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.

MACHINE LEARNING POWERED CAPACITY PLANNING FOR SEMICONDUCTOR FAB

Keyhoon Ko
Seokcheol Chang

VMS Global, Inc.
1952 Gallows Road
Suite 110
Vienna, VA 22182, USA

Won-Jun Lee
Byung-Hee Kim

VMS Solutions Co. Ltd.
U-Tower Building A #2001
Sinsu Street, 767
Yongin, 16827, REPUBLIC OF KOREA

ABSTRACT

Semiconductor wafers are manufactured by stacking hundreds of layers engraved with circuit patterns. Wafer fabrication process with the characteristic of re-entrant flow is a complex job-shop that consists of several work areas such as lithography, etch, and diffusion. Each work area has several workstations with one or more machines that execute the same operation. Capacity planning for a wafer fab is difficult; one must determine the required machine count to meet demands on time. This study proposes a methodology to find the optimal machine count for each workstation using an approach that combines optimization, simulation, and machine learning techniques. The experimental example demonstrates that this approach can systematically provide a good and practical solution.

1 INTRODUCTION

As the global semiconductor shortage continues, major semiconductor companies are rushing to build new wafer fabrication facilities (fab). Especially in this capital-intensive industry, it is very important to verify that target production can be achieved with planned resource investment. As a result, there is an increasing need for a next-generation capacity planning system that provides an optimal solution while enabling more accurate and faster calculations.

Capacity planning optimizes resource allocation to fulfill demand. Semiconductor manufacturing has some special features that complicate the analysis of such systems: (1) a large number of processing steps, (2) re-entrant flows, (3) batch machines, and (4) random machine failures. The main consequence of the re-entrant flow nature is that wafers at different layers in their manufacturing step must compete for the same machines.

Due to the complexity and importance of the problem, much research has been conducted on capacity planning for semiconductor fabs. Survey papers have categorized the problem according to decision level, scope, objectives, and approaches and have provided an in-depth review of the research (Wu et al. 2005; Geng and Jiang 2009; Uzsoy et al. 2017).

Traditionally, researchers have applied mathematical programming methods to optimize semiconductor fabs. Klemmt et al. (2012) applied a linear programming model; Barahona et al. (2005) tried a mixed-integer stochastic model and linear programming relaxation to minimize the unsatisfied demand considering budget constraints; Karabuk and Wu (2003) formulated a stochastic model for uncertain demand. However, the results from these approaches are not particularly executable since they do not consider production constraints such as sequence-dependent setups and queue-time constraints.

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

Simulation-based approaches that consider operation rules and constraints have also been conducted (Wang et al. 2018; Chen and Chen 2010). These methods may provide a practical solution, but it is not guaranteed to be optimal.

The present study proposes a methodology to find the practically optimal resource capacity using a combination of optimization, simulation, and machine learning techniques. The capacity planning problem is defined with three sub-problems, and the approach is described in Section 2. The proposed approach is illustrated and verified with an experimental example in Section 3. The conclusion and discussion follow in Section 4.

2 PROBLEM AND APPROACH

Long-term capacity planning requires one to assess whether resource capacity is sufficient for the demand projection of upcoming years (with a planning horizon of greater than 12 months). It also can be used for new fab construction planning, which focuses on validating the product mix and tool investment. Basically, it necessitates determining the optimal machine count for each workstation to deliver customer orders on time. However, capacity planning poses several challenges:

- *Inaccurate and incomplete master data:* Capacity planning software receives data from different sources such as sales, forecasts, enterprise resource planning (ERP), and manufacturing execution system (MES). If this information is not fully connected, it becomes a major challenge in the planning process, and human validation of all data is required.
- *Difficult to create new input data:* While making long-term plans, capacity planning should consider new products and/or new machines to be introduced in the near future. To ensure a fully integrated data set, new input data include not only new products and machines but also their associated data. Eqp-Arrange which defines the loadable relationship between job and equipment is an example, and its processing time is another.
- *Unbalanced Eqp-Arrange among machines:* Some machines can process more products, while others can do very few. Usually, general-purpose machines that can process a greater variety of jobs have a greater workload than dedicated ones. It is important to keep the balance between machines to maximize throughput. For example, assigning limited jobs to the dedicated machine first and allocating others to general ones may produce better results.
- *Inaccurate decision based on static work in progress (WIP) inflow:* An optimization approach usually assumes that the lot arrives at a predetermined time at the work area. This assumption simplifies the problem but is not always true and thus difficult to execute. In addition, integration of area-level decisions does not guarantee fab-level optimization.
- *Difficult to compare alternatives:* It is also important to have an intuitive way to compare results. Tentative key performance indicators are demand fulfillment rate, cycle time for each product, and resource utilization.

Capacity planning in this study is divided into the following three sub-problems:

1. *Optimize Eqp-Arrange:* To improve the workload balance between machines, this optimization process determines how to best allocate jobs and machines. The optimization results are provided in the form of the Eqp-Arrange list to be inhibited. Simulations with original data and reduced data are conducted and compared to verify the impact of the Eqp-Arrange optimization.
2. *Calculate the initial required machine count:* Another optimization model is formulated to calculate the optimal machine count to minimize the total tardiness of the demand. Since this optimization model does not consider operational rules and constraints, the result is used as an initial solution for the following machine-learning simulation.
3. *Optimize required machine count:* A machine learning technique, policy gradient with parameter-based exploration (PGPE), is applied to find the optimal solution with the initial solution obtained above.

More details are provided in the following sections.

2.1 Optimization 1: Optimizing Eqp-Arrange

When a machine processes 25 wafers with 10 seconds of tact time, its *workload* is 250 seconds. If a product has a demand of 1,000 wafers, each step of the product should process 1,000 wafers assuming step yield is 100%. For a specific step, if only one machine is loadable and its tact time is 36 seconds, then 36,000 seconds (i.e., 10 hours) of workload is assigned to the machine. If two machines are loadable for the same operation with the same tact, then each machine has a workload of 5 hours. In this manner, we can calculate static workload for each machine using demand, Eqp-Arrange, and tact time data.

Figure 1 presents static workload, which is calculated with 6 months (182 days) of demand for each workstation. To meet demand in 182 days, the workload must be less than or equal to 182 days. However, the workload of a machine may be much higher or much lower than the reference value, as shown in Figure 1. This variability is very typical of existing wafer fabs.

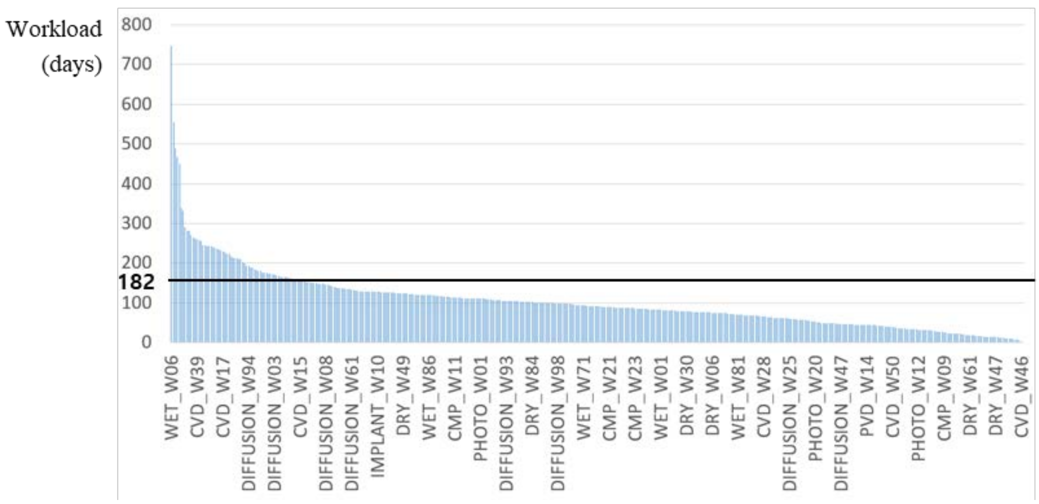


Figure 1: Static workload is unbalanced across workstations. Note that this static workload can be different from the optimization result or simulation result.

To maximize the step moves which is the total wafer counts processed at steps, it might be helpful that bottleneck machines process jobs that only they can do and leave the rest to other equipment. An optimization model, Optimization 1, can be formulated to improve the balance of the workload among workstations. Table 1 defines the parameters and variables for both optimization models (i.e., Optimization 1 and 2).

The objective of the model is to maximize the total on-time delivery for all demands:

$$Max \ \ \sum_{m \in P} \sum_{t \in T} u_{mt} \ .$$

The first constraint is the *WIP balance* constraint. For a product *m*, step *s* at time bucket *t*, the end on hand (EOH) WIP amount equals to beginning on hand (BOH) WIP plus incoming WIP minus outgoing WIP. For all steps except the last, $\forall m \in P, \forall s \in R_m, s \neq s_{ml}, \forall t \in T$

$$w_{mst} + \sum_{e \in E_{ms-1}} \sum_{t' \in T, t' \leq t} r_{ms-1t't} x_{ms-1et'} - \sum_{e \in E_{ms}} x_{mset} = w_{mst+1}.$$

Table 1: Set, parameters, and variables are defined for the optimization modeling.

Notation		**Description**
Set	P	Set of products
	R_m	Route, which is a series of steps for a given product *m*
	E	Set of workstations
	L_e	Eqp-Arrange, which is a set of loadable products/steps at a given workstation *e*
	E_{ms}	Eqp-Arrange, which is a set of loadable workstations for a given product *m,* step *s*
	T	Set of time buckets in the planning horizon
Parameter	d_{mt}	Demand quantity for product *m* at time *t*
	τ_{mse}	Processing time of product *m*, step *s* at workstation *e*
	η_e	Number of machines at workstation *e*
	c_{et}	Capacity of a single machine at workstation *e* at time *t*
	s_{mi}	i^{th} step of product m, where s_{ml} is the last step of the product
	$r_{mst't}$	The arrival ratio to the next step at time *t* when product *m* finishes step *s* at time t'
Variable	x_{mset}	Assigned number of product *m*, step *s* at workstation *e* at time *t*
	w_{mst}	Number of beginning on hand WIP of product *m*, step *s*, at time (bucket) *t*
	u_{mt}	Number of products *m* with on-time delivery at time *t*
	b_{mt}	Number of products *m* with delayed delivery at time *t*
	α_e	Additional machine count at workstation *e*

For the last step, $\forall m \in P, s = s_{ml}, \forall t \in T$

$$w_{mst} + \sum_{e \in E_{ms-1}} \sum_{t' \in T, t' \le t} r_{ms-1t't} x_{ms-1et'} - u_{mt} = w_{mst+1}.$$

The next constraint is the *delivery to demand* constraint: the current day's demand plus the previous day's backlog must equal the total of the current day's delivery and backlog. The demand for product *m* at time *t* should equal the on-time delivered amount plus the current delayed amount minus the previous delayed amount:

$$u_{mt} + b_{mt} - b_{mt-1} = d_{mt} \quad \forall m \in P, \forall t \in T.$$

The final constraint is the *capacity* constraint. The total processing time (i.e., *workload*), cannot exceed the capacity for each time and workstation. Every machine in a workstation is assumed to have the same processing time:

$$\sum_{(m,s) \in L_e} \tau_{mse}\, x_{mset} \le \eta_e c_{et} \quad \forall e \in E, \forall t \in T.$$

2.2 Optimization 2: Calculating the Initial Required Machine Count using Optimization

We construct another optimization model to determine the minimum required machine count to fulfill the demand. Note that Eqp-Arrange is updated with the result of the previous optimization model.

The objective function is to minimize the additional capacity to satisfy all demands:

$$Min \;\; \sum_{e \in \mathrm{E}} \alpha_e.$$

WIP balance constraints are the same as the previous model, but the *delivery to demand* constraint is different as this model does not allow late delivery for any demand:

$$u_{mt} = d_{mt} \quad \forall m \in P, \forall t \in T.$$

The *capacity* constraint is modified to consider additional machine capacity:

$$\sum_{(m,s)\in L_e} \tau_{mse}\, x_{mset} \leq \eta_e c_{et} + \alpha_e c_{et} \quad \forall e \in E, \forall t \in T.$$

2.3 Optimizing Required Machine Count using Machine Learning

This optimization task is essentially a search problem to find the optimal required machine count for each workstation. Reducing the solution space to bottlenecks would be effective and efficient. Simulation using the initial value (α_e in Table 1) as calculated by Optimization 2 is performed, and the daily utilization for each workstation is captured. If a workstation has more than one day when its daily utilization equals 100% during the plan horizon, it is identified as a bottleneck workstation.

The policy gradient with parameter-based exploration (PGPE) was proposed to address the issue that symmetric sampling in parameter space leads to lower variance gradient estimates (Sehnke et al. 2010). As a model-free reinforcement learning method, the policy is defined by a distribution over the parameters of a controller. The parameters are sampled from this distribution at the start of each iteration, and thereafter the controller is deterministic. Since the reward for each iteration depends on only a single sample, the gradient estimates are significantly less noisy, even in stochastic environments.

Figure 2 depicts the procedure of applying the PGPE algorithm, where its reward considers the total tardiness and maximum daily utilization to avoid excessive investment; parameters are mean μ and standard deviation σ assuming the parameter distribution is normal. The samples are symmetrically drawn from the predefined parameter distribution with each iteration. Each scenario is simulated in parallel, and the results are used to update the parameters. Details on symmetric sampling and gradient formulation are found in Sehnke et al. (2010).

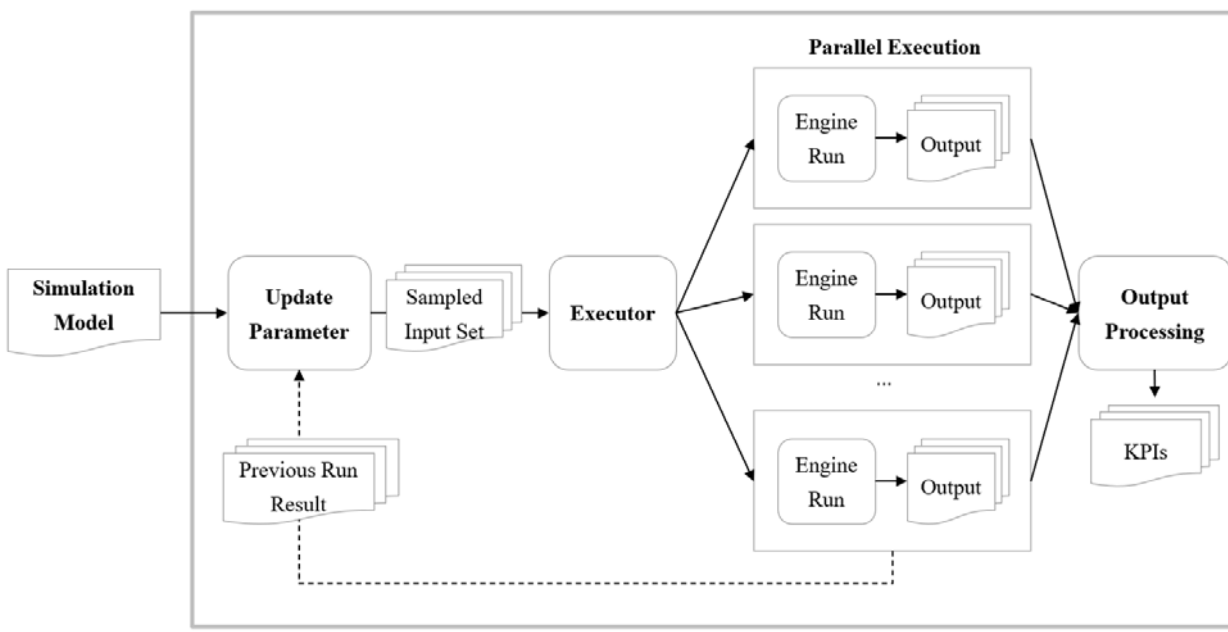


Figure 2: Machine learning environment using the PGPE algorithm.

3 EXPERIMENT AND RESULTS

As a planning and scheduling solution provider, VMS has some test data sets that have been improved and now reflect behavior quite similar to that of the real fabs. The test model has 14 products, which have 1,400–2,200 steps, including inspection and measuring steps. Eight processing work areas have 461 workstations in total.

For the sake of run-time, the planning horizon is 6 months instead of more than a year where the monthly demand is around 150–200K. The simulation uses total tardiness as the key performance indicator. Let L_j denote the lateness of job j, which is equal to the difference between the job's completion time C_j and its due date d_j. The tardiness of job j, T_j, is equal to max(0, L_j). Total tardiness is calculated as $\sum_j T_j$.

3.1 Eqp-Arrange Optimization

A linear programming model is formulated with 32,944 variables and 16,776 constraints, as described in section 2.1. It completes within 1.22 s and reduces the row count of Eqp-Arrange data from 13,252 to 8,056, as indicated in Table 2. Simulations prove that the reduced Eqp-Arrange set outperforms the original data set in terms of step moves and tardiness, as indicated in Tables 2 and 3.

Table 2: About 40% of Eqp-Arrange can be reduced, but total step moves are slightly increased.

AREA	EQP-ARRANGE COUNT			STEP MOVES		
	ORIGINAL	REDUCED	RATIO	ORIGINAL	REDUCED	DELTA
CMP	198	173	87%	6,130,246	6,115,685	-14,561
CVD	981	897	91%	31,384,663	31,369,849	-14,814
DIFFUSION	976	810	83%	28,605,689	28,847,976	242,287
DRY	875	785	90%	26,874,488	26,923,893	49,405
IMPLANT	561	551	98%	21,854,244	22,162,009	307,765
PHOTO	1,393	1,008	72%	36,333,136	36,650,670	317,534
PVD	132	126	95%	4,444,014	4,415,092	-28,922
WET	8,136	3,706	46%	133,165,816	134,047,570	881,754
Total	**13,252**	**8,056**	**61%**	**288,792,296**	**290,532,744**	**1,740,448**

Table 3: With the reduced Eqp-Arrange, total tardiness is slightly decreased.

PRODUCT	ORIGINAL	REDUCED
PROD01	780,500	788,484
PROD02	234,842	217,480
PROD03	303,075	303,262
PROD04	565,477	548,698
PROD05	1,032	1,116
PROD06	21,521	21,082
PROD07	123,590	110,163
PROD08	8,277,001	7,885,896
PROD09	557,740	528,790
PROD10	930,770	947,231
PROD11	125,919	127,217
PROD12	2,937,200	3,046,688
PROD13	734,656	722,864
PROD14	41,122	42,222
TOTAL	**15,634,445**	**15,291,193**

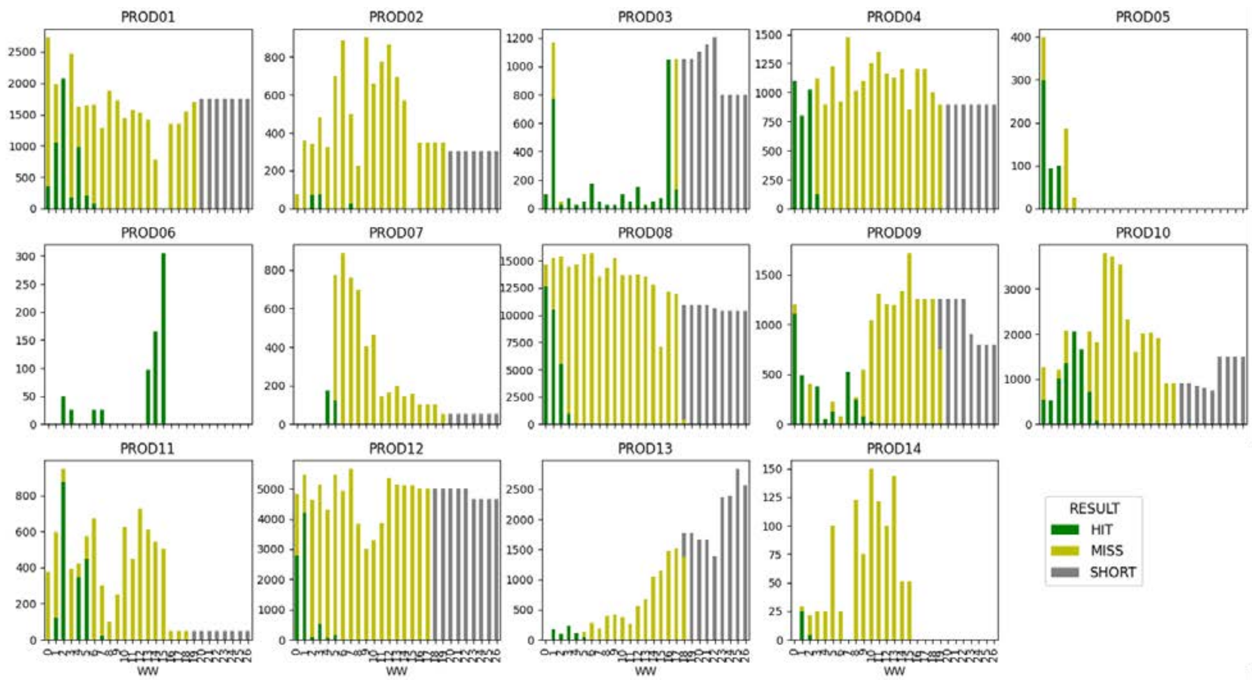
3.2 Calculation of Initial Required Machine Count

Optimization 2 modeled in section 2.2 provides the required machine count, as described in Table 4. Fifteen workstations out of 461 require more capacity. For example, although workstation CVD_W18 has four machines, it needs 10 (=4 * 2.5002) machines to complete demands on time. Instead of adding six additional machines, the simulation reduces the tact time for convenience. If the tact time of CVD_W18 is 10 seconds for a step, its new tact time is 4 (=10/2.5002) seconds. Note that optimization does not consider all constraints and rules; more machines may be required in reality.

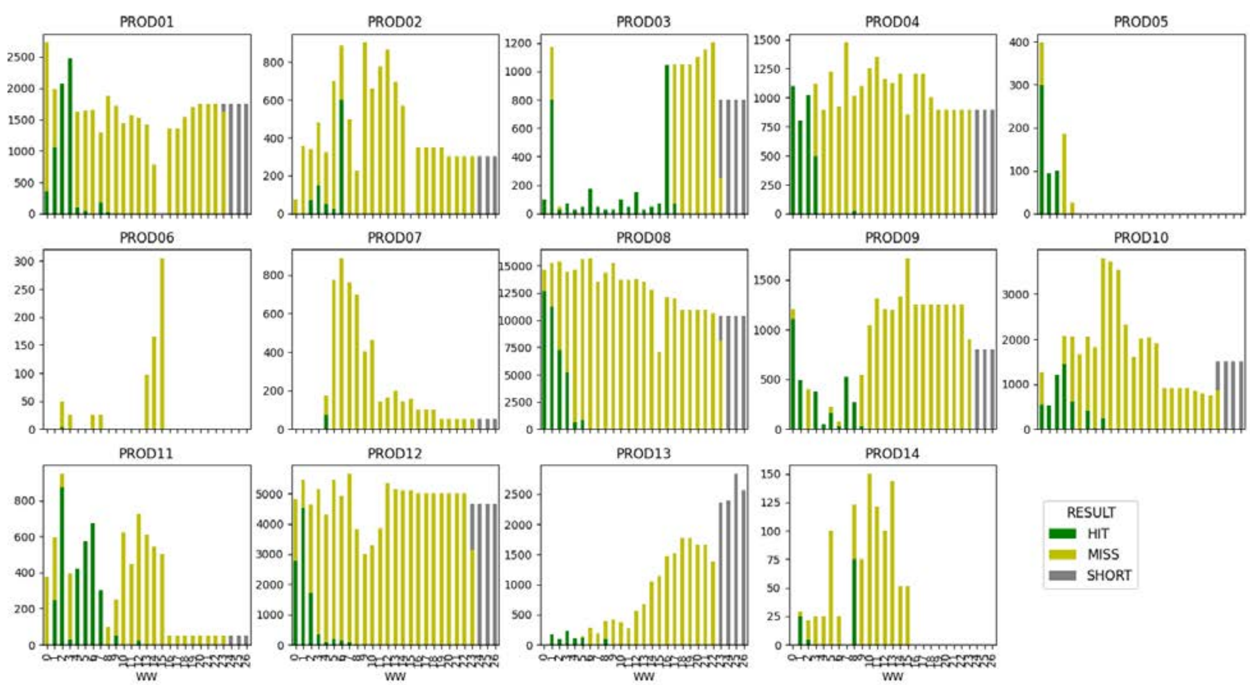
Table 4: Optimization results indicate that 15 workstations need to increase their capacity.

RESOURCE	**CAPA_RATE**
CMP_W12	1.3033
CVD_W06	1.2252
CVD_W17	1.0627
CVD_W18	2.5002
CVD_W19	1.1182
CVD_W26	1.0909
CVD_W39	1.4105
CVD_W48	1.1829
DIFFUSION_W88	1.0173
DIFFUSION_W89	1.2581
DIFFUSION_W95	1.1602
DRY_W10	1.5469
DRY_W41	1.1041
PHOTO_W14	1.0538
WET_W27	1.9589

A simulation is conducted to evaluate the impact of increased capacity with the initial required machine count. The total tardiness improves from 15,291,193 (See Table 3) to 6,697,307. The demand fulfillment chart in Figure 3 helps to assess whether the demand for each product can be achieved. The X-axis is work week over 26 weeks, and the Y-axis is the delivered quantity. Dark green indicates the quantity that met the due date (*HIT* the target), light green indicates the quantity completed with delay (*MISS* the target), and gray indicates the quantity that were not completed during the simulation (*SHORT*). The charts indicate that delivery was on time for the early months but became gradually delayed afterwards.



(a) Original capacity with reduced Eqp-Arrange



(b) Capacity increased by optimization

Figure 3: Demand fulfillment chart indicates that additional capacity reduced SHORT amount as well as total tardiness.

3.3 Optimizing Required Machine Count with PGPE

From the simulation with the initial value of increased capacity for 15 workstations, an additional 26 workstations with at least one day of daily utilization equal to 100% are identified as bottlenecks. We set up a 41-dimensional vector instead of a 461-dimensions and an initial parameter sigma as 0.5 for the PGPE algorithm. The reward considers total tardiness and maximum daily utilization as described in equation (1). The daily utilization term prevents increasing excessive capacity, and the constant c is set to 0.1. Note that the lower bound of each workstation capacity is 1.

$$r = -\Sigma T_j - c * \Sigma(100 - MaxDailyUtilization) \tag{1}$$

The average simulation run time is 19.3 minutes. Each iteration draws 11 samples from the multi-normal distribution—five symmetric pairs and one with the mean value—and runs six simulations in parallel. It takes 36.4 hours to complete 50 iterations. The left graph in Figure 4 indicates that the reward is increasing over 50 iterations. The right graph indicates the total tardiness is improved from 6,697,307 to 18,793; the additional equipment count is increased from 12.4 to 160.5 (this model originally had 1,653 pieces of equipment and 461 workstations).

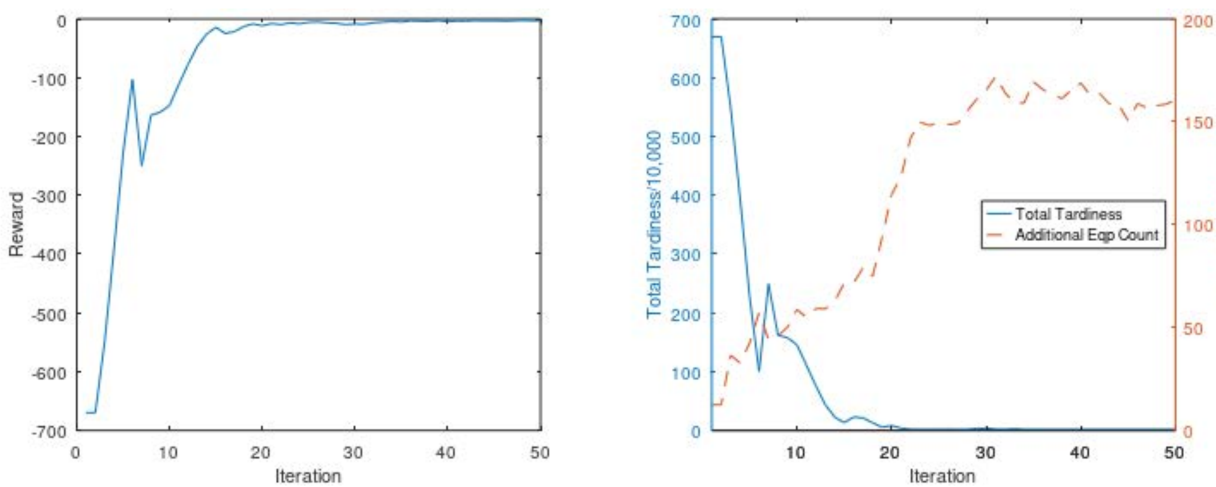


Figure 4: PGPE result indicates that the total tardiness improved over 50 iterations.

More capacity may lead to less tardiness by nature. Table 5 proves that our solution provides the minimum required machine count. OPT_CAPA is an initial value that is the same as the “CAPA_RATE” column in Table 4; PGPE_CAPA is the optimal value determined by machine learning; and MAX_BUSY and AVG_BUSY are maximum and average machine utilization, respectively, which are captured daily from the PGPE result. PGPE_CAPA is generally greater than OPT_CAPA for the 15 workstations. MAX_BUSY indicates that the workstations whose capacity increased are mostly bottlenecks and are a minimal set.

Table 5: Optimal value is mostly greater than the initial value and the results indicate that minimal capacity increases.

RESOURCE	OPT_CAPA	PGPE_CAPA	MAX_BUSY	AVG_BUSY
CMP_W12	1.3033	2.289	100.0	41.3
CVD_W06	1.2252	3.149	100.0	46.7
CVD_W17	1.0627	2.181	100.0	59.6
CVD_W18	2.5002	4.267	100.0	62.5
CVD_W19	1.1182	2.406	100.0	56.5
CVD_W26	1.0909	2.196	100.0	56.4
CVD_W39	1.4105	1.185	100.0	58.1
CVD_W48	1.1829	2.257	100.0	60.4
DIFFUSION_W88	1.0173	3.063	100.0	39.5
DIFFUSION_W89	1.2581	2.216	100.0	60.7
DIFFUSION_W95	1.1602	3.061	100.0	44.6
DRY_W10	1.5469	2.387	100.0	68.8
DRY_W41	1.1041	1.841	100.0	35.1
PHOTO_W14	1.0538	2.123	90.7	33.4
WET_W27	1.9589	1.828	100.0	84.7
CMP_W02		1.763	100.0	62.1
CVD_W07		1.278	100.0	61.2
CVD_W08		2.12	99.8	52.1
CVD_W14		1.114	100.0	82.6
CVD_W20		2.449	100.0	41.5
CVD_W23		1.547	100.0	81.5
CVD_W32		1.428	100.0	72.6
CVD_W35		2.11	100.0	54.2
CVD_W40		3.539	100.0	62.0
CVD_W43		2.695	100.0	40.2
CVD_W44		1.005	100.0	87.5
DIFFUSION_W82		2.035	97.2	57.4
DIFFUSION_W83		1.107	100.0	79.5
DIFFUSION_W90		1.251	100.0	81.2
DIFFUSION_W91		2.388	100.0	42.3
DIFFUSION_W92		2.298	100.0	54.5
DIFFUSION_W94		2.954	99.8	41.4
PHOTO_W02		1.809	100.0	54.5
PHOTO_W05		2.455	90.7	52.3
PHOTO_W07		1.848	100.0	54.8
PHOTO_W08		1.494	100.0	71.3
PHOTO_W19		1.296	100.0	61.7
PHOTO_W23		1.171	100.0	45.6
PHOTO_W32		1.259	100.0	68.0
WET_W21		1.529	100.0	48.9
WET_W77		1.126	100.0	56.6

Figure 5 indicates that most demands are met. For some MISS cases, the initial WIP location was already late to meet the due date. Another simulation that increases the capacity for each and every (461) workstations by 10 times results in a total tardiness of 18,881 (PGPE results 18,793); therefore, PGPE produces a near-optimal solution from a tardiness perspective.

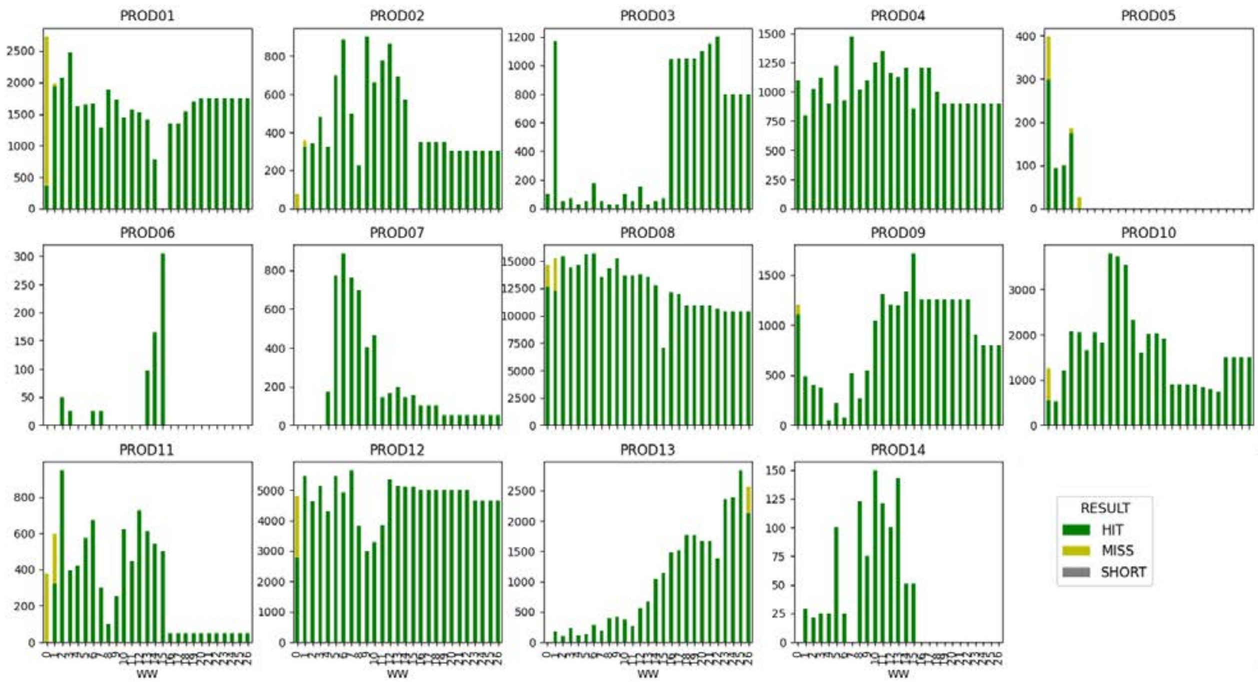


Figure 5: On time delivery is way improved with PGPE result.

4 CONCULUSION AND DISCUSSION

We divided the capacity planning problem into three sub-problems and applied a combination of optimization, simulation, and machine learning techniques. To resolve the unbalanced Eqp-Arrange issue, which is commonly found in real fabs, the optimization method (Optimization 1) was applied and resulted in better step moves and backlogs. Another optimization model (Optimization 2) identified bottleneck machines and provided their initial increased capacity, which reduced the search space for the practically optimal solution as an initial solution. Fab-level simulation verified the optimization result and identified another potential bottleneck machine. PGPE was used to run simulations in parallel and determine the optimal solution, which validated the proposed approach.

In practice, capacity planning is conducted together with planners from each work area. This approach may help these planners determine how many machines are required in various use cases: (1) to consider the budget and unit price for each machine, (2) to review the result if some workstation tool counts are fixed, (3) to justify the purchase of select machines, or (4) to calculate maximum throughput with the current machine set.

This study assumes parallel machines at a given workstation. However, these machines may have different features, processing times and Eqp-Arranges in reality. Further research on the comparison between workstation-level simulation and tool-level simulation are required. Future work should also explore the trade-off between accuracy and run-time of simulations.

REFERENCES

Barahona, F., S. Bermon, O. Günlük, and S. Hood. 2005. "Robust Capacity Planning in Semiconductor Manufacturing". *Naval Research Logistics* 52(5):459-468.

Chen, J. and C. Chen. 2010. “Capacity Planning of Serial and Batch Machines with Capability Constraints for Wafer Fabrication Plants”. *International Journal of Production Research* 48(11):3207-3223.

Geng, N. and Z. Jiang. 2009. “A Review on Strategic Capacity Planning for the Semiconductor Manufacturing Industry”. *International Journal of Production Research* 47(13): 3639-3655.

Karabuk, S. and S. D. Wu. 2003. “Coordinating Strategic Capacity Planning in the Semiconductor Industry”. *Operations Research* 51(6):839–849.

Klemmt, A., M. Romauch, and W. Laure. 2012. “Product Mix Optimization for a Semiconductor FAB: Modeling Approaches and Decomposition Techniques”, *In Proceedings of the 2012 Winter Simulation Conference,* edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 2101-2112, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sehnke, F., C. Osendorfer, T. Rückstieß, A. Graves, J. Peters, and J. Schmidhuber. 2010. “Parameter-Exploring Policy Gradients”, *Neural Networks* 23(4):551-559.

Uzsoy, R., J. F. Fowler, and L. Monch. 2017. “A Survey of Semiconductor Supply Chain Models Part II: Demand Planning, Inventory Management and Capacity Planning”, *International Journal of Production Research* 56(13):4546-4564.

Wang, L., C. Chen, and C. Chueh. 2018. “Development of a Capacity Analysis and Planning Simulation Model for Semiconductor Fabrication”. *The International Journal of Advanced Manufacturing Technology* 99:37-52.

Wu, S. D., M. Erkoc, and S. Karabuk. 2005. “Managing Capacity in the High-Tech Industry: A Review of Literature”. *Engineering Economist* 50(2):125–158.

AUTHOR BIOGRAPHIES

KEYHOON KO is the President of VMS Global, Inc. located in Vienna, VA, USA. He holds a BS from Sungkyunkwan University and an MS and Ph.D. from Korea Advanced Institute of Science and Technology (KAIST), all in industrial engineering. He has led a technical research center for 14 years and is currently responsible for all areas of global business including marketing and sales in the Americas and Europe. He has experience in virtual manufacturing of cover die and mold and in automotive, shipbuilding, display panel, and semi-conductor industries. His email address is fish@vms-solutions.com.

SEOKCHEOL CHANG is a Principal Consultant at VMS Global, Inc. located in Vienna, VA, USA. He holds a BS from Inha University and an MS and Ph.D. from Gwangju Institute of Science and Technology (GIST), Korea, all in industrial engineering. He is interested in planning and scheduling, optimization, simulation and digital twins, and machine learning. His email address is stniron@vms-solutions.com.

WON-JUN LEE is a Principal Researcher of VMS Solutions Co., Ltd. He received a BS, MS and Ph.D. from Korea Advanced Institute of Science and Technology (KAIST), all in industrial engineering. His recent research interests are modeling and simulation; fab scheduling; and machine learning, especially reinforcement learning. His email address is wjlee@vms-solutions.com.

BYUNG-HEE KIM is the President of VMS Solutions Co., Ltd. He received a BS from Sungkyunkwan University and an MS and Ph.D. from Korea Advanced Institute of Science and Technology (KAIST), all in industrial engineering. He is interested in simulation-based scheduling and planning, manufacturing information systems, BPMS, and virtual manufacturing. His email address is kbhee@vms-solutions.com.