Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

LEARNING-BASED ADAPTIVE DISPATCHING METHOD FOR BATCH PROCESSING MACHINES

Long Chen
Hui Xu
Li Li

College of Electronics and Information Engineering
Tongji University
Shanghai, 201804, CHINA

Lu Chen

Shanghai Belling
Shanghai, 200233, CHINA

ABSTRACT

This study aims to solve the scheduling problem of batch processing machines (BPMs) in semiconductor manufacturing by using a learning-based adaptive dispatching method (LBADM). First, an adaptive ant system algorithm (AAS) is proposed to solve the scheduling problem of BPMs according to their characteristics. Then AAS generates a lot of solutions for the jobs with different distribution of arrival time and due date. These solutions are taken as learning samples. Second, we analyze influencing factors by sample learning method from those solutions. With the help of linear regression, the coefficients of influencing factors can be calculated to build a dynamic dispatching rule adaptive to running environments. Finally, simulation results based on a Minifab model show that the proposed method is better than traditional ways (such as FIFO and EDD with maximum batchsize) with lower makespan and weighted tardiness.

1 INTRODUCTION

There is a batch processing style in semiconductor manufacturing, such as the diffusion machines and oxidation machines, which occupy 20-30% of the total number of machines. They can process more than one job together and the processing time is independent of the number of the jobs (Mathirajan et al. 2006). A good scheduling method can use the ability of batch processing machines (BPMs) efficiently to achieve expected performance while satisfying the constraints of maximum batch size and batching style.

In recent years, there have been many studies on BPMs scheduling problem in different manufacturing processes according to Li et al. (2012). They can be classified into static scheduling and dynamic scheduling.

In static scheduling, the swarm intelligence algorithm is an effective method to achieve better performance. For example, Wagner and Affenzeller (2006) implemented a lot of simulations on an intelligent simulation platform called HeuristicLab to demonstrate on intelligent algorithm such as ant colony optimization algorithm (Dorigo et al. 1996) can be used to solve the scheduling problems of parallel machines. Raghavan et al. (2006) used ACO to solve parallel batch processors problem with incompatible job families. Li et al. (2009) have determined that ACO-based scheduling of BPMs performed well in minimizing the total weighted tardiness, and also used in feeding control, batch size forming, the maintenance of machines as well as the bottleneck machines scheduling process (Li 2012; Chang et al. 2012).

The research on dynamic dispatching includes the composite priority dispatching strategy in feeding control (Wang et al. 2007), a hybrid ACO approach for a BPM (Guo et al. 2010), dynamic bottleneck dispatching (DBD, Zhang et al. 2009), the heuristic dispatching rules forecasting the movement of jobs to reduce waiting time (Solomon et al. 2002; Wang 2009), the ACO combining dynamic programming method to minimizing the makespan (Shao 2010), and ACO-based predictive analytics for case-oriented (Lakshmanan et al. 2011), dispatching rules considering machines and emergency jobs (Li et al. 2011) as

978-1-4799-2076-1/13/$31.00 ©2013 IEEE

well as adaptive learning in scheduling model (Noroozi et al. 2013) in the aspect of production line dispatching rules.

The research on the scheduling of BPMs has made a big step. However, we found it was hard for static scheduling to respond to the dynamic operating environment though it had optimization ability. As for dynamic scheduling, it focused on local optimization rather than global optimization. To satisfy a near-optimal optimization requirement and being adaptive to real-time environment, this paper proposes an adaptive method for solving scheduling problems of BPMs combined with parameter learning. The remainder of this paper is organized as follows. In Section 2, we present the problem description and assumptions. Then we outline the learning-based adaptive dispatching method in Section 3. In Section 4, we show computational experiments and results. Finally, we give conclusions and future research topics in Section 5.

2 PROBLEM DESCRIPTION

When scheduling BPMs, we should consider the difference among arrival time or due date of jobs, the family distribution of the jobs, the states of machines, current process of the jobs in a batch, and so on. Batches may be adjusted according to the actual situation.

To facilitate the discussion, the assumptions involved in the BPMs scheduling problem include (Li et al. 2011):

a. The machines in the parallel batch processing machines (PBPMs) are the same ones.
b. The scheduling of the jobs from multiple families is decided.
c. The processing time of a batch on one machine is independent of the number of the jobs in the batch.
d. Once processing begins on a batch, no job can be removed from or added to the machine until it finishes.
e. There are sequence-dependent random setup times for changeovers between jobs from different families, and no setup time between jobs from the same family.
f. The targets to optimize are the total weighted tardiness (TWT) and makespan. Then the objective function of the batch processing scheduling problem can be denoted as

$$min(\sum\sum w_{ij}T_{ij} + w_m max(C_m)) . \tag{1}$$

where $Max(C_m)$ is the maximum makespan of PBPMs; C_m is the makespan of BPM m; T_{ij} is the tardiness of job j from family i defined as the difference between its finish time and due date, and T_{ij} will be zero if job j is not late; w_{ij} is the weight of the tardiness of job j from family i, which is set as the reciprocal of the difference between its due date and arrival time; w_m is the weight of the maximum makespan.

3 LEARNING-BASED ADAPTIVE DISPATCHING METHOD

3.1 Framework

According to the characteristics of the scheduling of BPMs, we designed an learning-based adaptive dispatching method (LBADM), composed of three main parts, i.e., an adaptive ant system algorithm (AAS) to obtain optimal solutions for BPMs scheduling, a scheduling rule for real-time dispatching of BPMs, and the linear regression-based parameter setting method to let the scheduling rule be adaptive to various environments. Its framework is shown in Figure 1.

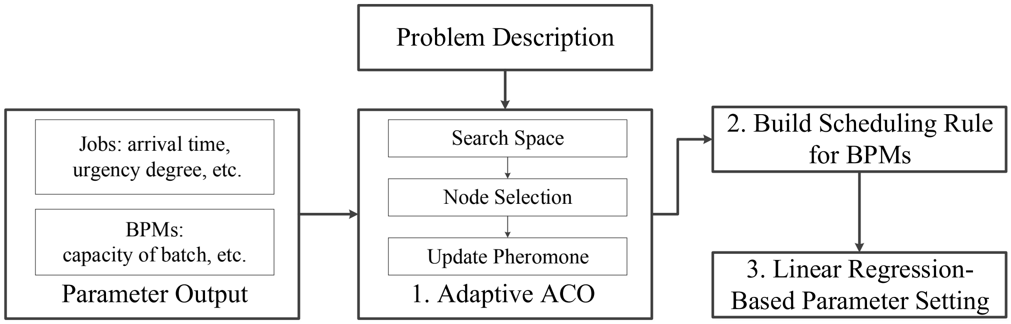


Figure 1: The Framework of LBADM.

3.2 Adaptive Ant System Algorithm (AAS)

The main workflow using AAS to obtain an optimal solution for BPMs scheduling includes two stages, i.e., the establishment of search space and finding an optimal solution with AAS.

3.2.1 Establishment of Search Space

Before we use AAS to find a solution, the first task is to build a search space for it. In this paper, the search space is composed of nodes which are combinations of the batches and the BPMs.

For L jobs, there are $C_L^1+ C_L^2+...+ C_L^L$ (C is the combination operator) batching styles, subject to the constraint of maximum batch size. In the case of many jobs (especially with a number of dynamic arrival jobs), this kind of batching style will result in lower computation efficiency. In this paper, we form the batches using the time window concept, denoted as

$$B(j,t,\Delta t)=\{ij \mid A_{ij}\le(t+\Delta t)\}\ . \tag{2}$$

That is, if job j from family i arrives (A_{ij}) in the time limit of Δt, job j then can be placed into the batch. Then the arrival time of a batch can be described as follows:

$$A_B=\max(A_1,A_2,...,A_L)\ . \tag{3}$$

Equation (3) means the last arriving job of a batch is selected as the arrival time of the batch.

3.2.2 Finding an Optimal Solution with AAS

There are three modules for finding an optimal solution with AAS, i.e., initialization, search process of AAS and update of pheromone. The flowchart is shown in Figure 2.

3.2.2.1 Initialization

The scale of artificial ants is settable, such as the number and generation. Task list and tabu list is empty at the beginning. The elements of phenomenon matrix are initialized as the same small positive value.

3.2.2.2 Search process of AAS

AAS dynamically adjusts its weighting values of pheromone in the selection process of the artificial ants. On the one hand, the probability of a randomly selection can be properly increased to avoid stagnation or local optimum, and on the other hand, it's useful for following ants to find a better solution after parameter adjustments, which will improve the ability of the algorithm to converge.

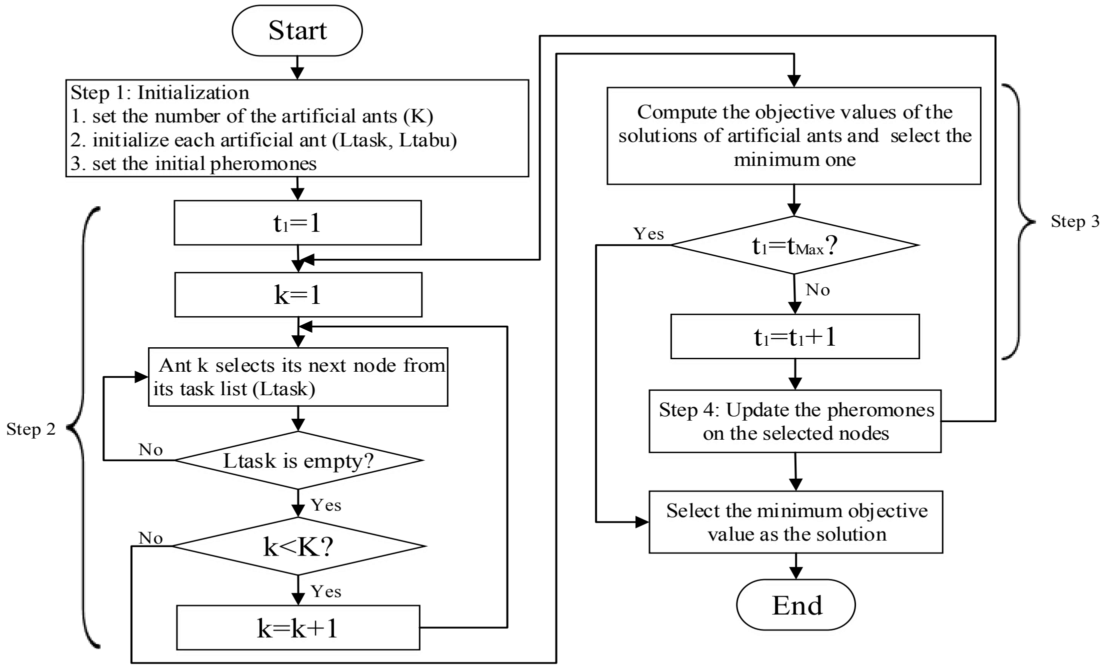


Figure 2: The Flowchart of the AAS.

The workflow of finding a solution with AAS is shown in Figure 2. As to the next node to select, each node in the task list (L^K_{task}) has a certain probability to be selected except those in tabu list (L^K_{tabu}), such as the nodes containing the same job from the same family on different machines. If one is selected, the others should be put into tabu list. Assuming the ant *K* moving from node *i* to *j* with the probability that is calculated from the following pseudo-code (Figure 3).

In the pseudo-code, α and β are parameters representing the relative importance of the pheromone and heuristic factor of AAS. Parameter τ_{ij} is the value of pheromone between node *i* and node *j*. The heuristic factor η_{ij} represents the arrival time, processing time, set up time and remaining processing time of jobs, etc. se results the nodes making minimal tardiness and makespan will have a greater probability to be selected.

```
/*Program of Node Selection */
/* Collection of nodes can be accessed for ant K */
α = random.Next(0,1); β = random.Next(0,1);
η[i, j] = A_B + Pt_B + Ts_B + Ud_B;
v[i, j] = (τ[i, j] ^ α) * [ (batch_Num * 1.0 / η[i, j]) ^ β];
sum+ = v[i, j];
P[i, j] = v[i, j] / sum;
/* Collection of nodes in tabu list (such as the same job from the same family on the other machine) */
P[i, j] = 0.0;
```

Figure 3: The pseudo code of node selection of AAS.

3.2.2.3 Update of Pheromone

Each ant will randomly choose a next node according to the density of its pheromone. According to the local update strategy in adaptive adjustment, the update of pheromone will take place in the nodes selected after completing search in one generation. The algorithm will record and compare the value of the objective function in each generation. The corresponding pseudo-code is as follows (Figure 4):

/* Update Pheromone */

$$\rho = value;\ \ Q = Const;$$

$$Obj\_Fun = \sum_i \sum_j w_{ij} T_{ij} + w_m max(C_m);$$

$$\tau[i,j] = (1-\rho) * \tau[i,j] + Q / Obj\_Fun;$$

Figure 4: The pseudo code of pheromone update strategy.

where ρ means the pheromone evaporation between t and $t+1$. At the initial time, $\tau_{ij}(0) = c$, in which c is a constant as well as Q. The constant Q will balance the order of magnitude.

3.3 Build scheduling rule for BPMs

Considering the characteristics of BPMs scheduling, we consider factors of a batch, including its waiting time before the machine, due date urgency degree, occupancy time on the machine and batch size, to build the scheduling rule for BPMs. The processing priority of batch B is defined as:

$$P_B = a * Wt_B + b * Ud_B + c * Ot_B + d * Bs_B \tag{4}$$

where

P_B- the processing priority of batch B;

Wt_B- waiting time of batch B. When learning the parameters, it is calculated as $Wt_B = St_B - A_B$, i.e., the time difference between its start time on the machine (St_B) and arrival time before the machine (A_B);

Ud_B- urgency degree of batch B. When learning the parameters, it is calculated as $Ud_B = \frac{(D_B - A_B)}{St_B}$, i.e., the ratio of the difference between its due date D_B and arrival time before the machine and its start time on the machine;

Ot_B- occupancy time of batch B, $Ot_B = Pt_B + Lt_M + Ut_M$, i.e., the sum of its raw processing time (Pt_B), load (Lt_M) and unload time (Ut_M) on the machine;

Bs_B- the batch size of batch B; and

a, b, c, d- the coefficients of those four information categories, i.e., a as the coefficient of waiting time of batch B (Wt_B), b as the coefficient of urgency degree (Ud_B), c as the coefficient of occupancy time (Ot_B), and d as the coefficient of the batch size (Bs_B).

3.4 Parameter Learning Method

The goal of parameters learning is to determine how the optimal results distribute in the solution set under the changes in parameters. These samples will be extracted from the distribution of optimal solution, and corresponding coefficients can be obtained through linear regression. The pseudo-code of the linear regression is demonstrated in Figure 5:

```
/* linear regression */
/* tour[i] is the dispatching order decided by adaptive ACO */
```
$Y[i] = tour[i];$

/* these four parameters (Wt_B, Ud_B, Ot_B, Bs_B) are got from adaptive ACO */

$X[i,0] = Wt_B; X[i,1] = Ud_B; X[i,2] = Ot_B; X[i,3] = Bs_B;$

/* get the coefficients according to $Y = X * \beta_{co}$ */

$\beta_{co} = (X^T * X)^\wedge(-1) * X^T * Y$

$a = \beta_{co}[0]; b = \beta_{co}[1]; c = \beta_{co}[2]; d = \beta_{co}[3];$

/* dispatching on parameter learning method */

/* these four parameters (Wt_B, Ud_B, Ot_B, Bs_B) are got from LBADM */

$X_L[i,0] = Wt_{BL}; X_L[i,1] = Ud_{BL}; X_L[i,2] = Ot_{BL}; X_L[i,3] = Bs_{BL};$

/* get the dispatching order */

$Y_L = X_L * [a,b,c,d]^T$

Figure 5: The pseudo code of parameters learning method.

4 SIMULATION RESULTS

We use Minifab, a simplified model based on the actual production line from of a semiconductor production line with three equipment groups and five machines (shown in Figure 6, Fowler 1994) to validate and verify the proposed learning-based adaptive dispatching rule.

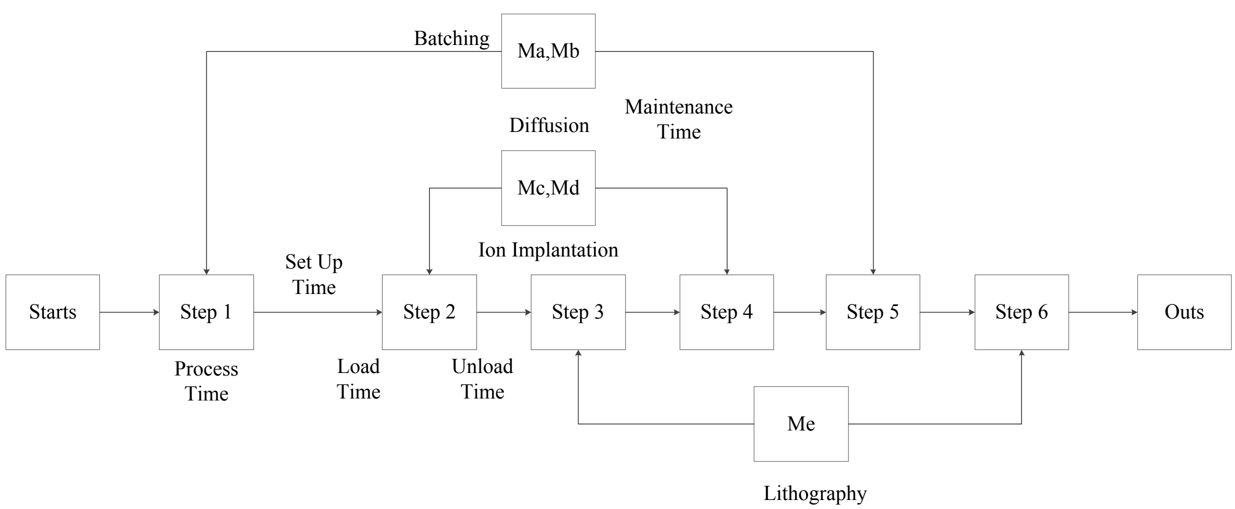
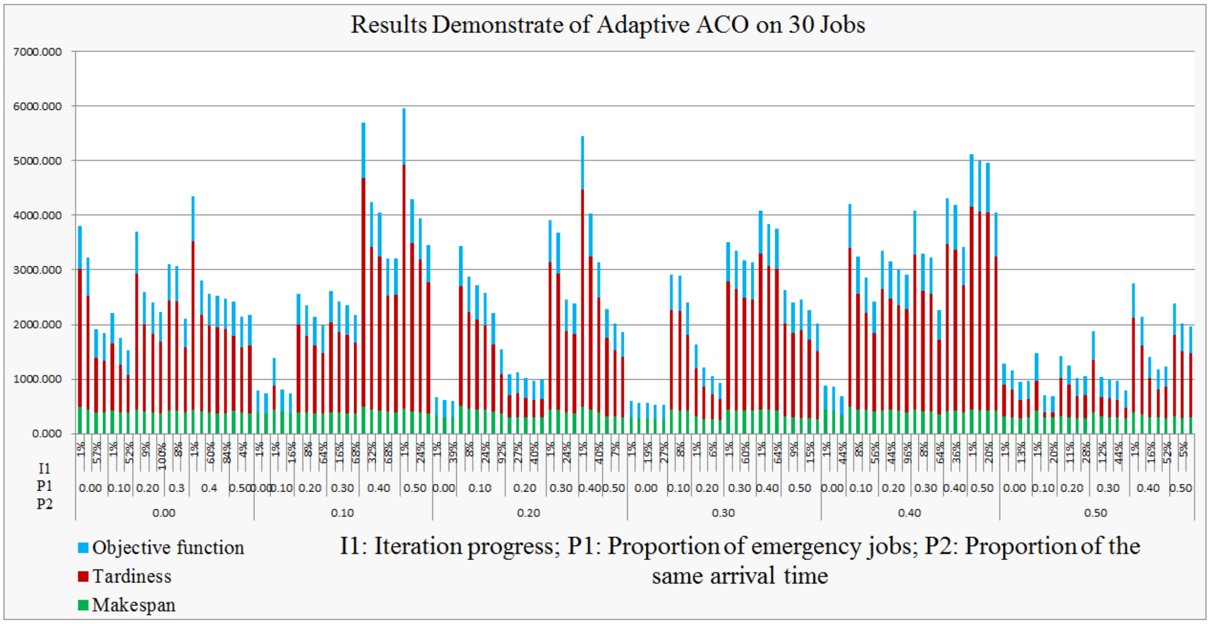


Figure 6: The Model of Minifab.

Minifab contains PBPMs and non-BPMs. The proper use of the intelligent algorithm in batch processing scheduling can significantly improve the scheduling result.

Figure 7 shows the convergence of adaptive ant colony algorithm. The simulation results show that the ant colony algorithm with adaptive parameter regulation can obtain objective expected. Under circumstances with different numbers of jobs, arrival time distribution and hot jobs ratio, it is helpful to prevent the algorithm falling into local optimum, and search a new path to produce good solutions.



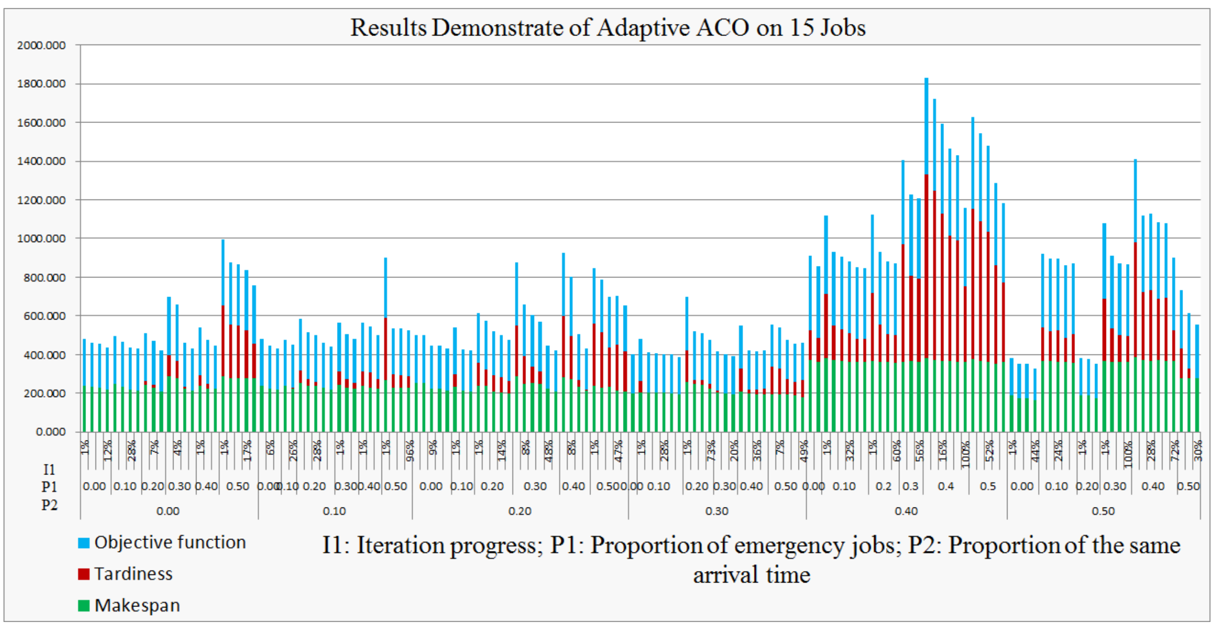
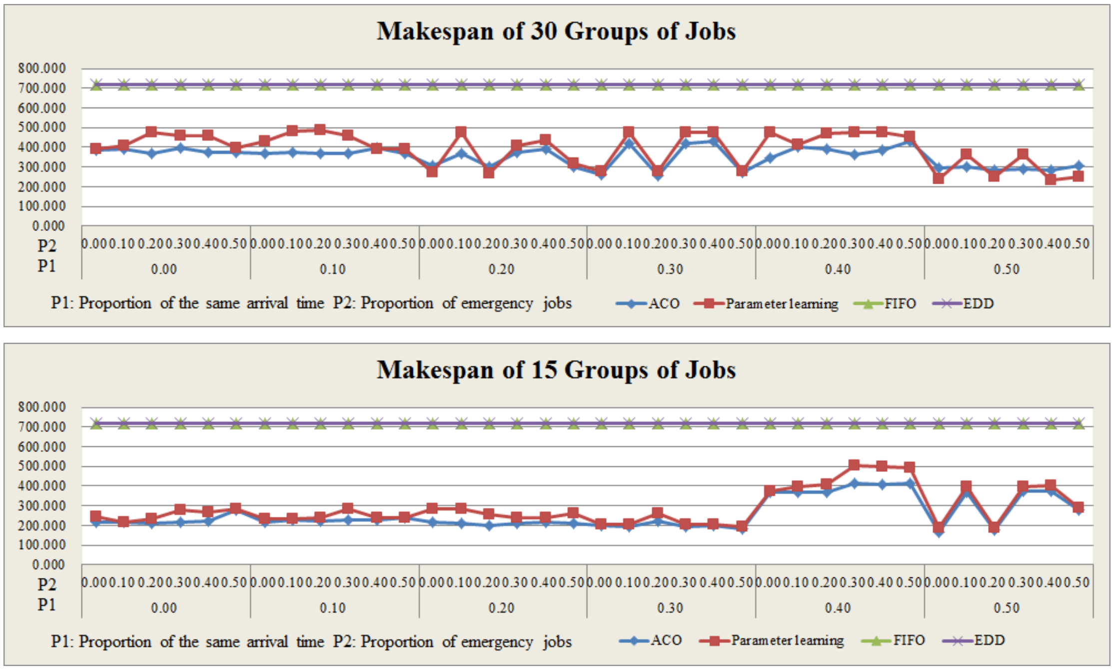
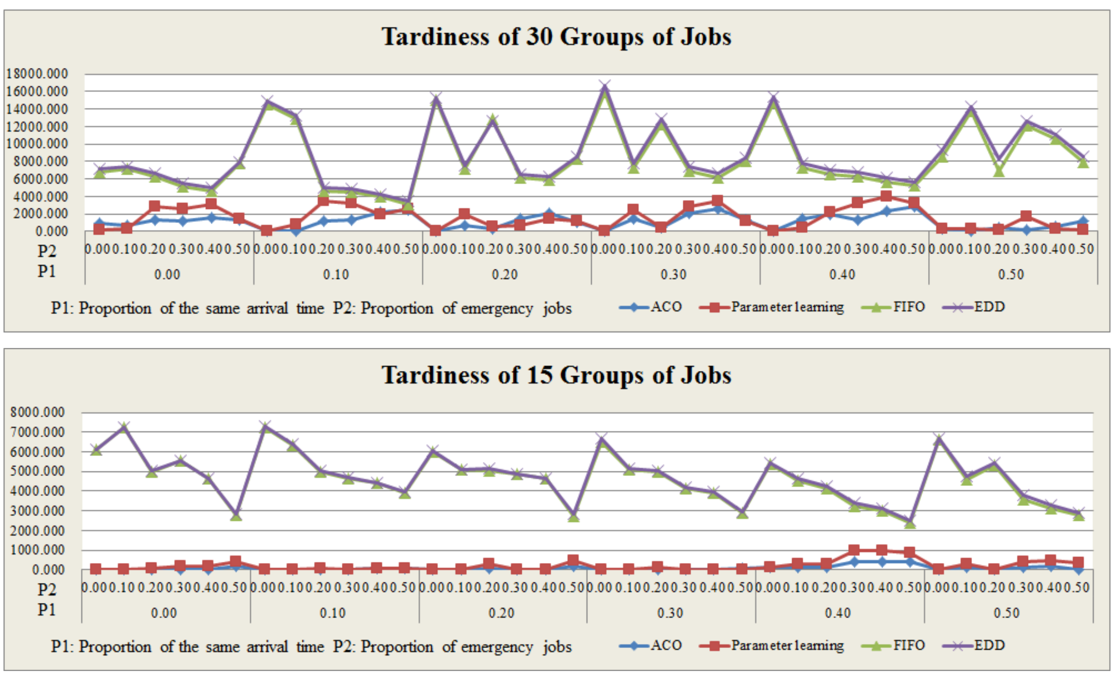


Figure 7: The results demonstration of AAS.

Dispatching results are shown in Figure 8, which shows the comparison among adaptive ant colony algorithm, as well as the combination of parameter learning with traditional scheduling rules (FIFO and EDD with maximum batchsize, they mean scheduling according to arrival time or due date under the batchsize of BPMs. And these method are used in semiconductor industry scheduling). It illustrates that, AAS can achieve better results. In the case of a large number of jobs, LBADM can fit the dispatching results obtained by AAS.



(a) LBADM comparing with scheduling rules in makespan



(b) LBADM comparing with scheduling rules in tardiness

Figure 8: The results comparison.

In addition, the makespan can be improved by 50% and tardiness can be reduced to 30% of the traditional way through LBADM. The tardiness of LBADM is greater than that of AAS. The main reason is that the weight of the tardiness is far less than that of makespan, which can make less effect. Thus it may work better with higher weight of the tardiness. In summary, the dispatch can be fitted in the situation of overall objective function. What's more, the time consumption of LBADM is far less than AAS, which is 10% to 30% of the latter.

Therefore, better decisions to meet production requirements can be obtained in future production scheduling by adjusting the value of weight in the objective function according to the actual situation in order to achieve optimization both in tardiness and makespan, and then considering adding more optimization objectives. Meanwhile, ACO needs to be improved for the further study of self-adaptivity.

5 SUMMARY

Batch scheduling problem in semiconductor production is an important part of semiconductor manufacturing enterprise management, but also the most difficult aspect, whose main task is to develop effective scheduling methods with limited production resources to achieve the target of economic performance optimal, and provide an effective way to improve the comprehensive economic benefits.

This paper takes an intelligent algorithm on pheromone mechanism as basis and proposes an adaptive dispatching rules combining parameter learning, which is different from the traditional scheduling methods. This method can obtain accurate characteristics of the scheduling information, consider the status information of the production line, and complete adaptive adjustment and parameter learning based on the production environment. This way can achieve better results than the traditional static dispatching methods as well as successfully used in semiconductor processing.

In future research work, more complex factors (such as setup time, maintenance tasks and process constraints) need to be added to scheduling strategy proposed in this paper, and thus to learning more effective parameters to optimize the performance of the production line especially the scheduling problem of BPMs to obtain better results.

ACKNOWLEDGMENTS

Research supported in part by National Natural Science Foundation of China (No.61034004, No.61273046), National 863 Project (No.2012AA041005), Shanghai Innovative Project (No. 11DZ1141103) and the Fundamental Research Funds for the Central Universities.

REFERENCES

Mathirajan, M., and A. I. Sivakumar. 2006. "A Literature Review, Classification and Simple Meta-analysis on Scheduling of Batch Processors in Semiconductor." *International Journal of Advanced Manufacturing Technology* 29(9–10): 990–1001.

Li, S. 2012. "Makespan Minimization on Parallel Batch Processing Machines with Release Times and Job Sizes." *Journal of Software* 7(6): 1203–1210.

Wagner, S., and M. Affenzeller. 2006. "The HeuristicLab Optimization Environment." Johannes Kepler University, Linz, Austria. Available via http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.7658&rep=rep1&type=pdf.

Dorigo, M., V. Maniezzo, and A. Colorni. 1996. "Ant System: Optimization by a Colony of Cooperating Agents." *IEEE Transaction on Systems* 26:29–41.

Raghavan, N. R. S., and M. Venkataramana. 2006. "Scheduling Parallel Batch Processors with Incompatible Job Families Using Ant Colony Optimization." In *Proceedings International Conference on Automation Science and Engineering (CASE 2006)*, 507–512. Piscataway, New Jersey: IEEE, Inc.

Li, L., F. Qiao, and Q. Wu. 2009. "ACO-based Scheduling of Parallel Batch Processing Machines to Minimize the Total Weighted Tardiness." *2009 IEEE International Conference on Automation Science and Engineering, CASE 2009*, edited by M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Stützle, and A. F. T. Winfield, 280–285. Piscataway, New Jersey: IEEE Computer Society, Inc.

Li, L. 2012. “Multi-ant Colony-based Sequencing Method for Semiconductor Wafer Fabrication Facilities with Multi-bottleneck.” *International Journal of Modelling, Identification and Control* 15(4): 259–266.

Chang, C. Y., and K. H. Chang. 2012. “An Integrated and Improved Dispatching Approach to Reduce Cycle Time of Wet Etch and Furnace Operations in Semiconductor Fabrication.” In *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2012*, edited by L. Gao, W. Shen, J. P. Barthès, J. Luo, J. Yong, W. F. Li, and W. D. Li, 734–741. Piscataway, New Jersey: IEEE Computer Society, Inc.

Wang, Z., Q. Wu, and F. Qiao. 2007. “A Job Dispatching Strategy Integrating WIP Management and Wafer Start Control.” *IEEE Transactions on Automation Science and Engineering* 4(4): 579–583.

Guo, C., Z. Jiang, and H. Hu. 2010. "A Hybrid Ant Colony Optimization Method for Scheduling Batch Processing Machine in the Semiconductor Manufacturing.” *2010 IEEE International Conference on Industrial Engineering & Engineering Management*, 1698–1701. Piscataway, New Jersey: IEEE, Inc.

Zhang, H., Z. Jiang, and C. Guo. 2009. “An Optimised Dynamic Bottleneck Dispatching Policy for Semiconductor Wafer Fabrication.” *International Journal of Production Research* 47(12): 3333–3343.

Solomon, L., J. W. Fowler, and M. Pfund. 2002. “The Inclusion of Future Arrivals and Downstream Setups into Wafer Fabrication Batch Processing Decisions.” *Journal of Electronics Manufacturing* 11(2): 149–159.

Wang, C. N. 2009. “The Heuristic Dispatching Method of Automatic Material Handling System in 300mm Semiconductor Fabrication.” *International Journal of Innovative Computing, Information and Control* 5(7): 1927–1935.

Shao, H., and H. Chen. 2010. “Minimising Makespan for Single Burn-in Oven Scheduling Problems Using ACO+DP Approach.” *International Journal of Manufacturing Research* 5(3): 271–285.

Lakshmanan, G. T., S. Duan, P. T. Keyser, F. Curbera, and R. Khalaf. 2011. "Predictive Analytics for Semi-structured Case Oriented Business Processes.” *Lecture Notes in Business Information Processing*, 66: 640–651.

Li, L., and H. Wu. 2011. “ACO- ICSA Based Scheduling of a Single Batch Processing Machine in Semiconductor Manufacturing.” *Advances in Information Sciences and Service Sciences* 3(10): 240–246.

Noroozi, A., H. Mokhtari, and I. N. K. Abadi. 2013. “Research on Computational Intelligence Algorithms with Adaptive Learning Approach for Scheduling Problems with Batch Processing Machines.” *Neurocomputing* 101: 190–203.

Fowler, J. 1994. “Modeling and Analysis Semiconductor Manufacturing Laboratory: SEMATECH Dataset.” Arizona State University, Tempe, A.Z. http://masmlab.engineering.asu.edu/ftp.htm.

AUTHOR BIOGRAPHIES

LONG CHEN is a graduate of Control Science and Engineering, Tongji University. His research involves swarm intelligence, scheduling of complex manufacturing systems dispatching and data mining. His email address is eedol2315@gmail.com.

HUI XU is a graduate of Control Science and Engineering, Tongji University. His research focuses on scheduling of complex manufacturing systems dispatching, simulation and modeling. His email address is peperxu@hotmail.com.

LU CHEN is the person in charge of project from Tongji University and Belling company, which engages the scheduling of semiconductor manufacturing. Her email address is chenlu@belling.com.cn.

LI LI is presently a Professor of Control Science and Engineering, Tongji University. Her research interests are in production planning and scheduling, computational intelligence, semiconductor manufacturing, and energy systems. Her email address is lili@tongji.edu.cn.