ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

IDENTICAL PARALLEL BATCH PROCESSING MACHINES BASED ON ACO MULTI-OBJECTIVE SCHEDULING

Rutuja Kulkarni *, Prof. A.M. Qureshi

* Department of Mechanical Engineering , KIT's College of engineering,Kolhapur, India.

ABSTRACT

In this paper, the Batch Processing Machine (BPMs) and its scheduling problem in a semiconductor wafer fabrication facility (fab) has been focused ,where in there are many BPMs, such as diffusion machines, oxidation machines and dry strip machines. The jobs processed on those machines cannot be batched together unless they use they have the same recipe of those BPMs. As a result, the scheduling problem of those BPMs is abstracted as identical PBPM with incompatible job families. In a fab, because most of upstream and downstream machines of the BPMs are non-BPMs, jobs must be batched or split regularly during their fabrication processes. Therefore, a good scheduling solution of those BPMs is important to efficiently utilize their capacity and satisfy the requirements of their downstream machines to balance the fab-wide workload and achieve better fab-wide operational performance.

KEYWORDS: BPMs, ACO algorithm, scheduling processes

INTRODUCTION

The batch processing machines (BPMs) have the ability to process more than one job together (called a batch). So the scheduling problem of the BPMs concerns not only the priorities of the jobs obtaining the processing service of a BPM, but the number of the jobs processed together on them. According to diverse classified criteria (such as the number of the BPMs and the job families), the scheduling problem of the BPMs can be further divided into several styles, e.g., a single BPM scheduling problem (SBPM), identical parallel BPMs scheduling problem (PBPM) , non-identical PBPM, the BPMs scheduling problem with compatible job families and the BPMs scheduling problem with incompatible job families.

Ant Colony Optimization (ACO), inspired by the foraging behavior of real ant colonies,which is a population-based approach. ACO has been successfully applied to several NP-hard combinatorial optimization problems, such as the Traveling Salesman Problem (TSP), Quadratic Assignment Problem (QAP), Job-Shop Scheduling Problem (JSP), Flow-Shop Scheduling Problem (FSP), etc.(Dorigo M. & Stützle T., 2004). However, few researchers have applied ACO to solve the BPMs scheduling problem. In this paper, firstly, we build an identical PBPM model concerned the practical considerations of incompatible jobs, dynamic job arrivals, sequence-dependent setup times and the qual-run requirements of advanced process control (APC). Then, we propose an ACO-based solution to simultaneously minimize the TWT and makespan of the jobs. Finally, the effectiveness of the proposed method is demonstrated by a variety of simulation experiments. The simulation results show that the proposed method produces smaller TWT and makespan than the com- mon Apparent Tardiness Cost-Batched Apparent Tardiness Cost (ATC-BATC) rule, Max Batch Size rule (MBS) and an Ant System algorithm (AS).

MATERIALS AND METHODS

1. Problem Description and Assumptions

1.1 Problem description

With the 3-field notation, the PBPM scheduling problem in this paper can be denoted as

$$M|Aij,Qi,Batch,incompatible| \min(\sum i \sum jwijTij + maxi,j(Fij)) \quad (1)$$

where M is the number of the BPMs in PBPM; Aij is the arrival time of job j of family i ; Qi is the qual-run time of family i ; wij and Tij are the weight, the tardiness and the completion time of job j of family i , respectively.

There are two way to solve a PBPM scheduling problem. One is to distribute the jobs to PBPM first, then batch the jobs and determine the priorities of the batches on each BPM. The other is to batch the jobs first, then distribute the batches to PBPM and sequence the batches on each BPM. Researchers have shown by extensive simula- tions that

the second way achieved better solutions with less computation time. Therefore, we have adopted the second style. There are two main constraints when forming the batches. First, only jobs belonging to the same family can be processed together. Second, the number of the jobs in a batch cannot exceed the capacity of the PBPM (i.e., maximum batch size constraint). Another important consideration is the trade-off between the waiting time for forming a full batch and the waste of the PBPM capacity. Distributing and sequencing the batches are the same as in other problems of scheduling par- allel machines. The issues to consider are the hot lots, workload balance and the utilization of the PBPM. It is also worthwhile to consider the trade-off between the setup times of schedul- ing and the qual-run requirements of APC. In real semiconductor manufacturing environ- ments, APC could achieve the best quality result by frequent changeovers between jobs from different families, possibly avoiding the need for qual-runs. However, frequent changeovers cost setup time and cause capacity loss. Instead of achieving the best quality, APC determines a parameter range which represents acceptable quality for every job family. Based on this range, APC provides a threshold value ni for each job family i . If a machine has been pro- cessing no less than ni jobs (or batches for BPMs) from family j(j = i) ,then before the next time it processes jobs from family i ,a qual-run is required on that machine. In a qual-run, no real job is processed. A blank wafer is processed to obtain the status of the machine so that the operator can properly set the machine parameters to achieve high quality results. Before the result of the qual-run is available, jobs cannot be processed on that machine. Therefore, a trade-off between the time lost for setups and the time lost due to qual-runs is required. However, most related research has not considered the qual-run requirements of APC.

1.2 Problem assumptions:

The assumptions involved in the PBPM scheduling problem include:
(i) The machines in the PBPM are identical; (ii) The PBPM scheduling problem is considered with a schedule horizon (e.g., one shift, one day or several days), within which the scheduling plan of the jobs from multiple families is decided; (iii) The processing time of a batch on one machine is independent of the number of the jobs in the batch; (iv) Once processing begins on a batch, no job can be removed from or added to the machine until it finishes. (v) There are sequence-dependent random setup times for changeovers between jobs from different families, and no setup times between jobs from the same family.

2. ACO-Based Solution

2.1 Build a search space:

Before we use an ACO algorithm to find a solution, the first task is to build a search space for the ACO algorithm. In this paper, the search space is composed of nodes which are combina- tions of the batches and the BPMs in the PBPM. For Ni jobs, there are C1 Ni + C2 Ni +...+ CB Ni (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.

$(\Delta t = dt \times Avg(Pij)$ (2)

where Pij is the processing time of job j of family i ; Avg(Pij) is the average processing time of the jobs; dt is a distribution parameter of Δt . At each batching decision point t (t is set as the earliest ready time of the jobs to be batched), thejobsoffamily i witharrival(ready)timelessthantheupperboundaryofthetimewindow interval $t + \Delta t$ is denoted as $M(j,t, \Delta t) = \{ij|Aij \leq t + \Delta t \}$ .Then, we batch the jobs in $M(j,t, \Delta t)$ subject to the maximum batch size constraint. We repeat the above process until all jobs have been assigned to a batch. The ready time of each batch equals the latest arrival time of the jobs in the batch. Finally, the search space (denoted as S)is built with nodes composed of the batches and the BPMs in the PBPM. Table 1 shows a simple example of building a search space. We assume that there are 2 machines in the PBPM and their batch size is 2 jobs. There are 2 families of jobs whose processing times are set to 10 min and 15 min, respectively. For each family, there are 3 jobs to be scheduled. Here dt is set to 1. Then the time window Δt can be com- puted as

$\Delta t = 1 \times (10 + 10 + 10 + 10 + 15 + 15 + 15)/6 = 12.5$ min .

The batches formed are shown in Table 2. These batches and machines constitute the search space

S={(l11,m1),(l12,m1),((l11,l12),m1),(l13,m1),(l21,m1),(l22,m1),(l23,m1),((l22,l23),m1),(l11,m2),
(l12,m2),((l11,l12),m2),(l13,m2),(l21,m2),(l22,m2),(l23,m2),((l22,l23),m2)} ,

whose size is 16 nodes.

Table 1. An example of building a search space

Job(lij)	l 11	l 12	l 13	l 21	l 22	l 23
Aij(min)	0	5	15	8	13	18

Table 2. The formed batches for the simple example

Batches	l 11	l 12	(l 11,112)	l 13	l 21	l 22	l 23	(l 22,l 23)
Abatch(min)	0	5	5	15	8	13	18	18

2.2 Find a solution with an ACO algorithm:

Step 1: Initialization. There are four main tasks in the initialization stage: to determine the number of the ants in the artificial ant colony, to set the termination conditions for the search, to initialize each artificial ant, and to set the initial pheromones on the arcs.

a) The number of ants in the artificial ant colony
b) The termination conditions
c) Initialization of each artificial ant
d) Initialization of the pheromone on each arc

Step 2: Each artificial ant searches for its solution. Artificial ant k selects its next node l from its task-list according to the so-called pseudorandom proportional rule.

Step 3: Determine whether the termination conditions are satisfied. First, we compute the objective values of the solutions obtained by the artificial ants. Then we select the minimum to compare with that of the last iteration. If the difference between these two consecutive minimum objective values is no more than a small positive value , we stop the search process. The tabu-list with the minimum objective value is taken as the solution.

Step 4: Pheromone updating: The pheromone trail update, both evaporation and new pheromone deposition, only applies to the arcs of the best-so-far solution, not to all the arcs.

DISCUSSION

3.1 The scheduling methods to be compared:

We compared the performances of the proposed ACO algorithm with those of ATC-BATC, MBS, a GA and an Ant System algorithm .As a common BPM scheduling rule, ATC-BATC has good performance on static BPM scheduling problems. To adapt it to dynamic jobs arrival, we made some small modifications to the rule.

3.2 The problem cases for the simulations:

We used the dry strip operations in a real wafer fab to demonstrate the proposed ACO algorithm. There are 3 identical machines for the dry strip operations in this wafer fab. Each machine has a capacity of 3 jobs and can process 4 different recipes. The thresh- old value for the qual-run requirements of each recipe is 3 jobs.

3.3 Determining the distribution parameter:

dt for time window Δt Although Mönch et al.(Mönch et al., 2005) presented the concept of the time window, they did not specify how to determine its value. We ran a number of simulations on random problem cases to determine the best value for the distribution parameter dt of the time window Δt . ATC-BATC was used for the simulations, with its look-ahead parameter k1 set to 0.5 (according to extensive simulations). Values of dt in the range from 0 to 2.0 at intervals of 0.25 were tested.

3.4 Determining the values of the parameters in the ACO algorithm:

Determine the probability parameter q0 Tuning the parameter q0 allows adjusting the degree of exploration and the choice of whether to concentrate the search around the best-so- far solution or to explore other solutions.

3.5 Comparison between ACO, ATC-BATC, MBS, GA and AS :

With the above results, the parameters of the ACO algorithm were set. In the simulations, we considered the impacts of the number and the arrival time distribution of the jobs on the ACO algorithm's performance. The number of jobs was grad- ually increased by multiplying the number of machines and the number of recipes on each machine. the number and the capacity of the machines and the number of the recipes play an important role on the average improvements on the TWT and makespan of the ACO algorithm. Less machines, the bigger capacity and more recipes, the more average improvements on the TWT and makespan are. In addition, the performance of MBS increasingly improved with more recipes and bigger capacity.

CONCLUSION

Batch processing machines play important roles in semiconductor wafer fabrication facili- ties. In this paper, we modeled the batch processing operations in a real wafer fab as an identical PBPM problem considering the practical complications of incompatible job families, dynamic job arrivals, sequence-dependent setup times and qual-run requirements of APC, and proposed an ACO algorithm to solve the problem with smaller TWT and makespan than ATC-BATC, MBS, GA and AS. The main contributions of the paper are to create a method applicable in a

production environment, to propose a better value for the time window Δt from simulations, and to apply the ACO algorithm to obtain the solutions.

ACKNOWLEDGEMENT

I would like to express my sincere thanks to my guide Prof.A.M.Qureshi for his motivation and useful suggestions which truly helped me in improving the quality of this paper. I take this opportunity to express my thanks to my teacher and friends for their encouragement and support.

REFERENCES

1. Mathirajan,M.& Sivakumar,A.I., (2006). A literature review, classification and simple meta- analysis on scheduling of batch processors in semiconductor. International Journal of Advanced Manufacturing Technology, Vol.29, No.9-10, July 2006,990-1001,ISSN 0268- 3768
2. Gupta A.K. & Sivakumar A.I. (2007). Controlling delivery performance in semiconductor manufacturing using look ahead batching. International Journal of Production Research, Vol.45, No.3, Feb 2007, 591-613(23), ISSN 0020-7543
3. Srinivasa Raghavan N.R. & Venkataramana M. (2006). Scheduling parallel batch processors with incompatible job families using ant colony optimization, Proceeding of the 2006 IEEE International Conference on Automation Science and Engineering, Oct 2006,pp.507- 512, ISSN 1545-5955, Shanghai.
4. Patel N-S (2004).Lot allocation and process control in semiconductor manufacturing - a dy-namicgameapproach.ProceedingsofIEEEConferenceonDecisionandControl,Vol.4, pp.4243-4248,,ISSN 0005-1179 ,Atlantis,Paradise Island, Nassau, Bahamas, Dec 2004
5. MonchL,BalasubramanianH,FowlerJ-Wetal.(2005).HeuristicschedulingofjobsonparallelbatchmachineswithincompatiblejobfamiliesandunequalreadytimesComputersandOperationsResearch,Vol.32,No.11,Nov2005,2731-2750,ISSN0305-054
6. Cai Y.W, Kutanoglu E, Hesenbein J et al.(2007). Single-machine scheduling problem with ad- vanced process control constraints,AEC/APC Symopsium XIX, Indian Wells, Calif. USA, pp. 507-512.
7. Chien C-F. & Chen C-H, (2007). A novel timetabling algorithm for a furnace process for semiconductor fabrication with constrained waiting and frequency-based setups. OR Spectrum, Vol.29, No.3, July 2007,391-419,ISSN 0171-6468