**Optimization report**

In this report, we look at how to improve a factory process that makes semiconductor wafers. Our goal is to make 1000 wafers as quickly as we can. Check out the documentation for the python simulator to understand how it works and the rules it must follow.

We have investigated three main parameters to improve the production performance:

1. The way the 1000 wafers are grouped into batches.
2. The times at which the batches are loaded into the input buffer of the production line.
3. The heuristic used by each unit to select the next batch to process.

Our approach for examining the three parameters involved holding all but one constant, and then observing how performance shifted as the variable changed. In this report, we will present the experiments conducted for each parameter, detailing our attempts and findings. Ultimately, we will share our conclusions regarding the optimal parameter settings we discovered.

**Parameter A: The way the 1000 wafers are grouped into batches**

1. We divide the 1000 wafers into equal batches. If 1000 isn't divisible by the batch size, we create an extra batch for the remaining wafers if it is over or equal to 20 left, or spread the extra wafers among the batches if it is less than 20 left. The graph under shows the different time used on the different batch sizes. With this experiment we can conclude that the best batch size if every batch is going to be approximately the same size is 36. The simulation time we used the batch size 36 was 5657.8.

Chart, line chart

Description automatically generated

1. In this experiment we tried to brute force our way to a better result than we achieved in the first experiment. We created a function that generates a list of batches with random sizes that adds up to 1000. This function is used in our brute-force function that runs as many iterations as we wanted and saves the best result in a csv file. We managed to get a simulation time at 5612.9 after approximately 100k iterations. We that the combinations of initial batches is way too big to use this method, but wanted to try our luck anyways.
2. In this experiment we wanted to analyze our best time to see if we saw any pattern. We plotted every batch and their sizes to analyze it.

Chart, line chart

Description automatically generated

**Parameter B: The times at which the batches are loaded into the input buffer of the production line**

TODO

**Parameter C: The heuristic used by each unit to select the next batch to process**

1. We prioritized tasks in the order they appear in the production line, giving higher priority to earlier tasks.
2. We prioritized tasks based on their processing time, giving higher priority to tasks with longer processing times.
3. We prioritized tasks based on their processing time, giving higher priority to tasks with shorter processing times.
4. We used a dynamic approach, prioritizing tasks based on how full their input buffer was, always choosing the fullest one.

Here is the result for the different heuristics tested. The settings are 1000 wafers split in 20 batches with 50 wafers each.

|  |  |
| --- | --- |
| Heuristic method | Time used |
| 1 | 6038.0 |
| 2 | 6220.0 |
| 3 | 5524.0 |
| 4 | 6152.0 |