Int J Adv Manuf Technol (2016) 84:315–325
DOI 10.1007/s00170-016-8410-1

ORIGINAL ARTICLE

An efficient adaptive dispatching method for semiconductor wafer fabrication facility

Li Li[1] · **Zhihong Min**[1]

Received: 28 May 2015 / Accepted: 20 January 2016 / Published online: 1 February 2016

Abstract To cope with uncertainty in semiconductor wafer fabrication facilities (fabs), scheduling methods are required to produce quick real-time responses. They should be well tuned to track the changes of a production environment to obtain good operational performance. This paper presents an efficient adaptive dispatching method (ADM) with parameters determined dynamically by real-time state information of fabs. ADM is composed of a dispatching rule considering both batch and non-batch processing machines to obtain improved fab-wide performance, several feature selection methods to determine key scheduling-related real-time state information, and a linear regression model to find the relations between the weighting parameters of the dispatching rule and the determined real-time state information. A real fab simulation model is used to demonstrate the proposed method. The simulation results show that ADM is adaptive to changing environment with better performance than a number of commonly used rules (such as FIFO, EDD, CR, LPT, LS, SRPT, and SPT) and an adaptive dispatching rule that considers only real-time ratio of hot jobs to the number of all jobs in a fab and the ratio of jobs with one third of photo steps left to the number of all jobs.

✉ Li Li
lili@mail.tongji.edu.cn

Zhihong Min
zhihongmin@foxmail.com

[1] School of Electronics and Information Engineering, Tongji University, Shanghai, China

Keywords Dispatching · Feature selection · Linear regression · Semiconductor manufacturing

1 Introduction

The advances in semiconductor technology have been accompanied by increased process complexity in semiconductor wafer fabrication facilities, called fabs for short. Due to extreme high capital investment, semiconductor manufacturers demand high overall equipment effectiveness and utilization. The increased process complexity and decreased feature size lead to more frequent off-specification results, job rework, and other uncertainty issues. Consequently, scheduling methods must have the ability to respond quickly to real-time rework and disruption situations. As an effective way, a dynamic dispatching rule has attracted growing attention in both academia and industries. Meanwhile, due to the complexity and strong-coupling relations between upstream and downstream machines, the fab-wide dispatching methods are preferred.

Generally, the dispatching challenges faced by the current fabs are as follows.

(1) There are hundreds even thousands of jobs in a fab. Due to their re-entrant workflows, the workload distributions on workstations are changeable over time. It is necessary to select proper dispatching decision fitting to the real-time running states to achieve better performance.
(2) The downstream and upstream machines are coupled closely due to re-entrant workflows. The movement of jobs on an upstream machine will have impor-

tant impacts on the operational performance of its downstream machines. The dispatching decision of downstream and upstream machines should be coordinated.

(3) There are multiple performance issues to be optimized. Some issues are contradictable, such as on-time delivery rate and cycle time. So it is necessary to make a trade-off between these performance issues.

To satisfy the above requirements, there are considerable related research results. For example, to cope with the complexities for multiple lot scheduling in a semiconductor test facility, Xiong [1] combined the heuristic best-first strategy with the controlled back tracking strategy based on the execution of the Petri nets to reduce set-up times. Lee et al. [2] designed a timed-extended object-oriented Petri net-based multi-objective scheduling method and a real-time dispatching approach to simultaneously optimize the unit profit, tardiness cost, and inventories of WIP and finished-products. Eivazy and Rabbani [3] presented an efficient dispatching method to prioritize the make-to-stock (MTS) and make-to-order (MTO) products in the queue of a workstation whenever a machine in this workstation became idle within a hybrid MTS/MTO production environment. Chiang et al. [4] presented an analytic hierarchy to determine an appropriate set of acceptable WIP deviation levels and the operational job priorities. Altendorfer et al. [5] presented a dispatching rule for multi-product, multi-machine job shops with routing flexibility to maximize the throughput and kept a low level of WIP. Li et al. [6] proposed a dispatching rule to improve the on-time delivery performance for a fab by considering the dispatching of bottleneck machines, non-bottleneck machines, batching machines, and hot jobs. Its full fab perspective made it possible to improve fab-wide operational performance.

Although many efforts are made to improve the dispatching decision of a fab, it is fair to say that a dispatching rule may be suitable to its specific environment only. When it is applied to a different one, however, its performance may significantly deteriorate. Therefore, it is important to select a proper dispatching decision suitable to real-time dynamically changing situations. To do so, machine learning and computational intelligence methods have been increasingly introduced to the dispatching decision processes over time.

For example, [7] developed a hybrid knowledge discovery model that used a combination of a decision tree and a backward propagation neutral network (BPNN) to find an appropriate dispatching rule from the production data and predicted the operational performance. Min and Yih [8] proposed a methodology by combining the simulation and competitive neural network approaches to select proper dispatching rules. Zhang et al. [9] integrated the simulation and response surface methodology to evaluate and optimize the dispatching rules and selected them based on real time system status. Pickardt et al. [10] proposed a two-stage hyper-heuristic for the generation of a set of work center-specific dispatching rules. The approach combined a genetic programming (GP) algorithm that evolved a composite rule from basic job attributes with an evolutionary algorithm (EA) that searched for a good assignment of rules to work centers. The resulting rule sets were robust to most changes in the operating conditions. Wu et al. [11] presented a fuzzy-neural ensemble and geometric rule fusion approach to optimize the performance of job dispatching in a wafer fabrication factory with an intelligent rule. The fuzzy c-means (FCM) and BPN ensemble approach were used to estimate the remaining cycle time of a job, which was an important input to their dispatching rule.

These efforts made a big step for the effective dispatching of fabs. One can select proper dispatching rules according to real-time information, but the original dispatching rules may limit the improved level of a fab. In particular, for the same fab, it is possible to change the original dispatching rules dynamically to cope with changing production environment to obtain better performance. This idea motivates this work to propose an efficient adaptive dispatching method (ADM) with parameters tuned according to real-time state information such that it is adaptive to the changes in a production environment. Different from our previous work on adaptive dispatching rule (ADR) proposed by [12], the real-time state information considered in this paper is determined by feature selection methods instead of those of ADR, i.e., the ratio of hot jobs to work-in-process (WIP) and the ratio of jobs with one third of photo steps left to WIP in a fab. The resultant advantages include the improvement of the operational performance and adaptability to various types of fabs.

The remainder of this paper is organized as follows. The main framework of ADM is introduced in Section 2. In Section 3, the feature selection methods are introduced to determine key scheduling-related real-time state information. Section 4 gives the results of fab simulation. Finally, we present the conclusions and future directions in Section 5.

2 Framework of ADM

To make ADM adaptive to real-time environment, we design a learning-based framework as shown in Fig. 1.

It has four steps:

(1) Generate samples with different weighting parameters of the dispatching rule of ADM and running states with simulations.

(2) Train a BPNN for verification with all samples.

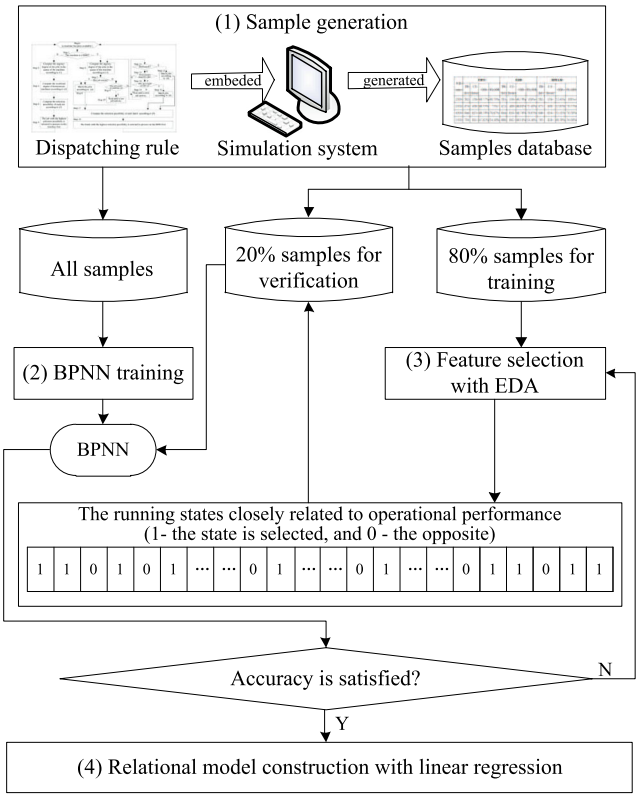

Fig. 1 The framework of the proposed ADM

(3) Select features by using a similarity measurement method, genetic algorithm (GA), and estimation-of-distribution algorithm (EDA) respectively.
(4) Build the relational model between the weighting parameters of the dispatching rule of ADM and running states to make it adaptive to real-time environment.

Therefore, ADM is composed of three main parts, i.e., a dispatching rule, feature selection module, and linear regression model. The workflow of the dispatching rule can be referenced in our early work [12, 13]. For non-batch processing machine, compute the selection possibility S_n of each job n according to

$$S_n = \alpha_1 \left(\frac{{R_i}^n * F_n}{D_n - t + 1} - \frac{{P_t}^n}{\sum_1^n P_i^n} \right) - \beta_1 \frac{\sum {P_{id}}^h}{T_{id}}$$

where

i : index of the available machine
id : index of the downstream machines of machine i
n : index of the jobs in the queue of machine i
t : dispatching decision point, i.e., dispatching time
D_n : due date of job n
F_n : ratio of the average cycle time (the sum of the processing time and queue time) of job n to its processing time
P_i^n : occupation time of job n on machine i
P_{id}^h : occupation time of job h on machine id , which is in the queue of machine id
R_i^n : remaining processing time of job n at machine i
S_n : selection possibility of job n
T_{id} : available time of machine id in one day
α_1, β_1 : measure the relative importance level between the on-time delivery (the first item) and workload balance performance (the second item)

For batch processing machine, compute its selection possibility Γ_k of each batch k according to

$$\Gamma_k = \alpha_2 \frac{{N_{ik}}^h}{B_i} + \beta_2 \frac{B_k}{max(B_k)} - \gamma \frac{{P_i}^k}{max({P_i}^k)} - \sigma \frac{{N_{id}}^k}{\sum_i^k {N_{id}}^k + 1}$$

where

k : index of the batches in the queue of machine i of type BPM
B_i : capacity of machine i of type BPM
B_k : the batch size of batch k
N_{ik}^h : the number of hot jobs in batch k
P_i^k : the occupation time of batch k on machine i
N_{id}^k : the maximum workload of the downstream machines of the jobs in the batch
Γ_k : selection possibility of batch k
α_2, β_2, γ, and σ : measure the relative importance level between those four categories

Obviously, the dispatching rule makes use of the dispatching decision-related information, such as due date of a job, workload of a machine, batch size, and occupation time (including processing, uploading, downloading, setup, and qualification- run time) of a job on a machine. In addition,weighting parameters (α_1, β_1, α_2, β_2, γ, and σ) are chosen for each category of information. They can be tuned to obtain anticipated operational performance. With different values of these weighting parameters and different working states, we can obtain a large number of running cases by simulating a fab with the dispatching rule, which are considered as samples to determine the running states closely related to the operational performance and build the relational model between the weighting parameters of the dispatching rule and those running states. The feature selection module includes several methods, such as a similarity measurement method, GA and EDA. They are used to find running states closely related to the operational performance from samples. The linear regression model is used to build the relational model between weighting parameters of the

Table 1 Parameters of ADM

No.	Attribute	Meaning
1	α_1	The weight of the urgent level of a job processes on the non-BPM
2	β_1	The weight of the workload level of the downstream machines
3	α_2	The weight of the number of hot jobs in a batch
4	β_2	The weight of the batch size of a batch
5	γ	The weight of the processing time of the job on the BPM
6	σ	The weight of the maximum workload of the downstream machines of the jobs in a batch

dispatching rule and the running states closely related to the operational performance.

3 Feature selection methods

There is much real-time state information in a fab. It is difficult to link all kinds of state information to the parameters of ADM. As a result, before building the relations between weighting parameters of ADM and real-time state information, it is necessary to select those having serious impact on fab-wide operational performance. This work considers the following feature selection methods to determine key real-time state information.

(1) Similarity measurement

The following similarity measure methods are investigated.

- Mean square error evaluation (MSEE) method

The mean square error is

$$e_k = \frac{1}{\Omega}\sum_{i=1}^{\Omega}(x_{ki} - x_{oi})^2$$

where

i : the index of samples;

Ω : the number of samples;

x_{oi} : the standard column representing the operational performance;

x_{ki} : the non-standard column representing a kind of state information.

Smaller e_k means a closer relation between the kth running state and the operational performance. This work selects the minimum e_k first. If e_k is larger than 0, select the columns with $min(e_k) \le e_k \le min(e_k)/0.8$ as feature sets; otherwise, select the columns with $min(e_k) \le e_k \le 0.8min(e_k)$ as feature sets.

- Fitting goodness evaluation (FGE) method

The fitting goodness is

$$R^2 = 1 - \frac{SSE}{SST}$$

where $SSE = \sum_{i=1}^{\Omega}(x_{oi} - x_{ki})^2$ and $SST = \sum_{i=1}^{\Omega}(x_{oi})^2 - \frac{1}{\Omega}(\sum_{i=1}^{\Omega} x_{oi})^2$.

Bigger R^2 means better fitting goodness. The column with R^2 closer to 1 means that it more approaches the standard column. This work selects the columns with R^2 being bigger than 0.8 as the feature sets.

- Spectrum analysis method (SAM)

Table 2 Fab-wide attributes

No.	Attribute	Meaning
7	WIP	Fab-wide WIP
8	HotLot	The number of hot jobs in WIP
9	Hotlot %	The ratio of hot jobs in WIP
10	Last_1/3_Photo	The number of jobs with one third photo processes left in WIP
11	Last_1/3_Photo %	The ratio of jobs with one third photo processes left in WIP
12	Bottleneck_M	The number of bottleneck machines
13	Enabled_M	The number of available machines
14	Bottleneck_M %	The ratio of bottleneck machines
15	Utility	The utility rate of the fab, i.e., the ratio between its work time and available time

Table 3 Attributes related to each machine group

No.	Attribute	Meaning
[1..9]6	WIP	WIP of the machine group
[1..9]7	WIP %	The ratio of WIP of the workgroup in fab-wide WIP
[1..9]8	HotLot	The number of hot jobs of the machine group
[1..9]9	Hotlot %	The ratio of hot jobs in WIP of the machine group
[2..10]0	Last_1/3_Photo	The number of jobs with one third photo processes left of the machine group
[2..10]1	Last_1/3_Photo %	The ratio of jobs with one third photo processes left in WIP of the machine group
[2..10]2	Bottleneck_M	The number of bottleneck machines in the machine group
[2..10]3	Enabled_M	The number of available machines in the machine group
[2..10]4	Bottleneck_M %	The ratio of bottleneck machines in the machine group
[2..10]5	Utility	The ratio between its work time and available time of the machine group

SAM makes Fourier transform on both standard and non-standard columns first. Then compute the average of the square of the difference between them. The column with a smaller average value is closer to the standard one.

- Correlation coefficient method (CCM)

The correlation coefficient is

$$\rho_{x_k,x_o} = \frac{Cov(x_k, x_o)}{\sqrt{D(x_k)}\sqrt{D(x_o)}}$$

where

$$Cov(x_k, x_o) = E((x_k - E(x_k))(x_o - E(x_o))$$

$$D(x_k) = E((x_k - E(x_k))^2) = E({x_k}^2) - (E(x_k))^2$$

x_0 : the standard column representing the operational performance;

x_k : the non-standard column representing the running states.

The column with the coefficient approaching 1 means this column has better positive correlation with the standard one.

- Inner product of normalization (IPN) method

The inner product of normalization is denoted as

$$S(x_k, x_o) = \frac{{x_k}^T x_o}{\|x_k\| \, \|x_o\|}$$

The column with bigger $S(x_k, x_o)$ is closer to the standard one.

(2) Genetic algorithm (GA)

The procedure of GA-based feature selection is as follows.

Step 1 : Initialize the parameters of GA, such as popsize, crossover, and mutation probability.

Step 2 : Generate individuals (chromosomes) randomly with binary coding. Their length equals the number of the running states. One gene represents one running state. A gene with 1 value means that this running state is selected as the feature related to the operational performance.

Step 3 : Repeat on this generation till the maximum number of iterations or sufficient fitness is achieved.

① Compute the fitness of each individual by using the BPNN for verification.

② Select individuals for reproduction with Roulette selection algorithm.

③ Breed new individuals through crossover and mutation operations to give birth to offspring.

④ Evaluate the fitness of new individuals.

⑤ Replace old individuals having the worst fitness with new individuals having better fitness to generate a new generation.

Step 4 : Output the individual with the best fitness. Then the genes with 1 value constitute the feature set representing the running states closely related to the operational performance.

Table 4 Attributes related to jobs

No.	Attibute	Meaning
106	Product version	The number of product versions in a fab
107	RP_Step_Avg	The average number of photo processes left in fab-wide WIP
108	RP_Step_Std	The standard deviation of photo processes left in fab-wide WIP

Table 5 Feature set selected by GA from 108 attributes

NA	Index of attributes in the feature set	Accuracy (%)	Time cost (h)
61	1,2,3,6,7,8,9,10,11,12,13,15,17,18,21,22, 25,32,34,35,36,37,38,39,40,42,44,45,46, 50,58,59,62,63,65,66,68,69,70,71,72,73, 74,78,79,80,81,84,85,87,89,91,97,99, 100,102,103,104,105,107,108	87	9
56	1,5,7,9,13,15,17,18,22,23,26,27,30,31,32, 33,34,37,40,43,45,46,47,48,51,55,58,60,61,62, 64,65,66,67,68,69,70,71,73,74,76,77,78,79, 80,82,83,86,90,91,95,96,101,106,107,108	87.9	9
53	1,6,7,8,9,11,14,16,20,21,23,27,29,32,33, 34,35,36,38,42,43,44,49,53,56,57,59,60, 61,62,64,65,68,70,71,72,74,75,77,79,80, 83,87,89,91,93,94,95,96,99,100,105,108	86.4	9.4
49	4,5,8,9,10,11,15,18,22,23,24,26,27,28,35, 37,38,39,41,45,46,48,49,52,54,61,62,64, 66,68,70,75,76,82,85,87,90,91,94,95,96, 97,99,101,102,104,105,107,108	84.6	10.9

(3) Estimation-of-distribution algorithm (EDA)

EDA is similar to GA. The main difference is that it generates new individuals by randomly sampling with a probabilistic model. A Gaussian model is used in this work.

4 Simulation results

A simulation model of an industrial fab is used to validate ADM. Its daily capacity is 7000 slices of WIP. There are nine machine groups in the fab, i.e., implanter, photo, spurting, diffusion, dry etching, wet etching, back thinning, particle vision and measurement (PVM), and stock. Spurting, diffusion, photo, and dry etching machine groups have bottleneck machines. The machines with over 50 % of average utility rate in the fab take about 34.8 %. To decrease the computational cost, we only replace the dispatching rules in those bottleneck machine groups with ADM. According to the practical requirement, we set the movement of the jobs in the fab per day (*Move*) as its performance measurement.

Table 6 Feature set selected by EDA from 108 attributes

NA	Index of attributes in the feature set	Accuracy (%)	Time cost (h)
58	1,2,3,5,10,11,12,13,14,15,18,20,21,22,25,26, 27,28,29,32,33,34,36,37,39,40,43,46, 47,49,51,55,58,61,64,66,68,69,70,71,75, 77,79,82,85,86,88,89,93,96,98,101,102, 103,104,105,107,108	89.1	12.8
55	5,7,9,10,11,13,14,17,18,19,21,23,25,27, 28,32,33,35,36,38,41,44,48,53,55,57,58, 59,67,69,70,72,73,74,75,76,77,78,79,81, 82,83,84,85,88,89,92,93,95,98,100,103, 104,105,108	90.8	9.6
51	3,8,9,21,23,24,28,32,33,35,37,38,41,43, 45,46,47,48,49,50,53,55,57,58,60,62,63, 66,68,70,72,73,76,77,78,80,81,82,83,86, 88,90,91,92,94,95,97,99,102,104,105	85.1	8.9
48	4,5,8,9,10,11,15,18,22,23,24,26,27,28,35, 37,38,39,41,45,46,48,49,52,54,61,62,64, 66,68,70,75,76,82,85,87,90,91,94,95,96, 97,99,101,102,104,105,107,108	88.2	9.6

Table 7 Feature set selected with common similarity measurement methods from 108 attributes

Algorithm	NA	Index of attributes in the feature set	Accuracy (%)	Time cost (s)
MSEE	44	1,7,8,10,11,12,16,18,20,22,23, 26,28,30,32,33,35,36,38,40,42, 46,48,50,52,53,56,57,58,60,61, 62,63,66,68,70,73,76,86,88,96, 106,107,108	88.3	0.04
FGE	44	1,7,8,10,11,12,16,18,20,22,23, 26,28,30,32,33,35,36,38,40,42, 46,48,50,52,53,56,57,58,60,61, 62,63,66,68,70,73,76,86,88,96, 106,107,108	88.3	0.04
SAM	45	1,7,8,10,11,12,16,18,20,22,23, 26,28,30,32,33,35,36,38,40,42, 46,48,50,52,53,56,57,58,60,61, 62,63,66,68,70,73,76,78,86,88, 96,106,107,108	88.0	0.7
CCM	49	4,7,10,11,12,14,15,16,20,21,23,24,25,26, 27,30,31,33,34,35,36,43,44,45,46,47,50,51, 53,54,55,60,61,65,66,67,70,71,73,74,75, 76,77,78,79,85,105,106,107	88.3	0.08
IPN	34	3,4,5,6,7,10,11,12,14,15,22,25, 32,33,34,35,42,52,53,54,55,56, 57,61,62,63,64,65,73,74,75,106,107,108	89.9	0.03

4.1 Candidate feature set

The parameters of ADM are enumerated in Table 1 and 102 attributes in Tables 2, 3, and 4 are related to the performance *Move*. Nine of them are selected in the view of the whole fab as given in Table 2. Ninety of them are selected in relation with machine groups, i.e., ten attributes for each machine group, number of 16–25, 26–35, 36–45, 46–55, 56–65, 66–75, 76–85, 86–95, and 96–105 are for implanter, photo, spurting, diffusion, dry etching, wet etching, back thinning, PVM, and stock workgroups, respectively, as given in Table 3. Three of them are selected in relation with jobs as shown in Table 4.

4.2 Feature section

To select the feature attributes that are the most related to performance *Move*, we generate samples first with the fab simulation system. There are 25 kinds of product versions released to the fab. The simulation time is 90 days. The first 30 days of simulation for warm-up and the last 20 days of simulation are not used for training. The training data is from the middle 40 days of simulation. The values of parameters α_1, β_1, α_2, β_2, γ, and σ are set randomly. We run simulation eight times. Then there are 320 samples for training, where 80 of them are used to test the effectiveness of the feature set selected, and 240 of them are taken as training data. We adopt cross validation to guarantee the training accuracy. Thus, 60 of 240 training samples are also for verification, and 180 of them are for learning.

First, we use a BPNN to predict the value of *Move* with 108 attributes. Then, we select 10 samples randomly to verify its accuracy. The simulation results show that its average accuracy reaches 86.1 %.

Second, we apply GA and EDA to find the feature set closely related to index *Move* with results in Tables 5 and 6.

The average accuracy of GA and EDA for 108 attributes is 86.5 and 88.5 %, respectively. Although they obtain some improvements over BPNN, their time cost is very high. The reason is that there are some redundant ones in 108 attributes, thereby negatively impacting the optimization effectiveness of GA and EDA. It is thus necessary to make attribute reduction.

Third, we apply common similarity measurement methods to reduce such redundancy existing in all 108 attributes. The reduction results are given in Table 7 .

Table 8 Feature set selected by GA from 34 attributes

NA	Index of attributes in the feature set	Accuracy (%)	Time cost (min)
21	5,6,14,22,25,33,35,42,53,54,56,57,61,62, 63,65,73,74,106,107,108	89.2	55
19	3,4,7,14,22,25,32,33,34,35,53,56,57,62, 63,64,73,74,75	87.4	52.5
17	3,5,6,11,14,15,32,33,34,35,53,57,65,74, 106,107,108	86.5	61.4
15	3,4,10,11,12,25,32,33,34,53,55,56,61,64, 65,73,74	88.1	56.9

Table 9 Feature set selected by EDA from 34 attributes

NA	Index of attributes in the feature set	Accuracy (%)	Time cost (min)
22	4,5,6,7,10,11,12,14,15,32,33,34,35,42,52, 55,57,62,65,74,106,107	86.5	49.6
19	4,7,12,15,22,33,34,35,42,54,55,57,61,62, 63,65,73,75,106	86.2	55.6
17	7,14,25,32,33,34,35,42,53,55,57,63, 65,74,75,106,108	92.3	53.3
15	4,7,15,33,34,42,52,53,57,62,73,74,106, 107,108	87.2	62

It is shown that the average accuracy of these common similarity measurement methods for feature selection with lots of attributes is better than that of GA and EDA. The result of IPN is the best. Hence, we use the 34 attributes found by it to build up training samples for GA and EDA to make feature selection. The results are shown in Tables 8 and 9. It is shown that the average accuracy of GA and EDA for 34 attributes obtained by IPN is 88.9 and 88 %, respectively. It is better than that of the prior cases using 108 attributes. In addition, the time cost decreases significantly. It is thus useful to apply GA and EDA to make feature selection for those attributes reduced with similarity measurement methods.

Fourth, we further make attribute reduction on 34 attributes obtained by IPN with similarity measurement methods to achieve further reduced feature set. The results are shown in Table 10.

It is shown that the average accuracy of these similarity measurement methods for feature selection with less attributes (34 attributes) is unsatisfied, lower than that of GA and EDA. As a result, we select 17 attributes obtained by EDA as the feature set most related to performance *Move*.

However, 17 attributes seem to be still excessive to build a rational model between them and the parameters of ADM. Then, we make additional attribute reduction. First, the utility of a fab and each machine group is set as the utility rate between consecutive 2 days. In applying ADM to a real fab, these attributes (i.e., 25, 35, 55, 65, and 75) are difficult to obtain. We delete these attributes from the feature set. Secondly, for a real fab, the number of bottleneck machines and available machines and the ratio of bottleneck machines in a machine group are stable. Thus, we delete attributes 32, 33, 34, 42, 53, 63, and 74. Thirdly, in the remaining attributes, only attribute 57 is related to a machine group. Others are the fab or job related. Thus, we delete attribute 57. Finally, we obtain a four-attribute feature set (7, 14, 106, and 108). The resulting accuracy is about 90.6 % and satisfactory.

In addition, the results of feature selection show that the parameters of ADM are closely related to the performance *Move*. It is necessary to build a relation model between the selected attributes and them.

*NA: the number of attributes related to *Move*

4.3 Relational model

The feature set {7, 14, 106, 108} represents the number of fab-wide WIP, the ratio of bottleneck machines in a fab, the number of product versions in a fab, and the standard deviation of the number of photo processes left in fab-wide WIP, respectively. We apply a linear regression method to the training samples to match the parameters with feature set {7, 14, 106, 108}. The model obtained is as follows:

$$\alpha_1 = 0.000065 * V_7 + 0.27 * V_{14} - 0.0314 * V_{106} - 0.137 * V_{108} + 1.178$$

$$\beta_1 = -0.000065 * V_7 - 0.27 * V_{14} + 0.0314 * V_{106} + 0.137 * V_{108} - 0.178$$

Table 10 Feature set selected with common similarity measurement methods from 34 attributes obtained by IPN

Algorithm	NA	Index of attributes in the feature set	Accuracy (%)	Time cost (s)
MSEE	25	4,6,7,10,11,12,22,32,33,34,35, 42,52,53,55,56,57,61,62,63,73, 74,75,106,107,108	85.6	0.02
FGE	24	6,7,10,11,12,22,32,33,34,35,42, 52,53,55,56,57,61,62,63,73,75, 106,107,108	83.7	0.01
SAM	25	4,6,7,10,11,12,22,32,33,34,35,42,52,53,55,56,57,61,62,63,73, 74,75,106,107,108	85.6	0.01
CCM	24	4,7,10,11,12,14,15,25,32,33,34,35,52,53,54,55,61,62,65,73,74, 75,106,107,108	86.1	0.02
IPN	24	6,7,10,11,12,14,15,22,32,35,52,53,54,55,61,62,63,65,73,74,75, 106,107,108	87.4	0.02

Fig. 2 Comparison Table of ADM with samples on performance *Move*

WIP (pieces)	*Avg_Move* (pieces)			*Best_Move* (pieces)			*Worst_Move* (pieces)		
	Samples	ADM	Improve (%)	Samples	ADM	Improve (%)	Samples	ADM	Improve (%)
> 9000	36543	37758	3.32	40847	40013	-2.04	29145	35775	22.75
8000 +	34368	35215	2.46	45954	42982	-6.47	21872	29751	36.02
7000 +	28907	31166	7.81	40808	37332	-8.52	23823	27581	15.77
6000 +	28418	28999	2.04	35803	40020	11.78	22204	22477	1.23
5000 +	26884	28354	5.47	34442	30153	-12.45	20925	28225	34.89
<5000	27990	30799	10.04	32300	34281	6.13	23125	24704	6.83

$$\alpha_2 = 0.000021 * V_7 + 0.07 * V_{14} - 0.0099 * V_{106} - 0.046 * V_{108} + 0.373$$

$$\beta_2 = 0.000029 * V_7 + 0.16 * V_{14} - 0.0138 * V_{106} - 0.058 * V_{108} + 0.522$$

$$\gamma = -0.000006 * V_7 - 0.01 * V_{14} + 0.0022 * V_{106} + 0.009 * V_{108} + 0.296$$

$$\sigma = -0.000046 * V_7 - 0.24 * V_{14} + 0.0220 * V_{106} + 0.099 * V_{108} - 0.194$$

where (V_7, V_{14}, V_{106}, V_{108}) are the values of feature set {7, 14, 106, 108}.

Then, we run simulations with ADM whose parameters α_1, β_1, α_2, β_2, γ, and σ are set as the model at 7000+ piece WIP level. The simulation results show that average *Move* of ADM is 31,740 pieces and that of the samples are 30,876 pieces. It means that ADM improves performance *Move* by 2.80 %. Furthermore, we prove that ADM is adaptive to variable environments with different workload (i.e., 5000−, 5000+, 6000+, 7000+, 8000+, and 9000+ pieces WIP levels). The simulation results are shown in Fig. 2. We can obtain the following conclusions from them. Comparing to average *Move* of the samples whose parameters are set randomly, it is improved by 2.04 % at least and 10.04 % at most with ADM. Although the best *Move* obtained by ADM is a little less than that of the samples, the worst one obtained by ADM is always much better (36.02 % at most) than that of the samples. Therefore, ADM is not only adaptive to environments with better performance but also able to smooth fab-wide WIP flows.

To compare ADM with the existing ones, we conduct vast simulations with common rules including first-in-first-out (FIFO), critical ratio (CR), earliest due date (EDD), longest processing time (LPT), least slack (LS), shortest remaining processing time (SRPT), shortest processing time (SPT), and our prior method, i.e., ADR, proposed by [12]. We build the relational models for each WIP level (i.e., 5000, 6000, 7000, 8000, and 9000 pieces). The release policy adopts a constant-WIP (CONWIP) method. The simulation time is 70 days. The first 30 days of simulation for warm-up and the last 40 days of simulation are used for training and comparison. The simulation results are shown in Fig. 3.

We can draw the following conclusions.

(1) Common rules (such as FIFO, CR, EDD, LPT, LS, SRPT, and SPT) achieve different performance for various WIP level. However, ADR and ADM can generally achieve better *Move* than them, especially for the workload at the [85 %, 115 %] range of the fab's capacity. For example, comparing to CR, respectively, at the 6000 piece WIP level, ADR and ADM improve *Move* by 4.65 and 6.90 %; at the 7000 piece WIP level, they improve *Move* by 5.74 and 10.01 %; at the 8000 piece WIP level, they improve *Move* by 8.66 and 9.69 %. Therefore, ADR and ADM are clearly more adaptive to changing environments than common rules.

(2) When the fab is seriously under-loaded (e.g., at the 5000 piece level, about 30 % underloaded) or overloaded (e.g., at the 9000 piece level, about 30 % overloaded), the improvements of ADR and ADM become less. For example, comparing to CR, at the 5000 piece WIP level, ADR and ADM improve *Move* by 3.04 and 3.24 %, respectively; at the 9000 piece WIP level, they improve *Move* by 3.63 and 5.35 %, respectively. They are because sufficient resources in the former reduce the differences among all the dispatching rules and too many jobs in the latter drastically narrow down the scheduling choices.

Fig. 3 Comparison Table of ADM with common rules and ADR on performance *Move*

WIP (pieces)	*Move* (pieces)										*Improvements comparing to CR* (%)									
	Samples	CR	EDD	FIFO	LPT	LS	SRPT	SPT	**ADR**	**ADM**	Samples	CR	EDD	FIFO	LPT	LS	SRPT	SPT	**ADR**	**ADM**
9000	30576	30507	29757	31286	30523	29455	28402	31494	**31434**	**31496**	0.23	0.00	-2.46	2.55	0.05	-3.45	-6.90	3.24	**3.04**	**3.24**
8000	31852	30430	29511	30983	30367	29517	29293	31444	**33064**	**33378**	4.67	0.00	-3.02	1.82	-0.21	-3.00	-3.74	3.33	**8.66**	**9.69**
7000	31589	30194	29550	31065	30449	29818	29033	30993	**31929**	**33216**	4.62	0.00	-2.14	2.89	0.84	-1.25	-3.85	2.65	**5.74**	**10.01**
6000	30821	29917	29930	31000	30944	30018	29183	31284	**31308**	**31981**	3.02	0.00	0.04	3.62	3.43	0.34	-2.45	4.57	**4.65**	**6.90**
5000	30541	30701	31021	31118	31010	30642	30104	31258	**31815**	**32344**	-0.52	0.00	1.05	1.36	1.01	-0.19	-1.94	1.82	**3.63**	**5.35**

Table 11 Improvements of ACM comparing with different dispatching rules at 95 % confidence interval

	ADM-CR	ADM-EDD	ADM-FIFO	ADM-LPT	ADM-LS	ADM-SRPT	ADM-SPT	ADM-ADR
Interval	[1056,3209]	[1087,3970]	[286,2498]	[600,3047]	[1389,3795]	[2238,4321]	[62,2313]	[0.3,1145]

(3) For various WIP levels, ADM always performs better than ADR. The average improvement of ADM on *Move* reaches 1.89 %. The largest improvement of ADM happens at the relatively heavier load, e.g., 7000 pieces WIP level. It improves *Move* by 4.27 % comparing to ADR. It proves that ADM can fully utilize the capacity of the fab. The smallest improvement (only 0.2 %) of ADM happens at the over-loaded case, i.e., at the 9000 piece WIP level, due to a smaller scheduling space. The superiority of ADM depends on its feature selection based on real-time states, while those of ADR are the ratio of hot jobs to WIP and the ratio of jobs with one third of photo steps left to WIP in a fab (set directly with the managers' experience and without a feature selection process). Thus, it is necessary to find the real-time states closely related to performance with proper feature selection methods to guarantee an adaptive rule's effectiveness.

4.4 Further analysis

In this set of experiments, the number of WIP changes from 5000 to 9000 with 1000 difference. Using the data in Fig. 2, 95 % confidence intervals for the true mean differences in *Move* are calculated. The approximate $100(1-\alpha)$ % confidence interval for X_n is defined as

$$\bar{X_n} \pm A\frac{S_n}{\sqrt{n}}$$

where X_n is the increased number of *Move* of ADM comparing to different dispatching rules, i.e., the samples; $\bar{X_n}$ is a sample mean of X_n based on a sample of size n (here n=5); S_n is the standard error of $\bar{X_n}$; A is the $100(1-\alpha)$ % (here $\alpha = 0.005$) percentage point of a t-distributed with n-1 degrees of freedom. The value of A is set to 2.776 in t-distribution table produced by [14]. The comparison results are shown in Table 11.

The 95 % confidence interval for *Move* lies completely above zero, which provides strong evidence that the proposed ADM is better than other dispatching rules, because its average Move is higher.

5 Conclusions

In this paper, an adaptive dispatching method (ADM) is proposed. It has adjustable weighting parameters to take into account real-time running state information in a fab. The feature selection methods are used to decide the most relevant information used to adapt the dispatching rule. The simulation results on a high-fidelity simulated industrial fab indicate that the proposed ADM is an effective and adaptive way to improve fab-wide operational performance.

It is noted that ADM proposed in this paper fits to improve the *Move* performance. If one wants to improve other performance indices, the same workflow should be performed while selecting the samples with better specified performance indices to build a suitable parameter model to achieve the objective.

The main deficiency of this work is the lack of consideration on process constraints and machine preventive maintenance schedules. Thus our future work is to investigate the scheduling problems with special process constraints, proposed by [15] and [16], and active machine preventive maintenance. We intend to pursue a good trade-off between timely handling of hot jobs and desired performance of *Move* and yield under process constraint requirements.

Acknowledgments This paper is supported in part by the National Nature Science Foundation of China under grants No. 51475334.

References

1. Xiong HH, Zhou MC (1998) Scheduling Of semiconductor test facility via petri nets and hybrid heuristic search. IEEE Trans Semicond Manuf 11(3):384–393
2. Lee YF, Jiang ZB, Liu HR (2009) Multiple-objective scheduling and real-time dispatching for the semiconductor manufacturing system. Comput Oper Res 36(19):866–884
3. Eivazy H, Rabbani M (2009) An efficient dispatching method in the hybrid make-to-stock/make-to-order semiconductor manufacturing systems. In: Proceeding International Conference on Computers and Industrial Engineering, vol 46, pp 1763–1768
4. Chiang DM, Guo RS, Pai FY (2008) Improved customer satisfaction with a hybrid dispatching rule in semiconductor back-end factories. Int J Prod Res 46:4903–4923
5. Altendorfer K, Kabelka B, Stocher W (2007) A new dispatching rule for optimizing machine utilization at a semiconductor test field. In: Proceeding IEEE/SEMI Advanced Semiconductor Manufacturing Conference, vol 46, pp 188–193
6. Li L, Qiao F, Jiang H, Wu QD (2004) The research on dispatching rule for improving on-time delivery for semiconductor wafer fab. In: Proceeding 2004 8th International Conference on Control, Automation, Robotics and Vision (ICARCV), vol 46, pp 494–498
7. Li L, Qiao F, Jiang H, Wu QD (2004) A Hybrid knowledge discovery model using decision tree and neural network for selecting

dispatching rules of a semiconductor final testing factory. Prod Plan Control 16:665–680
8. Min HS, Yih YW (2003) Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system. Int J Prod Res 41:3921–3941
9. Zhang H, Jiang ZB, CT (2009) Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology. Int J Adv Manuf Technol 41:110–121
10. Pickardt CW, Hildebrandt T, Branke J, Heger J (2013) Evolutionary generation of dispatching rule sets for complex dynamic scheduling problemss. Int J Prod Econ 145:66–77
11. Wu HC, Chen T, Branke J, Heger J (2013) A Fuzzy-neural ensemble and geometric rule fusion approach for scheduling a wafer fabrication factory. Math Probl Eng 10(1155):956–978
12. Li L, Sun ZJ, Zhou MC, Qiao F (2013) Adaptive Dispatching rule for semiconductor wafer fabrication facility. IEEE Trans Autom Sci Eng 10:354–364
13. Li L, Sun ZJ, Ni JC, Qiao F (2013) Data-based scheduling framework and adaptive dispatching rule of complex manufacturing systems . Int J Adv Manuf Technol 66:1891–1905
14. Banks J, Carson JSII, Nelson BL, Nicol DM (2013) Discrete-event system simulation, vol 66. Prentice-Hall, Saddle river, pp 1891–1905
15. Hu H, Zhou MC, Li Z, Tang Y (2013) Deadlock-free control of AMS with flexible routes and assembly operations using petri nets. IEEE Transactions on Industrial Informatics 9: 109–121
16. Qiao Y, Wu N, Zhou MC (2014) Scheduling of dual-arm cluster tools with wafer revisiting and residency time constraints. IEEE Transactions on Industrial Informatics 10:286–300