**Executive Summary**

**Notes (to delete this)**

**-Why we need such a tool  
-complexity and manufacturing scheduling challenge  
-what type of scheduling  
-what we want to achieve**

Today’s economy revers competitiveness, agility and sustainability. With key performance indicators of time-to-market and customer responsiveness, many manufacturing companies are adopting Industry 4.0 standards to optimize productivity, efficiency and flexibility via digital manufacturing solutions. While most of the manufacturing systems are operating near optimal productivity under stable conditions, certain disturbances such as product changes, dynamic work-in-progress (WIP) and urgent orders, which are inevitably present in the real product environment, can impact the overall manufacturing performance. Such production disturbances can result in an estimated 80% of overall equipment effectiveness (OEE) loss. One of such solutions is providing real-time end-to-end planning and control via an effective scheduling system that aims to handle and optimize the dynamic production environment, thereby reducing cycle time, increase machine utilization and throughput, meet due date and eventually achieving greater customer satisfaction and trust.

The semiconductor backend testing is one of the most highly complex manufacturing systems in terms of equipment, process flows and interdependency relations. Furthermore, the complexity is increased due to a large product variety, complex product to tester relations, multiple level of tester to hardware dependency, sequence dependency, dynamic determination of processing and indexing time, batch processing and rework flow. Besides its complexity, the real production environment is also subjected to many uncertainties and unpredictable events due to continuous arrival of new and unforeseen orders and intermittent occurrence of machine breakdown, and random process and yield variations. Thus, the scheduling is no longer simply a static optimization problem, but an ongoing reactive process. Therefore, the scheduling operations and optimization in semiconductor backend testing are highly demanding and challenging for both researchers and practitioners.

Generally, the basic scheduling approaches formulate schedules computed over a specific time frame assuming all problems and states are known in advance, without consideration of any disturbances in the actual production. In order to handle the dynamic production environment, short-term planning and scheduling can be considered. Schedules can be regenerated based on new events or limitations such as machine breakdown and WIP shortages or urgent order, respectively. A periodic scheduling regenerates periodically over a time period to update the schedules with new job list (based on existing WIP status) while event-driven regenerates schedules upon event occurrence.

Based on the above considerations given to the different facets in the scheduling process, this project aims to introduce a real-time manufacturing scheduler that is capable of helping the production accomplish operation strategies, optimize goals and deal with daily disruptions such as machine breakdown and WIP discontinuity in a more effective way. In the next section, the project objectives and scope are defined, followed by a detailed discussion of the problem definition, requirements and constraints.

**1.2 Project Objectives and Scope**

In this project, a scheduling problem based on a semiconductor backend test case study is performed via Optaplanner. It is a programmable platform that functions as a constraint satisfaction solver that optimizes business resource planning cases such as scheduling problem in the nature of this project.

The case study mainly focusses on the particular process of the test manufacturing whereby the integrated circuits (ICs) will be placed in an oven at desired high temperature for prolonged hours (~6-24 hours) to remove any moisture content. This process is also known as the bake-in operation and the lots are processed in the ovens in batches. Therefore, this project aims to schedule and optimize the semiconductor test operation by selecting the most optimal lot to load onto the available oven in future simulated time to:

1. Meet due date (improving delivery accuracy; processing the right products and quantity to fulfill delivery demand)
2. Minimize cycle time (time taken for lot to complete the bake-in operation)

The scheduling problem is simplified by making the following assumptions:

* Setup time and breakdown of oven and lot movement time in the operation (i.e. lot moving into and out of oven) are negligible.

1. **Problem statement and formulation**

The scheduling problem is defined as the assignment of selected lots from the WIP list to be processed onmachines in eachoperation while satisfying constraints and optimizing objective function. The criteria of the lot selection process from the existing WIP is dependent on the delivery plan provided (i.e. delivery accuracy, selecting the right lots to be processed to fulfill delivery demand) (see Appendix?) and its objective function is as follows

Minimize D(s): (1)

where *i* refers to the lot identification (ID); *j* refers to the product type; *q* refers to the lot quantity size and *Q* refers to the delivery demand. After the lot selection is performed, the scheduling is carried out with the following requirement and constraints:

- Each lot is composed of only one product

- During the bake-in operation, lots of the same product can be processed in the same oven(batch processing)

- However, only some products may be processed in certain ovens because of product requirement

* Ovens are independent from each other.
* All lots and ovens are available at initial state.
* The processing time at each oven is product and oven dependent.
* At a given time, the oven can only execute one operation.
* Processing at each oven must be completed without interruption once it starts
* The oven has a limited capacity.

The scheduling problem sets include the set of selected lots *I* to be processed, with each lot *i* corresponding to a specific product *j* and a set of ovens *L* with specific temperatures *k*. The attributes of the sets are shown in Table 1. The planning parameters are also specified in Table 2 which includes the lot quantity *qij*, processing time *pk* and oven quantity of different temperatures *rk*. The oven machines have different operating temperatures *k*, mainly 125 and 150 degree Celsius and each product *j* has different processing time *pjk*. Once the batch processing starts, the process cannot be interrupted. All the product processing requirements are shown in Appendix A. The variable of the scheduling problem will be defined as *Sni* where *S* denotes the scheduling solution; *n* is the ordinal value that measures the order of lot *I* sequence*.* (not sure if this is correct, can we check with changhe).

In summary, the scheduling problem considered in this project is a single-stage, parallel batch multi-product job scheduling problem. There are more than 100 different types of products, where same product units are grouped in lot form and assigned to oven in batches for processing.

**Table 1. Definition of problem sets**

|  |  |  |
| --- | --- | --- |
| **No** | **Problem sets** | **Attributes** |
| 1 | *Iij*: Set of selected lots that are scheduled for processing. (each lot corresponding to a specific product) | *i* = Lot ID *j* = Product Type  *qij* = Quantity |
| 2 | *Llk:*: Set of oven units available for processing the lots | *l* = Oven ID *k* = Oven Temperature Type *rk* = Oven quantity |

**Table 2. Planning parameters**

|  |  |
| --- | --- |
| **Planning Parameters** | **Definition** |
| *qij* | Total number of units within a lot *i* (each lot corresponding to a specific product *k*) |
| *pjk* | Processing time of each product *j* at specific operating temperature *k* |
| *rk* | Total number of oven units of specific operating temperature *k* |

**Table 3. Oven inventory in the factory.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operations** | **Machine** | **Quantity *rk*** | **Temperature Type *k*** |
| Bake-in | Oven | 15 | 125 |
| 7 | 150 |

The appropriate sequencing of lots on the oven in the bake-in operations can be determined via the optimization of the objective function which is derived based on the economic factory performance. The measure of the economic factory performance encompasses of timely delivery fulfillment (minimum makespan or cycle time) and maximum equipment utilization. A set of weightages *W* are considered to provide prioritization to the two performance measures as in usual practice, these measures often contradict each other. Herein, the objective function is:

(2)

where the cycle time is measured by averaging the total processing time of all the batch jobs in all ovens.

Conclusion

Complexity of real situation – now we are doing offline scheduling. Why is it not good enough. We need onling reactive scheduling

ORIGINAL TEXT

Introduction

As time-to-market and customer responsiveness are today’s key factors of competitiveness, agility and sustainability, many manufacturing companies are driving towards Industry 4.0 to boost productivity, efficiency and flexibility through digital manufacturing solutions. Today, most of the manufacturing systems are operating at near optimal productivity under stable conditions. However, disturbances such as product changes, dynamic work-in-progress (WIP) and urgent orders that are inevitably present in the real production environment can affect the overall manufacturing performance. It is estimated that such production disturbances can results in ~ 80% of overall equipment effectiveness (OEE) loss (Ylipaa, 2002). One of such solutions is providing real-time end-to-end planning and control via an effective scheduling system that aims to handle and optimize the dynamic production environment, thereby reducing cycle time, increase machine utilization and throughput, meet due date and eventually achieving greater customer satisfaction and trust.

The semiconductor backend testing is one of the highly complex manufacturing systems in terms of equipment, process flows and interdependency relations. Furthermore, the complexity is increased due to a large product variety, complex product to tester relations, multiple level of tester to hardware dependency, sequence dependency, dynamic determination of processing and indexing time, batch processing and rework flow. Besides its complexity, the real production environment is also subjected to many uncertainties and unpredictable events due to continuous arrival of new and unforeseen orders and intermittent occurrence of machine breakdown, and random process and yield variations Thus, the scheduling is no longer a static optimization problem, but an ongoing reactive process. Therefore, the scheduling operations and optimization in semiconductor backend testing are highly demanding and challenging for both researchers and practitioners.

Conventionally, the scheduling approaches formulate schedules computed over a specific time frame assuming all problems and states are known in advance, without consideration of any disturbances in the actual production. As such, these conventional approaches are inadequate to handle the manufacturing characteristics, and thus it is necessary to adopt a predictive-reactive scheduling which is more effective in addressing today’s manufacturing scheduling challenges. The predictive-reactive scheduling is a two stage approach whereby a schedule is first generated with predicted disruptions and subsequently updated in response to real disruptions, such as machine breakdown events or WIP shortages. Updating of schedules involves regeneration of schedules based on the new limitations and can be performed whenever necessary triggered periodically or event-driven. A periodic scheduling regenerates periodically over a time period to update the schedules with new job list (WIP status) while event-driven regenerates schedules upon event occurrence. Such predictive scheduling is key for planning support activities such as planning for machine maintenance and material procurement and also helps to improve performance compared with fixed sequencing and dispatching procedures.

Therefore, this project aims to introduce a real-time manufacturing scheduler that is capable of helping the production accomplish operation strategies, optimize goals and deal with daily disruptions such as machine breakdown and WIP discontinuity in a more effective way. In the next section, the project objectives and scope are defined, followed by a detailed discussion of the problem definition, requirements and constraints

1. **Introduction**
   1. **Background**

As time-to-market and customer responsiveness are today’s key factors of competitiveness, agility and sustainability, many manufacturing companies are driving towards Industrie 4.0 to boost productivity, efficiency and flexibility through digital manufacturing solutions. Today, most of the manufacturing systems are operating at near optimal productivity under stable conditions. However, disturbances such as product changes, dynamic work-in-progress (WIP) and urgent orders that are inevitably present in the real production environment can affect the overall manufacturing performance. It is estimated that such production disturbances can results in ~ 80% of overall equipment effectiveness (OEE) loss (Ylipaa, 2002). One of such solutions is providing real-time end-to-end planning and control via an effective scheduling system that aims to handle and optimize the dynamic production environment, thereby reducing cycle time, increase machine utilization and throughput, meet due date and eventually achieving greater customer satisfaction and trust.

The semiconductor backend testing is one of the highly complex manufacturing systems in terms of equipment, process flows and interdependency relations. Furthermore, the complexity is increased due to a large product variety, complex product to tester relations, multiple level of tester to hardware dependency, sequence dependency, dynamic determination of processing and indexing time, batch processing and rework flow. Besides its complexity, the real production environment is also subjected to many uncertainties and unpredictable events due to continuous arrival of new and unforeseen orders and intermittent occurrence of machine breakdown, and random process and yield variations Thus, the scheduling is no longer a static optimization problem, but an ongoing reactive process. Therefore, the scheduling operations and optimization in semiconductor backend testing are highly demanding and challenging for both researchers and practitioners.

Conventionally, the scheduling approaches formulate schedules computed over a specific time frame assuming all problems and states are known in advance, without consideration of any disturbances in the actual production. As such, these conventional approaches are inadequate to handle the manufacturing characteristics, and thus it is necessary to adopt a predictive-reactive scheduling which is more effective in addressing today’s manufacturing scheduling challenges. The predictive-reactive scheduling is a two stage approach whereby a schedule is first generated with predicted disruptions and subsequently updated in response to real disruptions, such as machine breakdown events or WIP shortages. Updating of schedules involves regeneration of schedules based on the new limitations and can be performed whenever necessary triggered periodically or event-driven. A periodic scheduling regenerates periodically over a time period to update the schedules with new job list (WIP status) while event-driven regenerates schedules upon event occurrence. Such predictive scheduling is key for planning support activities such as planning for machine maintenance and material procurement and also helps to improve performance compared with fixed sequencing and dispatching procedures.

Therefore, this project aims to introduce a real-time manufacturing scheduler that is capable of helping the production accomplish operation strategies, optimize goals and deal with daily disruptions such as machine breakdown and WIP discontinuity in a more effective way. In the next section, the project objectives and scope are defined, followed by a detailed discussion of the problem definition, requirements and constraints

**1.2 Project Objectives and Scope**

In this project, a scheduling problem based on a semiconductor backend test case study is performed using Optaplanner. The case study mainly focusses on the last stage of the test manufacturing process whereby the integrated circuits (ICs) grouped by lots undergo two-step operations, mainly the bake-in and scanning operations. During the bake-in operation, the ICs will be placed in an oven at desired high temperature for prolong hours (~6-24 hours) to remove any moisture content before undergoing the scanning process to inspect for mechanical defects and accurate markings. Subsequently, the ICs are packed in the reels to be shipped out to the customers. Therefore, the project aims to schedule and optimize the semiconductor test operation by choosing the most optimal lot to load onto the available equipment in future simulated time to:

1. Meet due date (improving delivery accuracy; processing the correct lots to meet delivery orders)
2. Minimize cycle time (time taken for lot to complete the two operations)
3. Maximize machine utilization (for maximum productivity)

The scheduling problem is simplified by making the following assumptions:

* Setup time of machines and lot movement time between operations are negligible.
* For scanning process, there is no product and machine dependency (i.e. all products can be processed in all machines)

1. **Problem definition and requirements**

The scheduling problem is defined as the assignment of selected *n* lots from the WIP list to be processed on *m* machines eachoperation while satisfying precedence constraints and optimizing objective function. There is a set of selected N = (N1, N2, …, Nn) in this problem where each Nn of product type k contains qn number of IC units product type. Based on the delivery order requirement which is a set of D = (D1, D2, …, Dk) where Dk contains the Qk delivery quantity of each k product, N lots will be selected from the WIP list for delivery fulfilment which is defined by,

(1)

After the delivery requirement is fulfilled and selected N set determined, the next requirement is the lot assignment to first bake-in operation which is a batch processing in the ovens and then to the Automated Optical Inspection (AOI) machines for scanning operation. The machine inventory list is shown in Table 1. The oven machines have different operating temperatures, mainly 125 and 150 degree Celsius and each product type has different bake-in time. Once the batch processing starts, the process cannot be interrupted. For the scanning process, it is a unit processing and each product type has its individual processing time. Each *n* lot can only be processed only on one predetermined machine at each operation. All the product processing requirements are shown in Appendix A.

With the use of Optaplanner, the appropriate sequencing of lots on the machines in the two operations can be determined via the optimization of the objective function. Herein, the objective function is:

Objective function f(s) = w1(1/CT(s))+w2Um(s)

Where CT(s) = sum of Tn, Um = sum of Tn/time period and w1 and w2 are the importance/scaling of the performance indicator. Where Tn is the time taken for the lot complete the operations. Time period is the delivery time frame

Other conditions and constraints are defined as follow:

* Machines are independent from each other.
* Lots must undergo bake-in process followed by the scanning process.
* All jobs and machines are available at time 0
* The processing time at each machine is product and machine dependent.
* At a given time, a machine can only execute one operation.
* Processing at each operation must be completed without interruption once it starts

Table 1. Machine inventory in the factory.

|  |  |  |  |
| --- | --- | --- | --- |
| Operations | Machine | Quantity | Type |
| Bake-in | Oven | 15 | 125 |
| 7 | 150 |
| Scanning | Automated Optical Inspection (AOI) | 36 | - |

1. **The Solution/Approach**

**System Architecture**

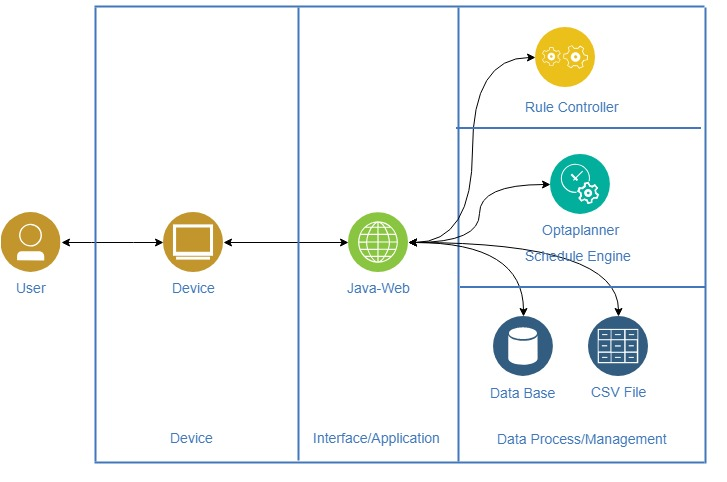


Figure 1. Flow of the effective scheduling system

This system is primarily accessed via a frontend webpage hosted on apache tomcat locally (at this point). The frontend will take information from CSV and Database to feed to the optaplanner solver and the Rule engine that does the packaging.

**System Flow**

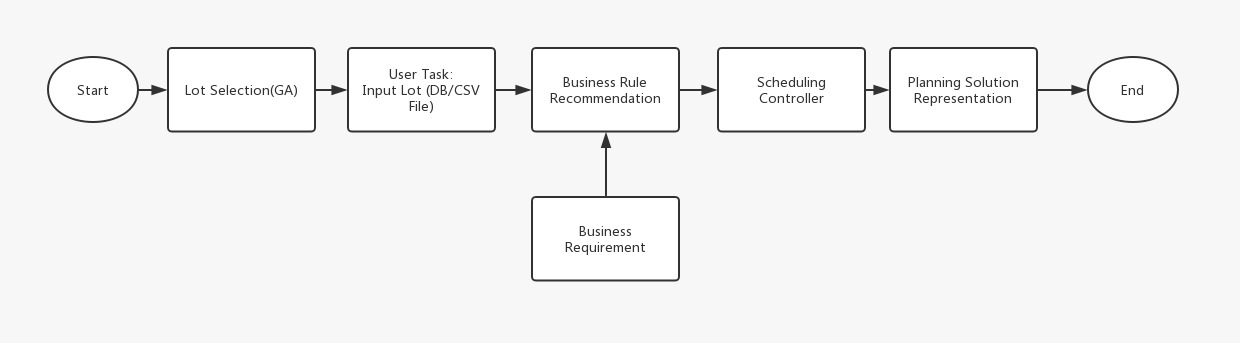


Figure 2. Flow of the effective scheduling system

Figure 2 illustrates the flow of this project. This will include:

* The Lot selector (Done by Genetic Algorithm)
* The File Upload
* Packaging Rule
* Scheduling
* Displaying Scheduled Results

**Lot Selection**

With the lot selection and scheduling problem requirement, constraints and objective functions defined in the previous section, the lot selection process is first performed via Excel Solver and subsequently, the selected lot list is passed to the effective scheduling system which is developed via Optaplanner. The Excel Solver based on the evolutionary algorithm is employed to solve the lot selection problem which is basic binary assignment problem.  When a lot is selected, a value of 1 will be assigned while an unselected lot will assign a value of 0. As the selection list is relatively large, a population size of 50, mutation rate of 0.075 and convergence of 0.0001 are set to reduce the computational time.

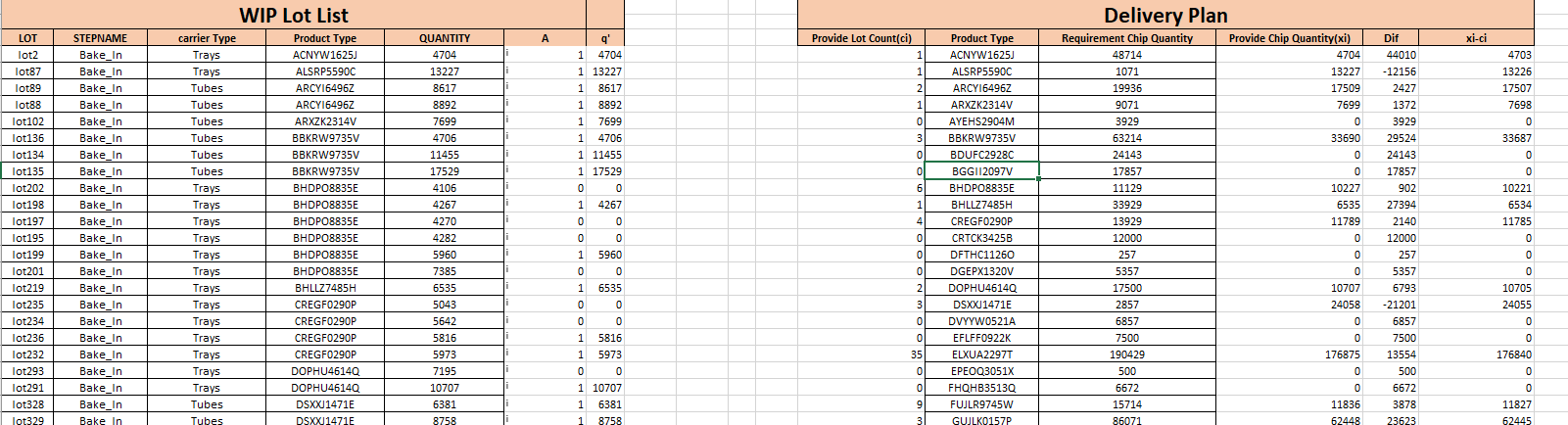
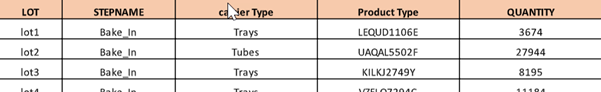


Table 1 : Excel GA Solver

**User task: Input Lot (DB/CSV File)**

From the user end, the process requires the user to upload a input lot file from the database in the format of a CSV (comma separated values) file. The file has headers in the following format as shown in Table 2 below. The CSV format provides a straightforward information schema in a compact, readable manner that allows any editing manual or otherwise and efficient processing by the scheduling controller based on the business rules. CSV format also allows easy sorting, filtering and manipulation based on any programmable queries.

Table 2 : Screenshot of Input file in CSV format



The csv file will contain certain crucial information we will need to schedule baking in the ovens:

* Lot ID
* Process Type
* Product Type
* Quantity

In the next process, we will use these user-fed details to generate a input file to feed to the solver.

**Packaging Rule**

To make use of the csv file, we must define a few rules. In order to utilize our resource (ovens) we need to know how many ovens we have, and more information on the product. This information will be maintained in a database table in mySQL.

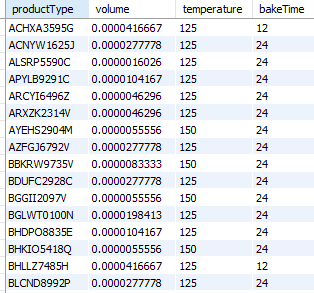
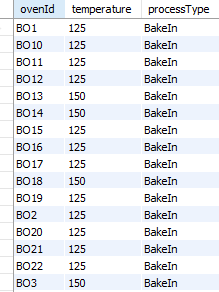
 

Figure 3: Product Information DB Table Figure 4: Oven Information DB Table

In Figure 3, for every product type, we have baking temperature and time requirement. Also, we have the “volume” variable, which stands for how much one piece of product will occupy the oven in %, out of 100 percent, given that all oven has the same capacity. The rule engine will take information provided in the csv (Product Type and Quantity), and cross reference with the DB tables. Then, on a first in first out (FIFO) basis, it will add cumulative volume percentage, and register one package when the next member of the stack will cause the 100% to overflow. At the end of this process, we will only have data in full packages, ready to be fed into the oven.

**Scheduling Controller (Solver)**

Subsequently, the production planner picks up the resulting package list and input it into solver. In this project, the Optaplanner solver engine is used. With proper class parameters (Figure 5) and hard/soft constraints set, these values will be fed to Optaplanner solver to perform a local search for the best solution.

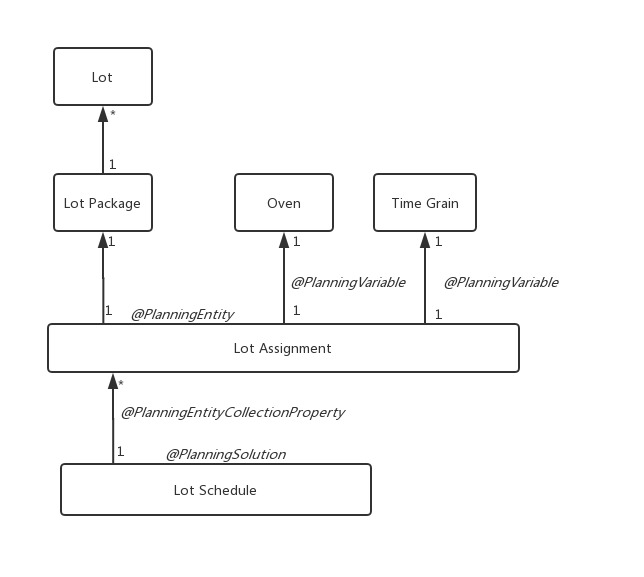


Figure 5: Class Diagram

The objective for this process is the lot scheduling calendar, displayed on a webpage. To fulfil it, we have to set certain hard and soft constraints so that the Optaplanner can perform local search properly. Our hard constraint will be the capacity of the oven, because it cannot be violated. For any instance of time involved, in hours, it will be used as soft constraint.

In this process, we are not setting a algorithm for Optaplanner to use. As such, it will use the default search algorithm (Late Acceptance). This will be for minimizing the amount of idle time in between processes.

After doing the local search, Optaplanner will return the data in time-start and time-end format, to be displayed in a chart.

**Planning Solution Representation**

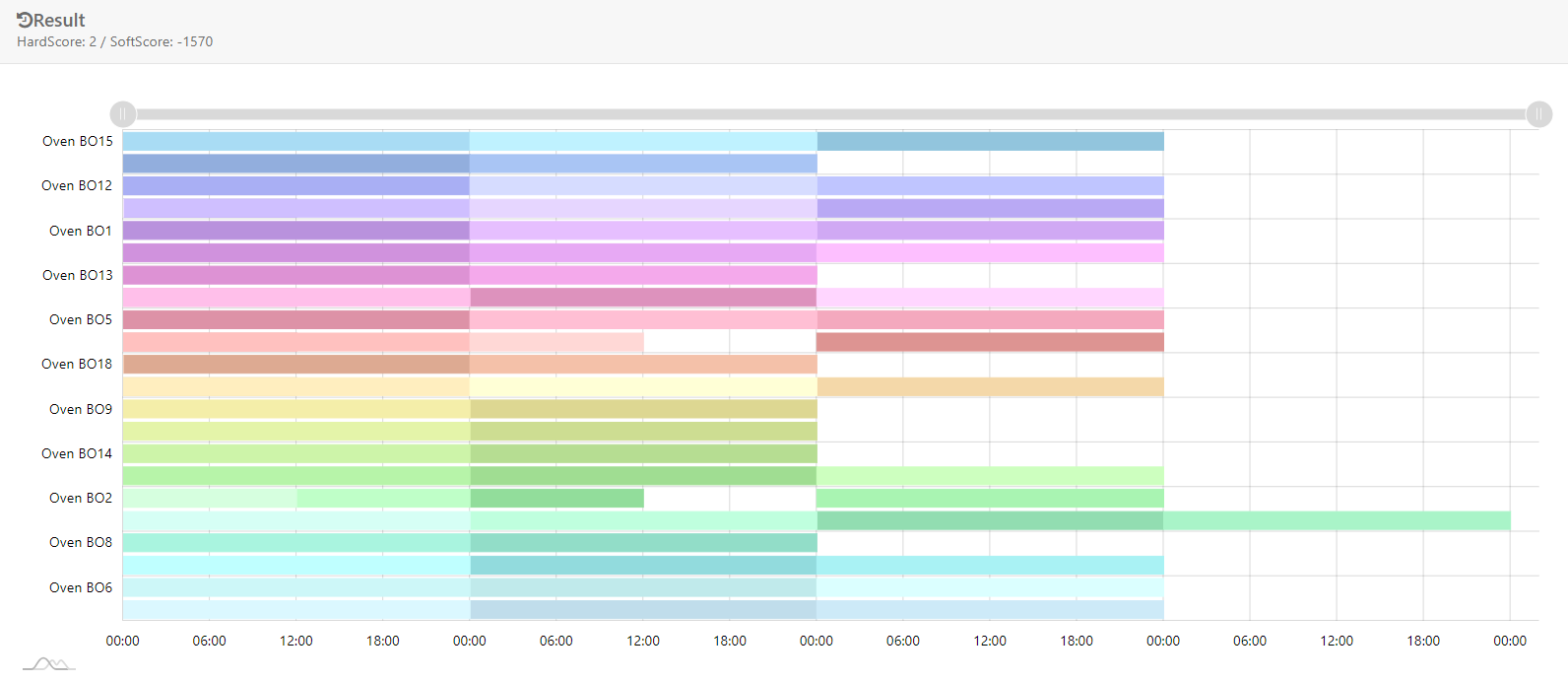


Figure 6: Class Diagram

After processing, the frontend will take information from the previous process to display the above chart (Figure 6). Engineers can then schedule operators/robots to put the parts in the ovens.

**To delete**

1. **Conclusion**

Appendix