International Symposium on Scheduling 2006
July 18 - 20, 2006, Tokyo, Japan
3A2

Online Rescheduling in Semiconductor Manufacturing

Mingang CHENG, Masao SUGI, and Jun OTA
Department of Precision Engineering, School of Engineering, the University of Tokyo, Japan
R822, Faculty of Engineering Bldg.14, 7-3-1 Hongo, Bunkyo-Ku, Tokyo, 113-8654, Japan
[cheng, sugi, ota]@prince.pe.u-tokyo.ac.jp

Masashi YAMAMOTO, Hiroki ITO, and Kazuyoshi INOUE
NS Solutions Corporation, Tokyo, Japan
3-3-1, Minatomirai, Nishi-Ku, Yokohama, Kanagawa, 220-8401, Japan
[myamamoto, hitoh, kinoue]@sysrdc.ns-sol.co.jp

Abstract

Semiconductor manufacturing is mainly characterized by diversity of products, different process types, and random failures. It is extremely difficult to solve manufacturing rescheduling problems in real time due to high frequency of disturbance occurrence (about 1 time/1 minute). This paper proposes a new approach for online manufacturing rescheduling. Different from traditional methods to have a scheduling process again, we revise the existing schedule to keep high schedule stability based on message passing; and during the execution of manufacturing, we improve its performance of the revised schedule on due date with an introduction of a quick local search by permutations of processes on semi-critical paths. Through the actual problems with about 200,000 processes, this method can effectively accommodate disturbances in less than 1 second, and obtain a better executable schedule in less than 1 minute. It is also evaluated to be more effective (higher schedule stability and less violation of due date) by a comparison with conventional dispatching-rule based methods, some of which have been actually applied in many facilities.

Keywords: Semiconductor manufacturing, online rescheduling, message passing, permutation, semi-critical path

1. INTRODUCTION

1.1 Background

Semiconductor Manufacturing is an industry that provides essential solutions, including microprocessors, integrated circuits, chipsets, memory, logic devices, and related products, to the electronic components and information technology industries as well as those they serve. It has several conspicuous characteristics as follows:

- Up to thousands of lots.
- Hundreds of process steps in a process flow.
- Hundreds of resources including batch tools (a tool that can manufacture several processes at the same time).

Due to its complex characteristics in manufacturing facilities, scheduling is necessary for semiconductor manufacturers. In reality, it is not uncommon that several thousand disturbances (for example, machine breakdown, variation of processing time, etc.) will occur in manufacturing facilities each day. Furthermore, even if a slight duration of disturbance happens, it will also invalidate the execution of the original schedule. In the dynamic, stochastic manufacturing environments, managers and supervisors should not only generate high-quality schedules but also react quickly to disturbances and revise schedules in a lower computation cost.

In recent years, effective rescheduling method is more practically mandatory in order to minimize the effect of disturbance on the performance of the system while maintaining the flexibility of schedule. There are several serious issues for semiconductor manufacturing rescheduling which should be resolved:

- Real-time response to disturbances.
- High-quality schedule.
  - *i).* Compliance with due date.
  - *ii).* High *schedule stability*.

Schedule stability is a performance measure on difference between the new schedule and the original one. In practice, vast changes of processing orders in recent production schedule will have an extra influence on recent production manufacturing, for example, lack of materials, difficulty in procurement of workers, etc.

Now, several scheduling methods have been proposed for actual manufacturing facilities to continuously generate and update production schedules. However, there still exists a large gap between the theory and practice of scheduling. To our knowledge, none of existing rescheduling methods has had a consideration of schedule stability yet.

1.2 Literature Review

As Gupta and Sivakumar (2004) said, semiconductor manufacturing scheduling problem is considered as one of the most complicated job shop scheduling problem. By using job shop scheduling methods, a lot of researchers have emphasized researches on manufacturing rescheduling since last century.

- **Mathematical programming techniques**

In recent years, with the development of high computational power of modern computers, mathematical programming techniques, which have been proved effective for job shop scheduling problem, are possible to be applied to semiconductor manufacturing by decomposing whole problem into a number of sub-problems, such as branch and bound method (Sung and Choung, 2000), and decomposition methods (Demirkol, et al., 1995), etc. These methods can provide optimal solutions. However, they are limited by the necessary enumeration and hence computer memory and CPU time.

- **Meta-heuristic methods**

Meta-heuristic methods obtain solutions by continuing to add small changes (termed as permutation) and evaluate schedules until there is no improvement, such as genetic algorithm (Cavalieri, et al., 1999) and simulated annealing (Yim and Lee, 1999), etc. These methods can provide better solutions in an insignificant computation time. They are gaining importance in solving the rescheduling problems. However, they are unsuitable for quick manufacturing rescheduling by the necessary large computation cost.

- **Heuristic rules (dispatching rules)**

Numerous methods based on dispatching rules (a policy which determines the processing priorities of processes on resources) for manufacturing rescheduling have been proposed (Manfred and Alexander, (1999); Dabbas and Fowler, 2003). Due to their low computation cost, ease of implementation, these methods have been recognized to be effective and actually applied in manufacturing facilities. However, they are weak on keeping high schedule stability due to the neglect of consideration of original schedule, and cannot guarantee that the system will operate at a good performance level.

In this paper, we will focus our research on the rescheduling process of recent production schedule to propose an effective online rescheduling method. Based on message passing, we accommodate disturbances in real time by revising the original schedule quickly into a new executable one with high schedule stability. During the execution of manufacturing recovering from disturbances, we introduce a local search algorithm to improve the performance on due date by permutations of two processes on semi-critical paths.

The rest of this paper is organized as follows. Section 2 has a description of the proposed online rescheduling system. Section 3 introduces the rescheduling algorithm based on message passing to accommodate disturbances. Section

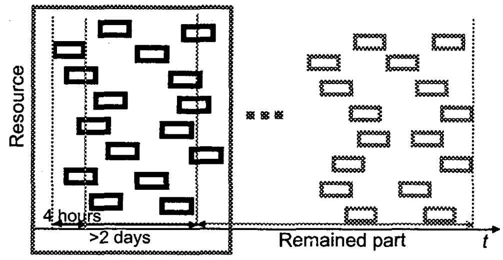


Fig.1 Production schedule in manufacturing facilities

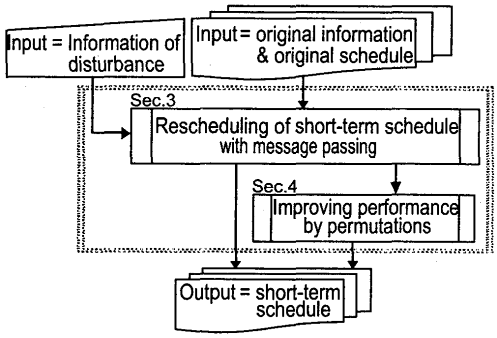


Fig.2 Online rescheduling system overview

4 introduces the local search by permutations on semi-critical paths. And Section 5 has an explanation on simulation and results. Finally, Section 6 concludes this research and has a description on future works.

2. RESCHEDULING METHODOLOGY

2.1 Effective Rescheduler in Manufacturing Facilities

For some reasons on the limitations of manufacturing environment and technology in semiconductor manufacturing facilities, an effective rescheduler is desirable to obtain a new feasible schedule with better performance in a shorter computation time (Fig.1). Effective reschedulers in manufacturing facilities are required the following properties:

- **Better performance**
  - *i).* Unchanging of original processing orders in recent manufacturing within a technological time (4 hours).
  - *ii).* Less difference between recent parts of the new schedule and that of the original one within a minimal procurement time (no less than 2 days).
  - *iii).* Good performance as that of the original schedule.

- **Lower computation time**
  - *i).* Quick response to disturbances (in less than 1 second).
  - *ii).* Rescheduling in a low computation cost (in less than 1 minute).

In this paper, to respond to disturbances in real time we propose a new rescheduling method by decomposing the whole schedule into two sub ones according to the procurement time and emphasizing on the recent production schedule (termed as short-term schedule).

2.2 Online Rescheduling System

In general, the greater the computation time takes in local search algorithm, the better the performance of the schedule is, and the more out-of-date the schedule will be when complete. It is this dilemma that we address our online rescheduling system by a dual rescheduling process (Fig.2). The procedure of dual rescheduling is outlined as follows:

- Revise the most recently generated schedule to accommodate disturbances in real time based on message passing.
- Improve the performance of the revised schedule during the execution of manufacturing after accommodation according to permutations of processes on semi-critical paths.

By the passing of message of disturbance among affected processes along the original processing orders, the original schedule can be revised to accommodate the disturbance in real time. In this way, a new feasible initial schedule with high schedule stability can be obtained. Moreover, by a quick local search from initial schedule, the performance on due date will be possibly improved in a short time.

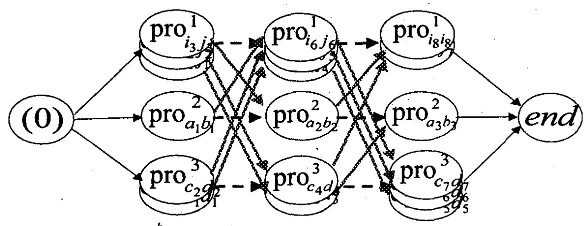
3. DISTURBANCE ACCOMMODATION

3.1 Message Passing on Rescheduling

Similar to job shop scheduling problems (*JSSPs*), semiconductor manufacturing rescheduling problems can be described by a disjunctive graph $G(V, C \cup D)$, considering batched processes (processes that are processed at the same time on a resource) as a combined one (Fig.3). Here, V is a set of vertexes representing processes, C is a set of conjunctive arcs representing the technological procedures of processes on a lot, and D is a set of disjunctive arcs representing the processing contexts of conjunctive processes (processes that are manufactured at different time on a resource). Message passing, which is considered as transmission of messages between two vertexes, works well on the distributed systems like scheduling problems.

Scheduling problems are formulated as constraints satisfaction. Hino, et al. (2001) proposes a recursive propagation method based on message passing for job shop scheduling. By the recursive one to one communication among processes in job shops, job shop scheduling problem can be solved in a decentralized way. However, it is difficult to solve semiconductor manufacturing rescheduling problems because of large-scale processes in manufacturing facilities.

In general, there are multiple paths from one process to another one as the disjunctive graph shows in Fig.3. During transmission of information, multiplex message updating of a process will possibly occur, which will result in the loss of computation time. To have a quick response to disturbances, we will have an introduction of a new rescheduling algorithm next by an improvement on normal scheduling algorithm based on message passing.



pro_{ij}^{k} : j-th process of i-th lot using k-th resource

Fig.3 Disjunctive graph of rescheduling problem

3.2 Sorting Rescheduling Algorithm

Here, we improve message passing with a sorting process. According to original starting times of affected processes, disturbance information is transmitted one after another in order to avoid redundant message updates. In this way, semiconductor manufacturing rescheduling problems can be solved quickly in a centralized manner.

The algorithm is shown in Fig.4. In this algorithm, we define a queue with the construction shown in Fig.4 (b) to register the addresses of affected processes and disturbance information.

As shown in Fig.3, there are two kinds of neighboring relations among processes: the one is on the same lot, and the other is on the same resource. Any of processes can notice the information to all of its successors based on message passing. Once a disturbance occurs, affected processes can revise their original state (start time) one by one with the recursive transmission of disturbance information in an original processing order. In this way, the existing schedule can be modified to accommodate the disturbance.

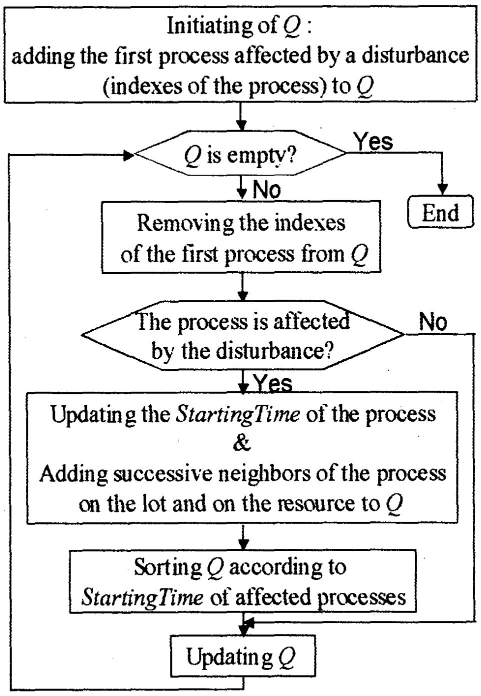
3.3 Evaluation of Sorting and Non-Sorting Algorithm

To compare the effectiveness of sorting and normal (non-sorting) algorithms on rescheduling, we compare their computation time and volume of message necessary for accommodating disturbances. In this case, an actual problem in manufacturing facilities with about 200,000 processes is cited. A disturbance is considered as the delay of starting time on Process (90, 0) at Resource 190 with different durations from t = 0. The interval of recent schedule (termed as scheduling span time) is set to be 25 days, which is much larger than minimal procurement time (2 days).

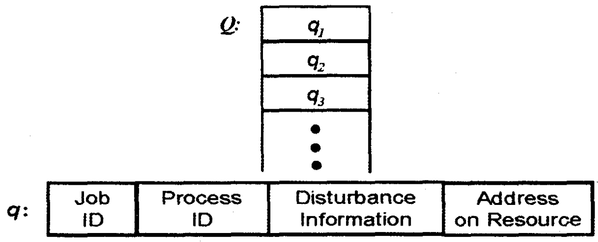
Two algorithms are coded in the Java ™ 2 SDK, Standard Edition Version 1.4.2 under Microsoft Windows XP operating system run by a PC with Intel Pentium M 2.0GHz and 1.5GB RAM.

Once the disturbance occurs in facilities, the original schedule is rectified to accommodate the disturbance. In this paper, for simplicity, the schedule is generated by the method based on a single dispatching rule of MWKR (most work remaining, which have been recognized to be more effective for scheduling on total processing time of manufacturing, is a policy that the process with the most work remained to be processed will be first processed.).

The result is shown in Fig.5. The horizontal axis is duration of disturbance, and vertical axes are computation time



(a) Flow chart of sorting rescheduling algorithm



(b) Construction of the Queue

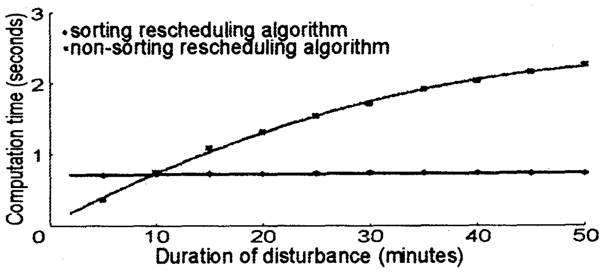
Fig.4 Sorting rescheduling algorithm

and volume of message transmitted, respectively.

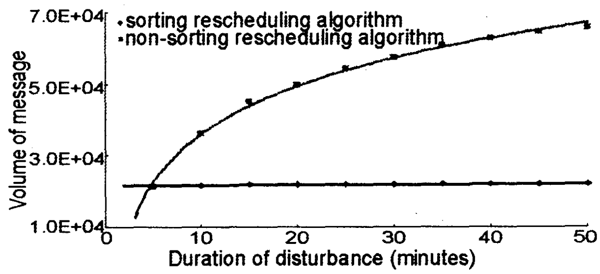
As Fig.5 (a) shows, the non-sorting rescheduling algorithm accommodates the disturbance in an increasing computation time; while the sorting rescheduling algorithm accommodates the disturbance in an approximately unchanging computation time. Moreover, as Fig.5 (b) shows, the volume of message transmitted in the non-sorting rescheduling algorithm increases with the increase of duration of disturbance; on the contrary, the volume of message transmitted in the sorting rescheduling algorithm is almost the same.

By the comparison of Fig.5 (a) and (b), we can know, sorting rescheduling algorithm accommodates the disturbance in a shorter time because the redundancy of message updating is avoided. It is more effective than non-sorting rescheduling algorithm.

As Fig.5 (a) shows, sorting rescheduling algorithm accommodates disturbance in a cost-effective manner (less than 1 second). However, there is not any consideration of performances in this algorithm. In actual manufacturing facilities, delivering orders on time is also important for any of manufacturers. In this respect, we will introduce local



(a) Computation time



(b) Volume of passing message

Fig.5 Comparison of non-sorting and sorting algorithms

search after rescheduling to improve the performance on due date in next section.

4. LOCAL SEARCH IN NEIGHBORHOOD

Due to the high-mix manufacturing of up to thousands of products, we can obtain different feasible schedules by changing the processing orders of products on resources. It is possible to improve the performance on due date quickly by permutating processes on semi-critical paths.

4.1 Semi-Critical Path

Nowicki and Smutnicki (1996) proposed a quick local search algorithm on makespan (defined as an interval between the beginning time and the finishing time of the schedule) by permutations on critical path (defined as the longest path from source (0) to sink (end) in Fig.3), which is only related to the last lot in the schedule. In order to consider the due dates of all lots, we introduce semi-critical path here by an expansion of critical path and define it as the longest path from the start of schedule to the end of a lot. It is composed of a sequence of critical processes. A sequence of consecutive critical processes on the same resource is called a critical block.

As Nowicki and Smutnicki (1996) said, a set of interchanges near the borderline of blocks can be introduced to improve the performance. As Fig.6 shows, by swapping the first two or the last two pairs of processes in a block of a semi-critical path, the performance on due date are possible to be improved.

However, not all permutations near the borderline of critical blocks can improve the performance on due date. Next, we will have a description on removing useless permutations that cannot improve the performance to have a quicker local search.

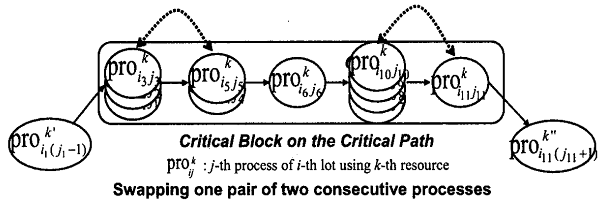
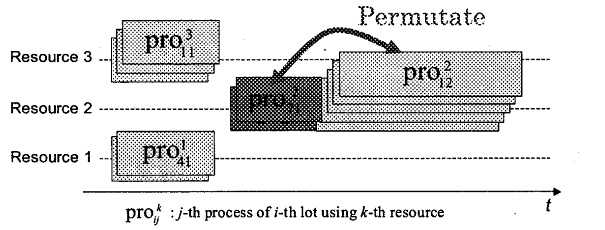
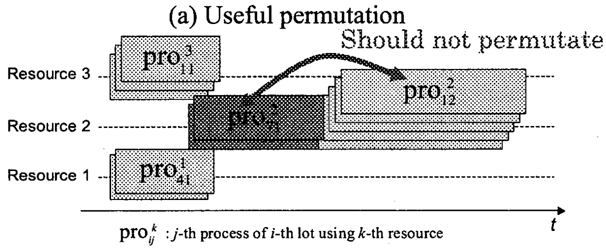


Fig.6 Permutation of processes on critical blocks



(a) Useful permutation



(b) Useless permutation

Fig.7 Example of permutations on a resource

4.2 Permutation Conditions

On semi-critical paths, a set of permutations (termed as useful permutation) might improve the performance of the schedule just under the conditions that:

A pair of processes and its previous pair of processes on the same lots have a difference of time.

Its previous pair of processes on the same resource is started after its previous pair of processes on the same lots finishes.

An example of two kinds of permutations on a resource is shown in Fig.7. By permutating the batched processes (pro_{71}^{2} , pro_{81}^{2}) and the batched processes (pro_{12}^{2} , pro_{22}^{2} , pro_{32}^{2} , pro_{42}^{2} , pro_{52}^{2}) on Resource 2, useful permutation can hasten the finishing time on Resource 2; while, useless permutation delays the finishing time.

By the introduction of useful permutations on semi-critical paths, it is possible to improve the performance in a shorter time.

5. SIMULATION RESULTS

5.1 Performance Measures

In this era, optimization of performance is still a hot topic in manufacturing rescheduling. The most common performance measures for rescheduling methods are schedule stability, tardiness and computation time.

- **Degree of unsimilarity (Schedule stability)**

Schedule stability is a non-regular performance measure, and defined as the measure on variation of processing orders of the new schedule from original one:

$$\eta_{\mathrm{Sequence}} = (1 - NA_{\mathrm{Same}} / NA_{\mathrm{Total}}) \times 100\% . \qquad (1)$$

Here, η_{Sequence} is the degree of unsimilarity, i.e., the proportion of the processing orders different with that in the original schedule; NA_{Same} is the number of arcs on resources in both the new and original schedules; and NA_{Total} is the number of total arcs on resources in the original schedule.

- **Violation of due date (Tardiness)**

Tardiness, defined as the total violation of due date, is an important performance measure for evaluating on-time delivering in manufacturing facilities:

$$T = \sum_{i=0}^{Lot.length} T_i = \sum_{i=0}^{Lot.length} \max\{c_{im_i}^{k} - \mathrm{dd}_i, 0\} . \qquad (2)$$

Here, T_i is the tardiness of i-th lot; $c_{im_i}^{k}$ is finishing time of i-th lot; and dd_i is due-date of i-th lot.

5.2 Experiment and Result

5.2.1 Scheduling span time

In general, the longer the rescheduler look-ahead time, the greater the computation time taken. To obtain a better scheduling span time for rescheduling, 4 actual manufacturing rescheduling problems with about 200,000 processes is cited. The original schedules are generated by the method based on a single dispatching rule of MWKR. For the sake of convenience, we assume that due date of each lot is equal to their finishing time in the original schedule. In this case, a disturbance of resource breakdown with the duration of 10 minutes is considered to occur at a resource from t = 0 with different scheduling span times.

The result is shown in Fig.8. The horizontal axis is scheduling span time for rescheduling; and the vertical axis is improvement on tardiness based on local search algorithm.

In this case, best scheduling span time for best improvement on tardiness can be found as 4 days. As for the other disturbances, we can also obtain similar results.

In general, with the decrease of scheduling span time in the earlier part before 4 days, the possibility will decrease accordingly due to the decrease of the number of critical blocks on semi-critical paths. On the contrary, in the later part after 4 days, with the increase of scheduling span time, the possibility will decrease as the same because of the increase of computation time on the calculation of semi-critical path and obtainment of feasible solutions after permutations.

5.2.2 Comparison with conventional methods

As we know, different algorithms have different per-

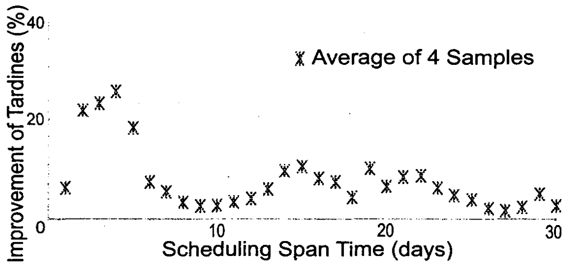


Fig.8 Rescheduling on different scheduling span time

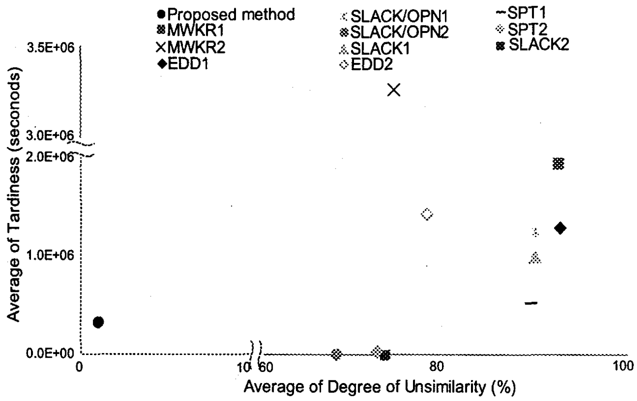


Fig.9 Classification map on rescheduling performance

formances on rescheduling. To evaluate the effectiveness of this method, an actual rescheduling problem in last simulation is cited. In this case, a disturbance of resource breakdown is considered to occur at a resource from t = 0 with different durations, and the scheduling span time is set to be 4 days.

The result of comparison with several dispatching-rule based methods (SPT (shortest processing time): a policy that a process on a resource with the shortest processing time will be processed first; EDD (earliest due date): a policy that the process of a lot nearest to the due date will be processed first; SLACK: a policy that a process with the least slack time to finish a lot will be processed first; SLACK/OPN: a policy that a process with the least average slack time will be processed first) is shown in Fig.9. Here, the last digital of each algorithm represents the difference of the rescheduling process as follows:

- 1 means that processes are distributed to its processing resources on average without any consideration of the state of resources while entering into facilities.
- 2 means that a resource selects a set of processes in manufacturing facilities when it is free from processing.

In Fig.9, the vertical axis is the performance on tardiness. Algorithms with better performance on tardiness are positioned in lower area; The horizontal axis is the performance on degree of unsimilarity. Algorithms with better performance on degree of unsimilarity are positioned in left area.

From the Fig.9, we can know, the proposed method obtains better performance (relatively lower tardiness and lowest degree of unsimilarity). It is more effective for semiconductor manufacturing rescheduling, especially on keeping high schedule stability.

6. CONCLUSIONS

In this paper, we propose an online rescheduling method for semiconductor manufacturing. Based on message passing, the original schedule can be revised to accommodate disturbances with high schedule stability in less than 1 second. By permutations of processes on semi-critical paths, the performance on due date can be improved in less than 1 minute. Compared with several dispatching-rule based methods, the proposed method is more effective for rescheduling on keeping schedule stability.

In the future, besides the accommodation of disturbance of resource breakdown, we must do more simulation on the accommodation of other disturbances.

References

Cavalieri S, Crisafulli F, and Mirabella O. (1999). *A genetic algorithm for job-shop scheduling in a semiconductor manufacturing system*. IEEE Trans Eng Manage, Vol.41, No.2, pp.957-961.

Dabbas RM and Fowler JW. (2003), *A new scheduling approach using combined dispatching criteria in wafer fabs*. IEEE Transactions on Semiconductors Manufacturing, Vol.16, No.3, pp.501-510.

Demirkol E, Uzsoy R, and Ovacik IM. (1995). *Decomposition algorithms for scheduling semiconductor testing facilities*. IEEE/CPMT International Electronics Manufacturing Technology Symposium, pp.199-204.

Gupta AK and Sivakumar AI. (2004). *Job shop scheduling techniques in semiconductor manufacturing*. International Journal Advanced Manufacturing Technology, Vol.27, Num.11-12, pp.1163-1169.

Hino R, Izuhara K and Moriwaki T. (2001). *Message exchange method for decentralized scheduling*. Proceedings of the 4th I4EEE International Symposium on Assembly and Task Planning Soft Research Park, pp.244-248.

Manfred M and Alexander K. S. (1999). *Comparison of dispatching rules for semiconductor manufacturing using large facility models*, Proceedings of the 1999 Winter Simulation Conference, pp.709-713.

Nowicki E and Smutnicki C. (1996). *A fast taboo search algorithm for the job shop problem*. Management Science, Vol.42, No.6, pp.797-813.

Sung CS, Choung YI. (2000). *Minimizing makespan on a single burn-in oven in semiconductor manufacturing*. Eur J Oper Res, Vol.120, pp.559-574.

Yim SJ, Lee DY. (1999). *Scheduling cluster tools in wafer fabrication using candidate list and simulated annealing*. Journal of Intelligent Manufacturing, Vol.10, No.6, pp.531-540.