Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

APPLICATION OF SIMULATION BASED REINFORCEMENT LEARNING FOR OPTIMIZING LOT DISPATCHING RULES OF SEMICONDUCTOR FAB

Chihyun Jung
Younghwan Kim
Hyeyun Kang
You-In Choung

SK Hynix
2091 Gyeongchung-daero Bubal-eup
Icheon-si, Gyeonggi-do 17336, SOUTH KOREA

ABSTRACT

Automatic lot dispatching system performs critical role in semiconductor FAB. It determines the order of jobs to process for individual equipment based on predefined rules. The dispatching rule should be aligned with the global objectives such as WIP move target, minimizing cycle time, maximizing total throughput, and etc. In this study, we examine a reinforcement learning (RL) algorithm to optimize dispatching rules, which should be aligned with the global objectives of FAB. A discrete event simulation model is developed as a learning environment for RL. The proposed methodology has been examined in real FAB. Promising results are presented and the limitation of the approach is discussed.

1 INTRODUCTION

1.1 Weighted Sum Priority Dispatching Rules

A semiconductor FAB can be considered as a big job shop production system. Job dispatching is critical task in such an environment. It decides the order of jobs to process in individual tools. FAB operation KPIs (key performance measures), such as total cycle time, tool utilization, and work-in-process (WIP) balance are affected by dispatching sequence. Modern semiconductor FABs have real time dispatching (RTD) system which automatically decides dispatching sequence. Most of RTD systems are rule based systems. Each tool group is associated with different dispatching rules. Basic concept of job sequencing is to evaluate job priorities which are determined by multiple priority factors and their importance. One popular dispatching method is to sequence jobs by the weighted sum of multiple priority factors. We call it as 'Weighted Sum Priority Dispatching (WSPD)'. Priority factors are properties which are associated with a job or an equipment, such as waiting time, due date, remaining daily throughput target, remaining Q-time, setup time, and etc. Individual priority factors are weighted depending on their importance which is different by FAB needs. For example, First-In First-Out dispatching rule can be implemented by assigning weight '100' to the priority factor 'waiting time', and assign zero weight to the other priority factors. WSPD is popular because it can flexibly cope with dynamic change of FAB environment.

1.2 Problem Description

Dispatching rules are usually maintained by manufacturing operation engineers. One of the major effort in maintaining WSPD rules is tuning the weights of priority factors so that the job priorities conform with the global objective of the FAB. Adjusting the weights is not trivial due to several reasons:

- Evaluating the long term consequences of changing the dispatching rule is very difficult.
- The dispatching priority factors are not directly related with FAB KPIs. For instance, it is hard to assess the impact of increasing the weight for 'minimum setup time' priority to the total lot cycle time of FAB KPIs.
- Units of the priority factors are different, although they are numerically measurable. Therefore, priority weights should scale the priority factors to normalize their impacts.

This study aims to automate the weight adjustment of dispatching rules.

2 SOLUTION APPROACH

In order to solve the problem, we examine reinforcement learning (RL) algorithm. RL algorithm consists of four elements including agent, state, action, and reward. The agent learns which action gives the most reward in each state. The reward is usually delayed like a chess game. Delayed reward is a major characteristic of dispatching problems as well. Present dispatching decision might not give immediate reward but contributes to future FAB KPIs. There are several studies which attempt to use RL for scheduling problems.

For our problem, the action of RL is defined as selecting weights for WSPD rule, the reward is FAB KPIs, and the state is FAB environment at a certain time. However, the real FAB environment cannot be used as the learning environment for RL. Because, engaging an immature algorithm with real FAB can jeopardize the production. And because, it doesn't give enough chance to learn abnormal incidents, which rarely happen in real FAB. Therefore, many studies on RL use simulation model as a learning environment.

We developed a discrete event simulation model which is a widely accepted modeling method for factory operation. Data regarding product, route, operation, process time, recipe, WIP, and operational constraints are prepared as input data for simulation. Individual lots are modeled as simulation entities, and process equipment is modeled as resource. The setup times and process times are modeled in detail, which are different by tool types, recipes, and lot sizes. The material handling system is abstracted, so that the simulation incurs a fixed transfer time for lot transfers.

We use TensorFlow library to implement RL algorithm. Whenever the lots are dispatched in the simulation, the reward is returned based on the simulation states at the moment. After that, the priority weights of dispatching rules are updated by the algorithm.

3 RESULT AND DISCUSSION

We applied the approach to a FAB of SK Hynix which is one of the leading memory chip manufacturer. The algorithm takes around 20 days of simulation clock time to converge the reward value. Both strength and weakness of our approach are observed from the result.

The strength of the approach is that the algorithm adjusted the weight of WSPD rules as a human engineer does in most cases. It selects the weights appropriately to scale up or down individual factors to normalize the unit difference between priority factors. It also effectively finds the right dispatching priority factors to weigh, according to the emphasis of given FAB KPIs. For example, if the given FAB KPIs emphasizes maximizing total wafer moves, the RL algorithm heavily weighs 'minimum setup time' priority factor.

The weakness of the approach are mainly two-folds. The first is unexplainable odd results. The RL occasionally gives heavy weights on a certain priority factors which seems unrelated with the given FAB KPIs in views of human engineers, of course. However, it is very difficult to analyze the reason due to the nature of RL algorithms. The second is the lag of reusability. When the simulation data is updated, for example, a new equipment introduced, the RL algorithm has to be re-trained.

As further research, we are planning to develop a simulation based AI scheduler which directly use deep learning technique to dispatch lots, instead of optimizing dispatching weight factors.