

Automation

AMHS in the Reinforced Dispatch Learning Environment of IC Fabs

GEORGE HORN, Middlesex industries Inc.

Reinforcement Dispatch Learning (RL) and the wafer transports

IT IS SAID THAT ACHIEVING UNIVERSAL dispatch solutions to front end IC manufacturing is NP hard*. The ultimate objective is optimum performance while balancing cycle time and utilization.

Machine Learning algorithms are being increasingly applied in the industry, and great hopes are placed on AI agents for dispatch solutions. The assumption is that if the current state of the system is known, however complex it may be, it can be improved via dispatch learning agents. Having past accumulated knowledge, outcomes may be automatically influenced in favor of presumed classes of benefit, via learning agents with defined Policy. Indications are that the application of such learning agents will outperform today's heuristic dispatch algorithms. The question remains how current AMHS designs, as executors of dispatch, will contribute to or detract from the above.

Dispatch system

An idealized workflow system is put forth below, to serve in understanding dispatch principles subject to reinforcement learning. A workflow model of the fab is constructed, where optimum utilization at the shortest possible cycle time is sought via the dispatch control of WIP content at each process step.

The model defines the smallest element in a fab as a process tool with an input buffer for WIP (the input buffer being an upstream segment of the AMHS transport). The process consists of work flowing through these elements sequentially and repeating through recursion. The computational task is to achieve max utilization at a minimum wafer lot content (cycle time).

In practical terms, we need to have at least one wafer lot waiting at the input of each such unit to assure utilization, while at the same time immediate dispatch of a processed wafer lot must be assured (no waiting storage of substrates at process output). In this picture, an element consists of an AMHS segment bringing and storing wafer lots before the process tool, and the process tool itself. The graphic model of the fab could then be likened to a Petri Net, where Transitions are controlled to limit minimum buffer requirements before the process tools. Transitions or gates control the WIP flow into a tool, and gates control the branching of incoming workflow to destination tools. Ideal buffer contents associated with single lot tools would need a buffered capacity of 2 lots, while batch tools would have double the batch size capacity to

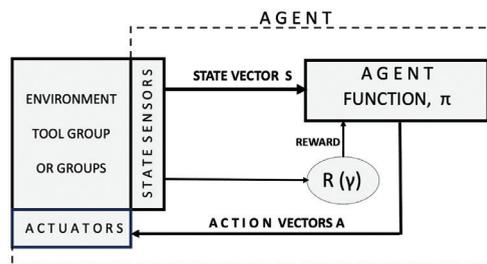


Figure 1. The state of a system and the action of an agent to improve it. The agent's action is based on the current state of the environment, on its past experience with it, and on its policy. Its learning is continually reinforced via a reward assessment.

buffer. Overall, substrate content of the fab would be that of the true process capacity plus double the true process capacity in wait. Such a workflow system can closely be approximated with digitized conveyor transports [3].

System state and actions

Reinforcement learning is an autonomous process rewarding the learning Agent, depending on its Action, in moving the system State towards achieving a Policy. At each time step, Δt , the system state, s , is observed and the Agent may choose any Action available, a , moving the system into a new State, s' , and rewarding the Agent with R . Such is a Markov process, since each step is independently decided of a previous one, Ref [2] and FIGURE 1.

In our model, the overall State of the environment is represented by $S = \{R_a(s, s') \text{ At Each } S_1, S_2, \dots, S_i\}$, where S_i characterizes the WIP distribution of a product type, $S_i = \{s_1, s_2, \dots, s_i\}$ on incoming AMHS segments of s_i .

*In computational complexity theory, NP-hardness (non-deterministic polynomial-time hardness) is the defining property of a class of problems that are informally "at least as hard as the hardest problems in NP".

And a concurrent action of the Agent would be to regulate WIP flow into a tool or at WIP flow branching. $A = \{A_1, A_2, \dots, A_i\}$, where $A_i = \{a_1, a_2, \dots, a_i\}$. The Agent function would strive to achieve WIP in wait distributions at tools that do not exceed WIP content double the size of ideal WIP [3]. $\pi_i = P(A_i = a_i \mid S_i = s)$. In other words, the Agent policy will tend to converge towards an even, minimum WIP content in the tool buffers. Even as it starts out from an unbalanced system state.

Timing

Encoding neural networks with the actual nodes of a semiconductor dispatch system, and its representation of system states, has demonstrated successful Agent Learning and Agent action for the Semiconductor environment. These demonstrations generally assume a practically instantaneous delivery of Actions. However, to implement those actions in the real fab's physical environment involves timing for the Actions. As a consequence, considering today's well accepted Discrete vehicle AMHS, the above policy statement cannot be instantaneously executed. In other words, $A_i = a_i \mid S_i = s$ cannot happen.

For the assessment of system states, followed by an intelligent agent's calculations, and release of its control actions, a time period is required. Ideally, for the control action to be effective, its release should be made before the state of the system has changed on its own. A computationally intensive process (dependent on the complexity of the controlled state)

Yet, it is likely that the states of the system in typical IC manufacturing will rapidly and unpredictably change, and in a stochastic fashion, considering a functional set of processes, or the whole. 50% of the changes occur in less than 5 minutes, while 80% of the changes occur in less than 15 minutes [1]. This considered, an iteration of the

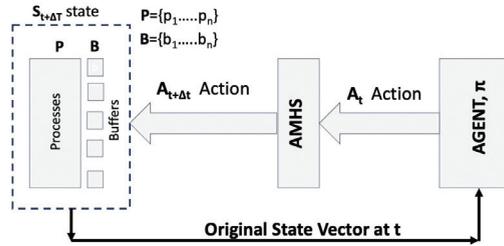


Figure 2. The action of an Agent is based on the original state of the environment while the arrival of that action is delayed randomizing its benefit due to the altered state of the environment at the time of the action's arrival.

Reinforced learning cycle should be as short as possible. Today's computer systems, collecting state matrix vectors from the system and solving algorithms of reinforced learning (ex. neural networks), followed by a matrix of actions, may take tens of seconds to a minute. In general, appraising the parameters of semiconductor manufacturing process states and the issued actions to modify them may be computer fast. If, however, such machine learning is applied to wafer lot dispatching in semiconductor fabrication, then consideration must be made to the so-called wafer lot "moving agents", i.e. the AMHS, in segments or as a whole, having compatible execution capabilities. This, however, is not the case. Average delivery times of current AMHS designs (discrete vehicle transport) are in the several minutes, up to 15 minutes range, and are themselves stochastically distributed. This, in general, frustrates the actions issued by the agent to various degrees, and is likely to result in the reduction or elimination of rewards (FIGURE 2). Overall, the scatter of system states approaching the desired goal is adversely affected.

An example instance of such a counterproductive Action maybe where the system state routinely changes in less than five minutes, while the Action corresponding to the original state of the system arrives with a delay of 10 minutes.

The AMHS is the essential tool in

delivering an Action from the Agent. Current use AMHS technology (discrete vehicle transport) depends on excess WIP accumulation in order for it to assure WIP availability at tool inputs. But, to improve the chances of success for Reinforcement Learning Dispatch, the AMHS should be able to deliver WIP without delay. Thus, conveyor-based AMHS should be considered. Because of the all-time availability of conveyor

transport at the output of a process, the WIP can directly move, without waiting times, to the next process, and thus become the buffer for it. 

About the author

Mr. Horn has worked for many years towards understanding the roles AMHS can play in the manufacturing process (Cf. publications IEEE Transactions) and has pointed out ways to improve AMHS roles. Currently he is the president of Middlesex General Industries, Inc. A manufacturer of conveyor AMHS solutions. He can be reached at gwhorn@midsx.com

REFERENCES

1. Operations Management in Automated Semiconductor Manufacturing with Integrated Targeting, Near Real Time Scheduling and Dispatching. Nirmal Govind et All, IEEE Transactions on Semiconductor Manufacturing, Vol. 21, No 3.
2. Autonomous Order Dispatching in the Semiconductor Industry Using Reinforcement Learning. Andreas Kuhnle et All, Elsevier B.V. 2019.
3. Towards Lean Front End IC Manufacturing (with AMHS Implants), George W Horn, IEEE Transactions on Semiconductor Manufacturing, 2022.
4. Deep Reinforcement Learning for Semiconductor Production Scheduling, Bernt Washneck, et All, GSaME, Universität Stuttgart, (Grant SemI40) and support by Infineon Technologies.
5. Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning, Cong Zhang et All, Singapore Institute of Manufacturing Technology, A*STAR. 34th Conference on Neural Information Processing (2020).