = \begin{cases} 1; & \text{if model m task i is performed at station j} \\ 0; & \text{otherwise} \end{cases}$$

2. Station time for model m at station j

$$st_{mj} = \sum_{i=1}^{T_m} t_{mi} * x_{mij} \ \ \forall m \in M \ \ \forall j \in S$$

3. Station time difference from Takt time for model m

$$D_m = \sum_{j=1}^{S} |C - st_{mj}| \ \ \forall m \in M$$

2.3.4 Constraints

1. To ensure each task is assigned to only one station

$$\sum_{j=1}^{S} x_{mij} = 1 \ \ \forall i \in T \ \ \forall m \in M$$

2. To ensure all predecessors h for task i are performed at the same or a previous station as task i

$$\sum_{j=1}^{k} x_{mhj} \geq x_{mik} \ \ \forall h \in P_{mi} \ ; \ \forall i \in T \ \ \forall m \in M$$

3. If tasks are the same for each bike type, ensure these tasks are at the same station.

$$x_{1ij} = x_{2ij} = x_{3ij} \ \forall i \in \{\text{tasks in common}\} \ \ \forall j \in S$$

2.3.5 Objective Function

$$min \sum_{m=1}^{M} D_m$$

2.3.6 Other Options for Objective Functions

Minimizing the number of stations:

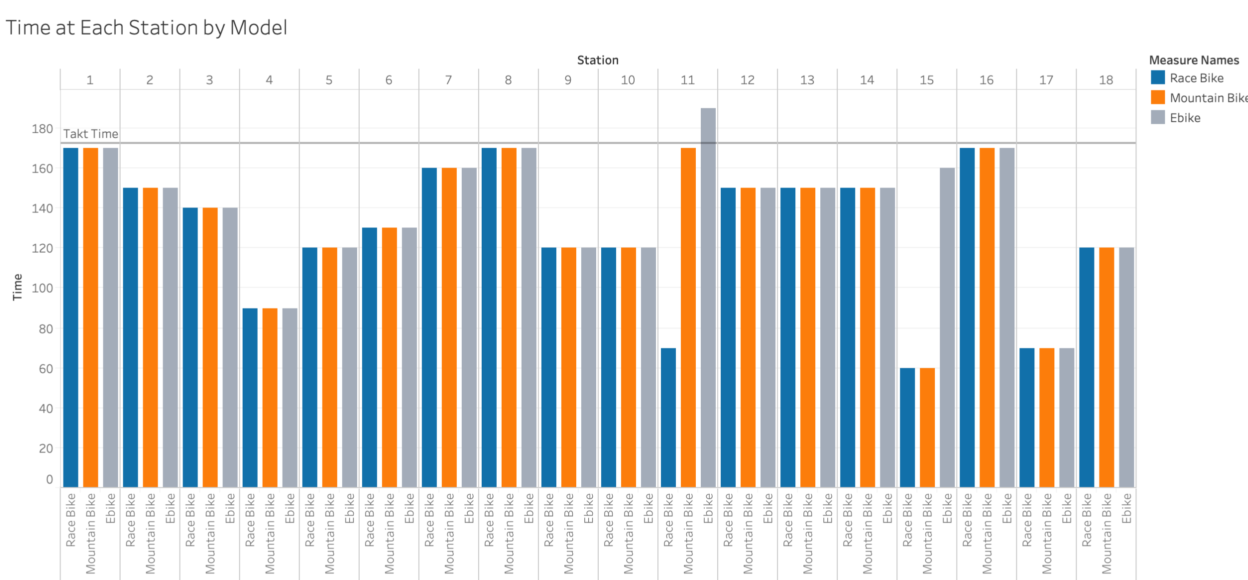
$$min \ \{numStations\}$$

Minimizing the cycle time:

$$min \ \frac{\sum_{j=1}^{S} \frac{st_{1j}}{S} + \frac{st_{2j}}{S} + \frac{st_{3j}}{S}}{M}$$

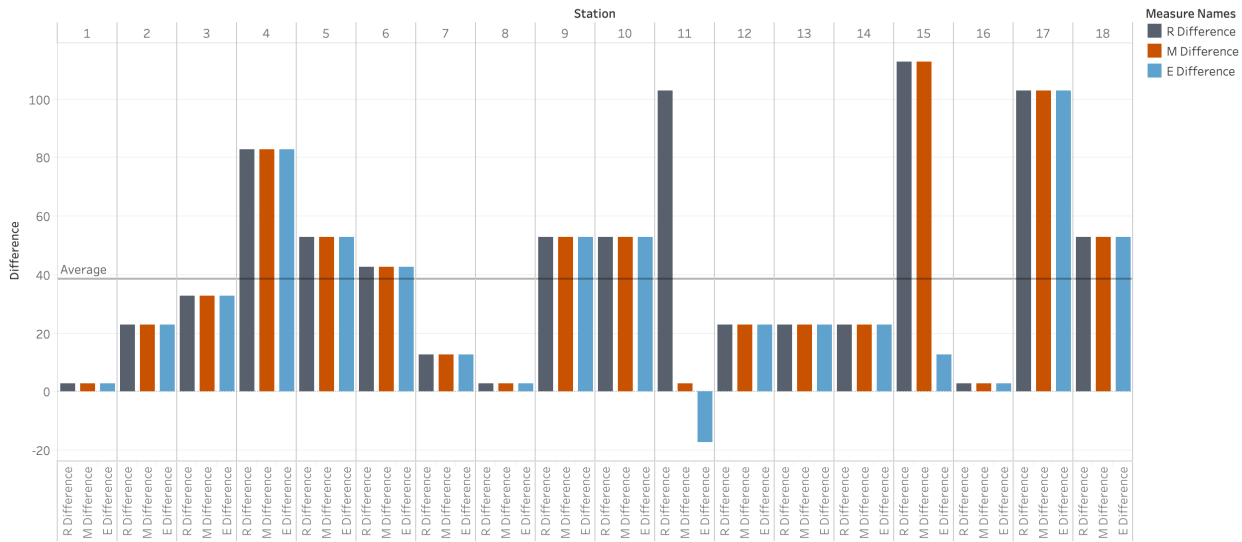
2.4 Optimal Result Analysis

To see which tasks are placed at which station please see the Task Relations Sheet in the ALLBIKE-DATA excel document.



Race Bike, Mountain Bike and Ebike for each Station. Color shows details about Race Bike, Mountain Bike and Ebike. The view is filtered on Station, which excludes Null.

Difference from Takt Time



E Difference, M Difference and R Difference for each Station. Color shows details about E Difference, M Difference and R Difference. The view is filtered on Station, which excludes Null.

3 Sequencing Optimization

In this part of our problem, our objective is to level the time between each bike in the sequencing order of the daily production. Sequencing is necessary to produce a lean manufacturing system that eliminates waste. In this scenario we should ultimately have a sequence that minimizes idle time at each station.

3.1 Parameters

S := Set of all Stations
M := Set of all models
C := Takt or Cycle time
P := Set of all positions in the sequence

3.2 Indices

s := Each station $s \in$ S
p := Each position $p \in P$
m := Each model $m \in$ M
d_m := Demand for each model

3.3 Variables

1. Binary Decision Variable

$$x_{mp} = \begin{cases} 1; & \text{if model m is placed in at position p} \\ 0; & \text{otherwise} \end{cases}$$

2. Idle time for model m at station s in position p

$$it_{smp} = (C - st_{sm}) * x_{mp} \quad \forall m \in M \quad \forall s \in S \quad \forall p \in P$$

3.4 Constraints

1. To ensure every position is only assigned one of the models

$$\sum_{m=1}^{M} x_{mp} = 1 \ \ \forall p \in P$$

2. To ensure demand is met for each bike type

$$\sum_{p=1}^{P} x_{mp} = d_m \ \ \forall m \in M$$

3.5 Objective Function

$$min \sum_{p=2}^{P} \sum_{s=1}^{S} |(it_{s1p} + it_{s2p} + it_{s3p}) + (it_{s1p-1} + it_{s2p-1} + it_{s3p-1})|$$

This objective function is looking at the idle (or overload) time at each station for the bicycle at a position and the position before it. If the bike has idle time it will be positive, if the station is over takt time it'll be a negative value. So in essence we want the smallest number when these two values are added. For example, if bike 1 is in position with a positive idle time of 100, the next bike in the line (at this same station) has three options; +100s idle time if bike 1 is selected next, +50s idle time if bike 2 is chosen, or -17s overload time if bike 3 is chosen. With this objective function you will get the values 100+100=200s, 100+50=150s, or 100-17=83s and the algorithm will choose to place bike type 3 because we have selected to minimize this function.

3.6 Optimal Result

To see the final optimal sequence please see the sequence1-optimal sheet in the Sequencing1 excel file along with two other sequences to create variation in the model.

4 Simulation

After optimizing the sequence for the given balanced assembly line we would like to next simulate the factory. After the simulation set up is complete we need to find a few measures that will assess the viability of our optimal sequence against others.

4.1 Measuring Variance Between Models

1. Station Utilization:
   Compare station utilization's for each station

2. Cycle Time:
   Measure cycle time of each bike coming off the assembly line Take average as Output Data

3. Idle Time:
   Measure the idle time at each station or the difference between Takt time