MASTER OF TECHNOLOGY (INTELLIGENT SYSTEMS)

PROJECT REPORT

Reasoning Systems

SDOT

Scheduling & Despatch Optimisation Tool

TEAM MEMBERS OF HEX

YE CHANGHE
ZHANG HAIHAN
PAMELA LIN YAN LING
CH'NG WEI LUEN
LIM LI WEI
PREM s/o PIRAPALA CHANDRAN

Contents

EXECUTIVE SUMMARY

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, there are certain disturbances such as product changes, dynamic work-in-progress (WIP) and urgent orders, which are inevitably present in the real product environment which can impact the overall manufacturing performance. One possible solutions is providing real-time end-to-end planning and control via an effective scheduling system that aims to handle and provide for the dynamic production environment.

In this project, our group has proposed an integrated solution that is able to address the planning and control issues for the scenario of scheduling in baking of IC (integrated circuit) chips in the semiconductor industry. We implement and propose a solution we call SDOT (Scheduling & Despatch Optimisation Toolkit) and discuss conceivable limitations of the system and suggest possible enhancements that can modify it to handle more complexity in scheduling problems.

1.0) PROBLEM DESCRIPTION

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.

1.1) PROJECT OBJECTIVES & 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.

1.2) PROBLEM STATEMENT AND FORMULATION

The scheduling problem is defined as the assignment of selected lots from the WIP list to be processed on machines in each operation 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 A) and its objective function is as follows

$$\text{Minimize D(s): } \left| \sum_{\substack{i=1 \\ j=1}} q_{i,j} - Q_j \right| \tag{1}$$

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 q_{ij}, processing time p_k and oven quantity of different temperatures r_k. The oven machines have different operating temperatures *k*, mainly 125 and 150 degree Celsius and each product *j* has different processing time p_{jk}. Once the batch processing starts, the process cannot be interrupted. All the product processing requirements are shown in Appendix B. The variable of the scheduling problem will be defined as S_{ni} where *S* denotes the scheduling solution; *n* is the ordinal value that measures the order of lot *I* sequence.

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	I_{ij}: Set of selected lots that are scheduled for processing. (each lot corresponding to a specific product)	i=Lot ID j = Product Type q_{ij} = Quantity
2	L_{lk}:: Set of oven units available for processing the lots	l=Oven ID k = Oven Temperature Type r_k = Oven quantity

Table 2. Planning parameters

Planning Parameters	Definition
q_{ij}	Total number of units within a lot i (each lot corresponding to a specific product k)
p_{jk}	Processing time of each product j at specific operating temperature k
r_k	Total number of oven units of specific operating temperature k

Table 3. Oven inventory in the factory

Operations	Machine	Quantity r_k	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 $f(s)$ is:

$$CT(s) = \frac{\sum_{l=1} \sum_{\substack{n=1 \\ i=1}} t_{ni}}{\sum_{k=1} r_k} \tag{2}$$

where the cycle time $CT(s)$ is measured by averaging the total processing time of all the batch jobs in all ovens.

2.0) APPROACH AND SOLUTION

2.1) SYSTEM ARCHITECTURE

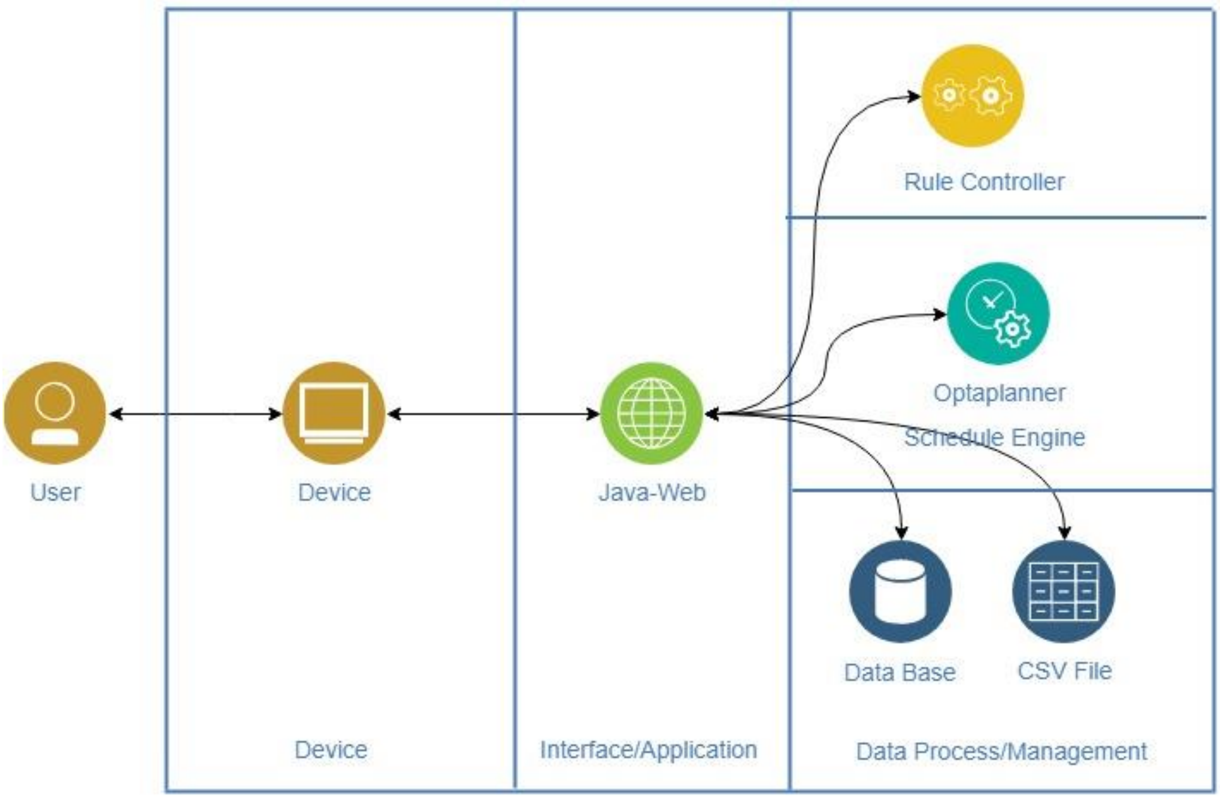

Figure 1. Flow of the effective scheduling system

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.

2.2) SYSTEM FLOW

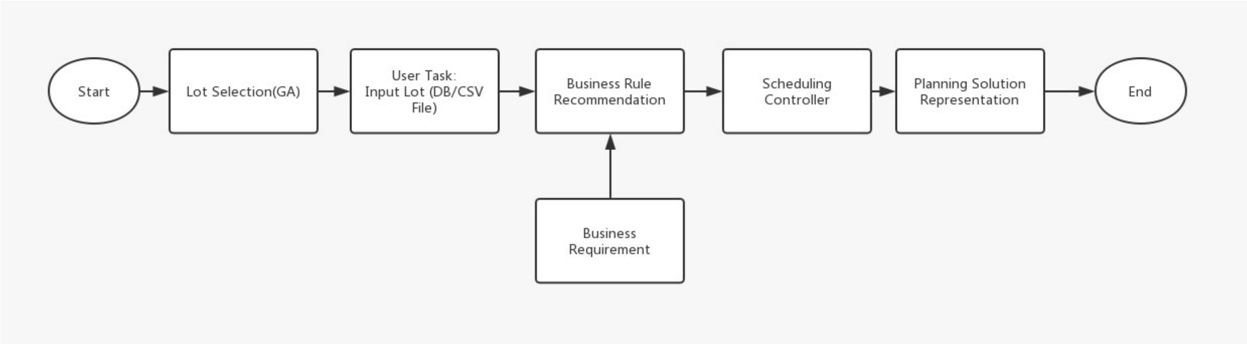

Figure 2. Flow of the effective scheduling system

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

Each of these components in the flow is explained below to provide an understanding of how they are integrated for the solution to work.

Lot Selection (done by Genetic Algorithm)

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 a 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.

WIP Lot List						
LOT	STEPNAME	carrier Type	Product Type	QUANTITY	A	q'
lot2	Bake_In	Trays	ACNYW1625J	4704	1	4704
lot87	Bake_In	Trays	ALSRP5590C	13227	1	13227
lot89	Bake_In	Tubes	ARCYI6496Z	8617	1	8617
lot88	Bake_In	Tubes	ARCYI6496Z	8892	1	8892
lot102	Bake_In	Tubes	ARXZK2314V	7699	1	7699
lot136	Bake_In	Tubes	BBKRW9735V	4706	1	4706
lot134	Bake_In	Tubes	BBKRW9735V	11455	1	11455
lot135	Bake_In	Tubes	BBKRW9735V	17529	1	17529
lot202	Bake_In	Trays	BHDPO8835E	4106	0	0
lot198	Bake_In	Trays	BHDPO8835E	4267	1	4267
lot197	Bake_In	Trays	BHDPO8835E	4270	0	0
lot195	Bake_In	Trays	BHDPO8835E	4282	0	0
lot199	Bake_In	Trays	BHDPO8835E	5960	1	5960
lot201	Bake_In	Trays	BHDPO8835E	7385	0	0
lot219	Bake_In	Trays	BHLLZ7485H	6535	1	6535
lot235	Bake_In	Trays	CREGF0290P	5043	0	0
lot234	Bake_In	Trays	CREGF0290P	5642	0	0
lot236	Bake_In	Trays	CREGF0290P	5816	1	5816
lot232	Bake_In	Trays	CREGF0290P	5973	1	5973
lot293	Bake_In	Trays	DOPHU4614Q	7195	0	0
lot291	Bake_In	Trays	DOPHU4614Q	10707	1	10707
lot328	Bake_In	Tubes	DSXXJ1471E	6381	1	6381
lot329	Bake_In	Tubes	DSXXJ1471E	8758	1	8758

Delivery Plan					
Provide Lot Count(ci)	Product Type	Requirement Chip Quantity	Provide Chip Quantity(xi)	Dif	xi-ci
1	ACNYW1625J	48714	4704	44010	4703
1	ALSRP5590C	1071	13227	-12156	13226
2	ARCYI6496Z	19936	17509	2427	17507
1	ARXZK2314V	9071	7699	1372	7698
0	AYEHS2904M	3929	0	3929	0
3	BBKRW9735V	63214	33690	29524	33687
0	BDUFC2928C	24143	0	24143	0
0	BGGII2097V	17857	0	17857	0
6	BHDPO8835E	11129	10227	902	10221
1	BHLLZ7485H	33929	6535	27394	6534
4	CREGF0290P	13929	11789	2140	11785
0	CRTCK3425B	12000	0	12000	0
0	DFTHC1126O	257	0	257	0
0	DGEPX1320V	5357	0	5357	0
2	DOPHU4614Q	17500	10707	6793	10705
3	DSXXJ1471E	2857	24058	-21201	24055
0	DVYYW0521A	6857	0	6857	0
0	EFLFF0922K	7500	0	7500	0
35	ELXUA2297T	190429	176875	13554	176840
0	EPEOQ3051X	500	0	500	0
0	FHQHB3513Q	6672	0	6672	0
9	FUJLR9745W	15714	11836	3878	11827
3	GUJLK0157P	86071	62448	23623	62445

Table 4 : Excel GA Solver

<u>User task: Input Lot (DB/CSV File)</u>

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 5 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.

LOT	STEPNAME	carrier Type	Product Type	QUANTITY
lot1	Bake_In	Trays	LEQUD1106E	3674
lot2	Bake_In	Tubes	UAQAL5502F	27944
lot3	Bake_In	Trays	KILKJ2749Y	8195

Table 5 : 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.

productType	volume	temperature	bakeTime
ACHXA3595G	0.0000416667	125	12
ACNYW1625J	0.0000277778	125	24
ALSRP5590C	0.0000016026	125	24
APYLB9291C	0.0000104167	125	24
ARCYI6496Z	0.0000046296	125	24
ARXZK2314V	0.0000046296	125	24
AYEHS2904M	0.0000055556	150	24
AZFGJ6792V	0.0000277778	125	24
BBKRW9735V	0.0000083333	150	24
BDUFC2928C	0.0000277778	125	24
BGGII2097V	0.0000055556	150	24
BGLWT0100N	0.0000198413	125	24
BHDPO8835E	0.0000104167	125	24
BHKIO5418Q	0.0000055556	150	24
BHLLZ7485H	0.0000416667	125	12
BLCND8992P	0.0000277778	125	24

Figure 3: Product Information DB Table

ovenId	temperature	processType
BO1	125	BakeIn
BO10	125	BakeIn
BO11	125	BakeIn
BO12	125	BakeIn
BO13	150	BakeIn
BO14	150	BakeIn
BO15	125	BakeIn
BO16	125	BakeIn
BO17	125	BakeIn
BO18	150	BakeIn
BO19	125	BakeIn
BO2	125	BakeIn
BO20	125	BakeIn
BO21	125	BakeIn
BO22	125	BakeIn
BO3	150	BakeIn

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 ovens have the same capacity. The rule engine will take information provided in the CSV file (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 the Optaplanner solver to perform a local search for the best solution.

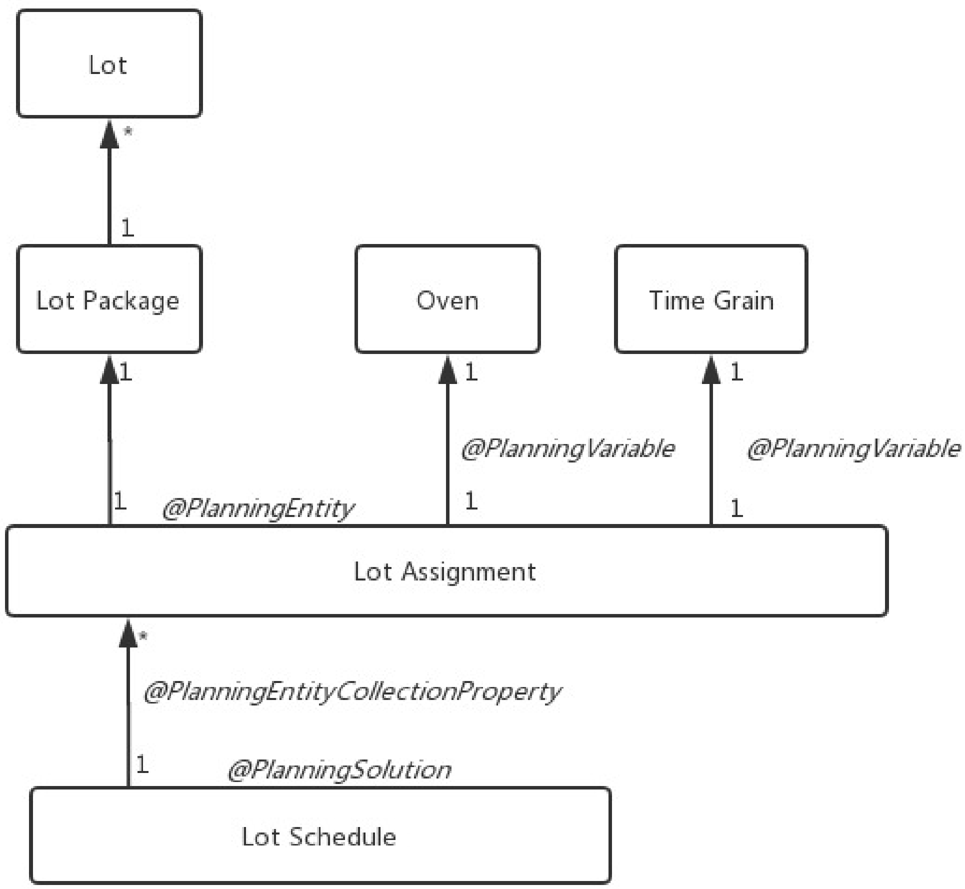

Figure 5: Class Diagram

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

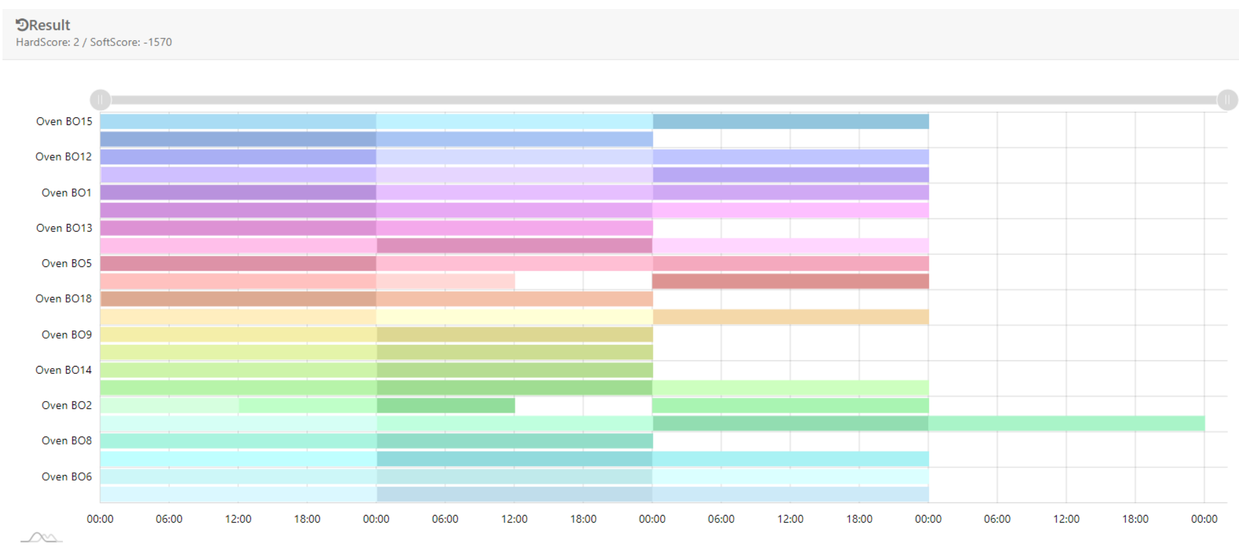

Figure 6: Screenshot of Planning Solution Representation

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.

3.0) CONCLUSION

In this project we propose and implement an effective and predictive system, SDOT (Scheduling and Despatch Optimisation Tool), as a solution to address the complexity of problems in predictive scheduling based on software platforms of Excel Solver and Optaplanner. The integrated solution through these two platforms help the test manufacturing firm to meet its delivery demand in due time. Conventional planning based on static calculation poses a challenging task for the production planner due to the large size and makes lot scheduling prone to human errors. With our scheduling system, SDOT, the complex inter-relations between the product requirement, machine parameters, business requirement such as delivery demand fulfilment can be accounted to provide a schedule plan to meet factory performance measure. The solution approach is two fold, where firstly the lot selection is performed using Excel Solver. This relies on application of an Evolutionary computation approach using Genetic Algorithms (GA). Following that, the optimal selected based on the GA is passed to the scheduling system developed using Optaplanner which takes into account the constraints of the system as codified in the form of business rules.

3.1) IMPROVEMENTS FOR THE SDOT SYSTEM

Job shop scheduling problem is a well-researched topic. There are many comprehensive and robust solutions and approaches that have helped to address the manufacturing challenges. The SDOT system we have proposed and implemented is a predictive scheduling system. In reality, scheduling is a more complex problem with multiple states and a myriad of dynamic parameters that needs consideration to achieve an optimal solution. Below, we explain some of the limitations present and potential enhancements that could be integrated into SDOT to improve its functionality.

Predictive systems vs Reactive system

SDOT is currently designed as a predictive scheduling system based on certain premises and conditions. This could be enhanced by a reactive system whereby the tasks and their constraints may become known only incrementally or while the system is already running based on the current knowledge domain. In reality, there arises dynamic instances such as urgent orders, unpredicted machine breakdown and WIP shortages which may require a better solution. Based on knowledge of the new constraints and limitations, the scheduler can react by recalculating the optimal schedule.

Local optimal vs global optimal

Currently SDOT uses GA and search techniques that may result in solutions that are considered feasible for a current frame in time based on current knowledge. This may possibly yield solutions as a local optimum within its search parameters. For a more comprehensive system, that gives consideration to downstream processes, the availability and the states of the machine, a global optimum can be aimed for, by employing ‘forward-looking’ strategies. However due to the complexity of such techniques , this is not considered for this project.

Evaluation of other techniques

For this project, only one scheduling strategy/optimization/search technique is used. To test and evaluate how different optimization techniques perform and fare against one another based on varying system conditions, they can be run via a simulation model. For example, the schedule results output from the different search/optimization techniques can be then used as an input to a virtual factory simulation model to evaluate other factory performance measures such as operating equipment efficiency (OEE).

Appendix

APPENDIX A: DELIVERY PLAN

Screen shot of Delivery Plan and WIP Lot list
Document uploaded as "GALotSelection.xlsx"

Delivery Plan

Provide Lot Count(ci)	Product Type	Requirement Chip Quantity	Provide Chip Quantity(xi)	Dif	xi-ci
1	ACNYW1625J	48714	4704	44010	4703
1	ALSRP5590C	1071	13227	-12156	13226
2	ARCYI6496Z	19936	17509	2427	17507
1	ARXZK2314V	9071	7699	1372	7698
0	AYEHS2904M	3929	0	3929	0
3	BBKRW9735V	63214	33690	29524	33687

WIP Lot List

LOT	STEPNAME	carrier Type	Product Type	QUANTITY	A	q'
lot2	Bake_In	Trays	ACNYW1625J	4704	1	4704
lot87	Bake_In	Trays	ALSRP5590C	13227	1	13227
lot89	Bake_In	Tubes	ARCYI6496Z	8617	1	8617
lot88	Bake_In	Tubes	ARCYI6496Z	8892	1	8892
lot102	Bake_In	Tubes	ARXZK2314V	7699	1	7699
lot136	Bake_In	Tubes	BBKRW9735V	4706	1	4706
lot134	Bake_In	Tubes	BBKRW9735V	11455	1	11455

APPENDIX B: Product Processing Requirements