LOADING DECISION OF BATCH PROCESSORS FOR MINIMIZING TARDINESS

[1]PYUNG-HOI KOO, [2]PARAMITHA MANSOER

[1,2]pukyong national university, daechang, inc.
E-mail: [1]phkoo@pknu.ac.kr, [2]mithamansor@gmail.com

Abstract- A batch processor is capable of processing several jobs at the same time up to their capacities. When a batch processor completes its job and there are jobs waiting in the queue, a loading decision should be made whether it starts processing a batch of jobs immediately or waits for more jobs to arrive before initiating a batch service. This paper addresses loading decision problem for batch processors with the objective of tardiness minimization. We presents a loading procedure where due-date information of the jobs are considered. Simulation experiments are performed to validate the performance of the proposed model.

Index Terms- batch processor, loading decision, batching, tardiness minimization.

I. INTRODUCTION

The batch processors process a number of jobs simultaneously as a batch. They can be seen in such industries as semiconductor manufacturing, aircraft manufacturing, metal working, hospital sterilization services, theme park rides, and so on. Figure 1 shows a schematic representation of batch processing operations. The jobs arriving at a batch processor is formed as a batch before being served on the batch processor. In general, there is a limitation to the number of batches that can be loaded on a batch processor. The jobs in a batch are processed together and released at the same time. The jobs may be grouped into different different job types. In Figure 1, there are two types of jobs (circle and triangle types) It may be impossible for the jobs of different types to be loaded together in the same batch. Each job type may have different processing time on the batch processor.

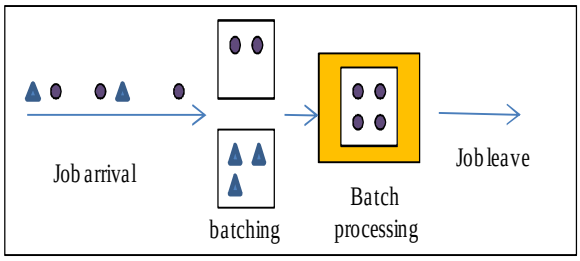


Figure 1. Schematic representation of a batch processor

This paper addresses a loading problem for the batch processors. There are two situations in which loading decisions are made: machine-initiated and job-initiated. When a machine completes the service of a batch and there is at least one job waiting in its waiting queue, a decision is made whether to start a job right away with a partial batch or wait for additional future arrivals to form a larger batch size. Here, selecting the next batch for the batch processor involves decisions on both which job type to be processed next and how many jobs of the selected job type to put in the batch. Selecting the next batch refers to the prioritization of the lots that are put together in a batch. Determining how many jobs in a batch is related to a trade-off between starting the batch immediately and waiting for more lots to arrive. If the number of jobs in the buffer is less than the capacity of the batch processor, starting a batch immediately underutilizes the machine below its capacity. However, waiting for more jobs to arrive before initiating the service increases the queueing time for all the jobs currently in the queue. The job-initiated loading decision is made when a job arrives at a batch processing station and finds at least one batch processor idle. In this case, the decision on whether the job family including the new job is loaded immediately should be made. In this paper, we present a look-ahead loading procedure in which information of the products in queue as well as the upcoming products are considered. Most of the previous research works (which will be discussed in the later section) on loading decisions attempt to minimize the flow time and holding cost of work in process. However, we understand that the due-date related performances are becoming more important than the system-related because of following two reasons. First, customer satisfaction may be the highest competitive strength in today's fiercely competitive business environment. Delivering products to the customers on time is critical for the customer's satisfaction.

The second reason is related to the production flows. In general, a bottleneck machine exists in a production flow line and it should be utilized in full capacity. In order to fully utilize the bottleneck machine, all the other operations including batch machine should feed the products to the bottleneck machine smoothly to prevent the bottleneck machine from being idle due to starvation. In general, production schedule is constructed in detail for the bottleneck machine. Then, the schedule of the bottleneck machine can be

considered a due date for the jobs in the upstream processes. Hence, the loading procedure presented in this paper is constructed with the objective of minimizing tardiness of the jobs. Simulation experiments are conducted to examine the performance of the presented model.

II. PROBLEM DESCRIPTION AND LITERATURE STUDY

Loading strategies of batch processors may be classified into two policies, threshold policy and look-ahead policy, according to the use of knowledge on future arrivals of products. The most basic control strategy of the threshold policy is the minimum batch size (MBS). In this control strategy, processing of a batch is started when the number of products waiting in the queue becomes greater than or equal to the predetermined number. There are many studies dedicated to find the optimal MBS size in many different environments, from single-product single-machine environment to multiple-product multiple-machine environment [1],[10],11],[16]. As this paper focuses on look-ahead control problems, our discussion on previous literature will be mostly on look-ahead policies. Because of its simplicity, many industries use MBS based loading strategy. Recently, since near-future information can be collected through data gathering systems, look-ahead strategy is becoming popular in loading decisions in which near future information such as job arrival and machine status is included in decision making. Glassey and Weng (1991) may be among the first to use look-ahead information for loading decisions. They present a batching heuristic called DBH (dynamic batching heuristic) that takes future lot information into account for the single-machine single-job type scenario. Fowler et al. (1992) develop a new heuristic, called the next arrival control heuristic (NACH), which is a modified version of DBH. The difference is that NACH considers only the next arrival lot. The experimental results indicate that NACH is robust in the sense that it performs well even with errors in the prediction of next arrival time. Weng and Leachman (1993) propose a loading heuristic called MCR (minimum cost rate). The difference between MCR and DBH is the choice of look-ahead horizon. While DBH uses fixed look-ahead horizon, namely process time, MCR uses process time plus prior waiting time. Robinson et al. (1995) present rolling horizon cost rate (RHCR) which is a combination of cost rate calculation in MCR and rolling horizon decisions in NACH. Instead of considering future arrival information, RHCR uses the information from upstream and downstream operation. Tajan (2011) presents a heuristic based on model predictive control (MPCH). At each decision instance, MPC optimally solves a related deterministic problem with truncated horizon. Given the sequence of decisions from the simpler problem, only the first loading decision is implemented and the decision is iterated. The research works above attempt to improve system related performance, lead time or waiting time, to develop their control rules. However, the due-date related performance has received more attention recently. Tardiness is a common measure for the due-date related performance. Some researchers deal with loading decisions of batch processors considering due date and tardiness. Kim et al. (2001) present three batching decision making strategies, MMBS, MDBH and PUCH. MMBS and MDBH are the modification of MBS and DBH where slack times are additionally included in control decision making. PUCH is a modification of MMBS which gives urgency to each job and selects job family with highest urgency. Kim et al. (2010) extend their research by introducing PRALC (Priority Rule-based Algorithm with Look-Ahead Checks) for multiple product type multiple BPMs environment. Gupta and Sivakumar (2006) present a control heuristic called look ahead batching (LAB) for single machine single product type cases. The decision of LAB is made considering arrival time and due date of incoming lots. The objective of LAB is to minimize the average tardiness as well as variation of tardiness. Later, Gupta et al. (2007) improve LAB by considering both tardiness and earliness. Sha et al. (2007) develop a due-date oriented look-ahead control rule, namely LBCR. It is a combination of dispatching rule, CR (Critical Ratio), and look-ahead batching strategy, NACH where CR value is first calculated and adapted for NACH. Cerecki and Banerjee (2010) present NACH-T, a multiple-job look-ahead strategy where minimum tardiness for each job type at each future arrival is calculated and the effects of a job's tardiness on all the other job types are considered in control decision making. Mansoer and Koo (2013) presents a new real-time loading strategy, LBT (Look-ahead batching for Tardiness minimization), for single job type case where the jobs arriving after decision making point are also considered. They show their strategy outperforms the previous models including MMBS and NACH-T.
This paper extends the work in Mansoer and Koo (2013) by considering batch processing station with multiple job types. (The term, LBTm will be used for this new control strategy)

III. A NEW LOADING STRATEGY FOR BATCH PROCESSOR

This section presents a new loading strategy for a batch processor for minimizing tardiness. Some important characteristics of the system under consideration are as follows:

(1) A batch processor can accommodate a limited number of jobs.
(2) It can be estimated when the batch processor finishes its current batch and becomes available.
(3) A machine can process any jobs regardless of their job types. However, the jobs of different job types cannot be formed as a batch.
(4) The jobs arrive to the batch processor dynamically. The arrival time of the jobs (arriving in the near future) can be estimated. Each job has its own due date.
(5) All the jobs within a job type have the same processing time. However, the processing time for each job type may be different from each other.

In LBTm, machine available time as well as job arrival time is considered together for decision making. The decision time for this strategy occurs when a batch processor is available or when a job arrives to the queue. When a loading decision is to be made, jobs of all types are formed as a batch temporarily and priorities of the batches are calculated based on the total tardiness, and then loading time and the jobs to be loaded are selected. A detailed description is given below for each step.
Step 1: At current decision time, t0, total tardiness for each job type is calculated, then the job type with minimum total tardiness is selected as a loading decision alternative. Before calculating the total tardiness, set of jobs of type j to be loaded at decision time t0, Bj0, are formed as a batch based on earliest due date rule. After identifying Bj0, the total tardiness is calculated both for Bj0 and the jobs that are not going to be loaded at this time. The equations for total tardiness calculation in LBTm are adapted from Cerecki and Banerjee (2010). The job type with the smallest tardiness value is then considered as a loading decision alternative at time t0. j* is recorded as the job type to be loaded at time to.

Step 2: At the time that the nth upcoming job arrives to the batch processor, tn, the total tardiness of the job type that arrives at that decision time is calculated and considered as a loading decision alternative. Here, instead of calculating total tardiness value for all possible job types, the calculation of total tardiness should only be done to that specific job type. Based on this consideration, that job type is stated as job type j*. Then, the set of jobs of type j* to be loaded at decision time tn, Bjn, is formed as a batch. The total tardiness is calculated in the same way (only with the change of subscript o to subscript n) stated in the previous step. After the total tardiness is calculated, total tardiness at time tn is recorded. Step 2 is iterated for each future job arrivals. Step 3: In this step, the result obtained from step 1 and step 2 are compared to find the loading time, t_{n*}. The loading time is selected by $n^* = \arg_n \min$ (total tardiness at time t_n). If n* is o, then the batch processor should start right away by loading a batch with job type j*. Otherwise, the batch processor stays idle to wait until one next arrival and then runs the LBTm logic again.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental design is adapted from Gupta (2006) with a modification in due-date assignment. In this experiment, three strategies are compared in a multiple job type single machine environment. The manufacturing systems are modeled with ARENA software package and the heuristic is written in Visual Basic Application (VBA). The simulation runs for 12,000 time units, in which the first 2,000 time units were treated as warm-up period to secure the statistical reliability.
The simulation was run with a single batch processor, which has capacity of six jobs, and five different job types, which has different processing time for each job type. The processing times are set with the average of 60 time units. This average processing time is used to determine the inter arrival time. The inter arrival was adjusted in a way to obtain desirable traffic intensity (TI). The observed traffic intensities for this experiment are 40%, 50%, 60%, 70% and 80%. The traffic intensity is defined as : TI = (average processing time)/(inter arrival time×maximum batch size). The due-date for each job is set by using following equation: d_j= arrival time + (1+UNIF[0,k])*(processing time). The due-date is dependent on a tightness index, k. For the base scenario, we assume that k is 2. The LBTm strategy takes advantage of a certain look-ahead number. In order to determine the look-ahead number for this experiment, pre-experimental runs were conducted.
The pre-experimental runs were conducted in traffic intensity of 60% under different look-ahead number. The result of this pre-experimental runs can be seen in Figure 2. It is seen that the average tardiness shows a decreasing slope when the look-ahead number increases. However, a large look-ahead number requires more information and increases the complexity of decision making. Due to those reasons, the look-ahead number for this experiment is set to be 5, with the consideration that the slope from look-ahead number 5 onwards is not steep.

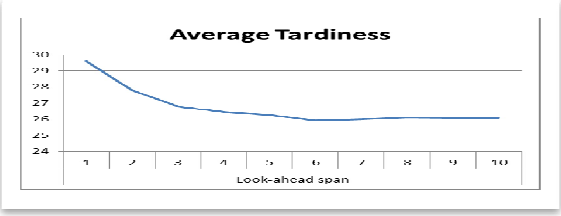


Figure 2. LBTm Average Tardiness on Different Look-ahead Values

The proposed strategy, LBTs, is compared to existing strategies which deal with tardiness minimization on MPSM environment, MMBS and NACH-T. MMBS is modified MBS which considers due-date. The logic for MMBS is when MBS requirement is fulfilled, load the product type with most products in queue. If there are ties between product-types, the one with larger tardiness value is loaded. MMBS will be set as a benchmark for the experiments. Since MMBS performance on different TIs is dependent on MBS value, the MBS value is specified differently for different TIs, MBS is 1 when TI is 40%, MBS is 2 when TI is 50%-70%, and MBS is 3 when TI 80%. While NACH-T is used as it is described in Cerecki and Banerjee (2010). We compare the performance from different loading strategies over different traffic intensities. Due to this purpose, the due-date tightness, k, is set to be 6. The result of the experiment can be seen in Figure 3. It is seen that the proposed method, LBTm provides smaller tardiness value in comparison to the other strategies over different traffic intensities. The better performance of LBTm over NACH-T can be explained from the differences in the heuristics. The essential difference between NACH-T and LBTm is that, at the first step of the heuristic, NACH-T suggests individual decision for each product type without considering the effects on the other product types while MLBT considers the effects on the other product types from the very beginning of the heuristic. Another difference is that, in NACH-T, immediate loading is suggested whenever a full batch is formed while LBTm considers the upcoming products even if a full batch has been formed because upcoming products might have tighter due-date compared to the products in full batch.

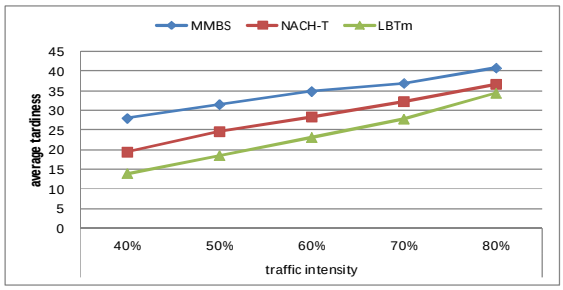


Figure 3. Tardiness Comparison over Traffic Intensity

The performance of NACH-T and LBTm is examined on different due-date tightness over traffic intensity. Figure 4 shows the percentage of performance improvement of LBTm over NACH-T. It is seen that LBTm always performs better than NACH-T in different due date tightness. Another observations is that, even with different due date tightness, the graphs show similar declining curve, from which it can be assumed that the performance of the strategies are more closely related to traffic intensity in comparison to due date tightness.

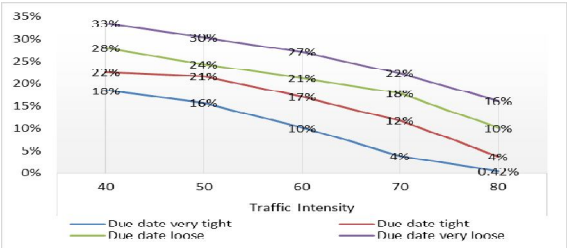


Figure 4. Percentage of Performance Improvement of LBTs over NACH-T

CONCLUSION AND FUTURE RESEARCH

This paper presents a look-ahead loading strategy for batch processors for multiple job type environments. Our loading strategy takes advantage of near-future information from the jobs waiting in queue as well as upcoming jobs. Unlike the existing loading policies, even when the number of jobs in queue is greater than capacity, loading may be delayed for more urgent jobs. The experimental results show that the new loading strategy provides performance of good quality. Our work is now ongoing to observe the effect of uncertainty on the current model. It will be also interesting to see the effect of the loading decision at batch processor on the system-wide performance when the batch processor is one of the processors in manufacturing flow systems.

ACKNOWLEDGEMENT

This research was supported by Basic Science research program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A4A01014897).

REFERENCES

[1] Akcali, E., Uzsoy, R., Hiscock, D.G., Moser, A.L., Teyner, T.J., Alternative loading and dispatching policies for furnace operations in semiconductor manufacturing: a comparison by simulation, Proceedings of the 2000 Winter Simulation Conference, 1428-1435, 2000.

[2] Cerekci A, Banerjee A., Dynamic control of the batch processor in a serial-batch processor system with mean tardiness performance, International Journal of Production Research, vol. 48(5), pp. 1339-1359, 2010.

[3] Fowler JW, Phillips DT, Hogg GL, Real-Time Control of Multiproduct Bulk-Service Semiconductor Manufacturing Processes, IEEE Transactions on Semiconductor Manufacturing, vol. 5(2), pp. 158-163, 1992

[4] Glassey C, Weng W Dynamic batching heuristic for simultaneous processing, IEEE Transactions on Semiconductor Manufacturing, vol. 4(2), pp. 77-82, 1991.

[5] Gupta A, Sivakumar A Optimization of due-date objectives in scheduling semiconductor batch manufacturing, International Journal of Machine Tools and Manufacture, 46, pp. 1671-1679, 2006.

[6] Gupta A, Sivakumar A. Controlling delivery performance in semiconductor manufacturing using look ahead batching, International Journal of Production Research, vol. 45(3), pp. 591-613, 2007.

[7] Kim, Y.D., Kim, J.G., Choi, B. and Kim, H.U Production scheduling in a semiconductor wafer fabrication facilitiy

producing multiple product types with distinct due dates, IEEE Transaction on Robotics and Automation, vol. 17(5), pp. 589-598, 2001.

[8] Leachman, R.C., Kang, J. and Lin, V. SLIM: Short cycle time and low inventory in manufacturing at Samsung Electrnics, Interfaces, vol. 32(1), pp. 61-77, 2002

[9] Mansoer, P. and Koo, P.H., A control strategy of batch processing machines in semiconductor manufacturing, Proceedings of 17th International Conference on Industrial Engineering, 830-837, Busan, Korea, 2013.

[10] Neuts, M.F., Nadarajan, R., A Multiserver queue with thresholds for the acceptance of customers into service, Operations Research, 30(5), 948-960, 1982

[11] Phojanamongkolkij, N., J.W. Fowler, J.K. Cochran, Determining operating criterion of batch processing operations for wafer fabrication, Journal of Manufacturing Systems, 21(5), 363-379, 2002.

[12] Robinson, J.K. Fowler, J.W., Bard, J.F. The use of upstream and downstream information in scheduling semiconductor batch operations, International Journal of Production Research, vol. 33(7), pp. 1849-1869, 1995.

[13] Sha, D., Hsu, S., and Lai, X. (2007). Design of due-date oriented look-ahead batching rule in wafer fabrication, International Journal of Advanced Manufacturing Technology, vol. 35, pp. 596-609, 2007

[14] Tajan, J., Sivakumar, A., and Gershwin, S., Control of single batch processor with incompatible job families and future job arrivals, IEEE Transactions on Semiconductor Manufacturing, vol. 24(2), pp. 208-222, 2011.

[15] Weng, W. and Leachman, R.C. An improved methodology for real-time production decisions at batch-process work stations, IEEE Transactions on Semiconductor Manufacturing, vol. 6(3), pp. 219-225, 1993.

[16] Xia, C., Michalidis, G., Bambos, N., and Glynn, P., Optimal control of parallel queues with batch service, Cambridge University Press, 289-307, 2002.

★★★