Journal of Manufacturing Systems xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical paper

Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization

Xiaokun Chang [a], Ming Dong [a,*], Dong Yang [b]

[a] Department of Operations Management, Antai College of Economics & Management, Shanghai Jiao Tong University, 535 Fahua Zhen Road, Shanghai 200052, PR China
[b] Department of Information Management, College of Management, Shanghai Donghua University, 1882 Yan'an Road, Shanghai 200051, PR China

ARTICLE INFO

Article history:
Received 11 December 2011
Received in revised form 16 June 2013
Accepted 9 July 2013
Available online xxx

Keywords:
Integrated delivery
Dispatching rule
GA-based simulation optimization methodology
Response surface methodology

ABSTRACT

In a wafer fabrication Fab, the "integrated delivery", which integrates the automated material handling system (AMHS) with processing tools to automate the material flow, is difficult to implement due to the system complexity and uncertainty. The previous dispatching studies in semiconductor manufacturing have mainly focused on the tool dispatching. Few studies have been done for analyzing combinatorial dispatching rules including lot dispatching, batch dispatching and automated guided vehicle (AGV) dispatching. To handle this problem, a GA (genetic algorithm) based simulation optimization methodology, which consists of the on-line scheduler and the off-line scheduler, is presented in this paper. The on-line scheduler is used to monitor and implement optimal combinatorial dispatching rules to the semiconductor wafer fabrication system. The off-line scheduler is employed to search for optimal combinatorial dispatching rules. In this study, the response surface methodology is adopted to optimize the GA parameters. Finally, an experimental bay of wafer fabrication Fab is constructed and numerical experiments show that the proposed approach can significantly improve the performance of the "integrated delivery system" compared with the traditional single dispatching rule approach.

1. Introduction and literature review

Semiconductor manufacturing is one of the most sophisticated manufacturing processes with characteristics of large processing steps, re-entrant material flows, and batching processing. In the past decades, the dispatching decision problems of processing tools and material handling in wafer fabrication are usually investigated separately due to the modeling complexity. However, an effective dispatching mechanism which properly integrates tools dispatching with AMHS dispatching is becoming a more widely used strategy to achieve high performances in semiconductor wafer fabrication systems.

Most approaches to the semiconductor manufacturing scheduling problem can be classified into four categories: heuristic rules, mathematical programming techniques (such as branch and bound, Lagrangian relaxation and queuing network model), neighborhood search methods (such as Tabu search, genetic algorithm and filtered beam search) and artificial intelligence techniques (such as artificial neural networks and expert/knowledge-based systems) [1].

In recent years, many studies have focused on the dispatching rules for scheduling semiconductor wafer fabrication systems. Heuristic rules and mathematical programming techniques are widely used for this problem. Yang and Chang formulate a multi-objective model for IC (Integrated Circuit) sort and test [2]. In their study, Lagrangian relaxation is used to look for an approximate Pareto boundary and a new algorithm is designed to solve the dual problem. Li et al. propose a dispatching rule to improve on-time delivery without decreasing throughput and increasing cycle time [3]. Wu et al. develop a line balance-starvation avoidance (LB-SA) algorithm based on a proposed simplification model of the processing route for a Fab with machine-dedication features [4]. Monch and Driebel also propose a heuristic rule, named modified shifting bottleneck heuristic, for wafer fabrication Fab [5]. Among these approaches, dispatching rules, by far, are the most commonly used tools for shop floor control. The reason behind this is that they are simple to implement, quick in reacting to the changes encountered on the shop floor, easy to understand and require a low computational load [6].

Pierce and Yurtsever present a value-based concept. They mainly concentrate on generating profits [7]. Based on that, Hsieh and Hou develop a production-flow-value-based job dispatching rule (PFV) by the theory of constraints (TOC) for wafer fabrication [8]. The TOC and profitability costs estimation of a WIP-wafer lot are derived to prioritized jobs based on their profitability. Although

* Corresponding author.
E-mail address: mdong@sjtu.edu.cn (M. Dong).

0278-6125/$ – see front matter © 2013 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jmsy.2013.07.001

dispatching rules are usually sub-optimal and myopic, over the years, researchers have successfully introduced advanced rules that are capable of improving multiple performance measures simultaneously [6]. Dabbas and Fowler develop a composite dispatching rule for each station, which combines multiple dispatching rules, including local and global rules, into a single rule [9]. It is designed to simultaneously maximize the on-time delivery and minimize the variance of lateness and cycle time by using a linear combination with relative weights. Bahaji and Kuhl also propose a composite dispatching rule and prove it to be robust for the average and variance of flow time, as well as due-date adherence measures [10]. Another new weighted composite rule suggested by Yang et al. also take into consideration the general advantage of the earliest due date (EDD) and shortest processing time (SPT) rules, with the weights determined from the historical data [11]. However, those rules above are designed for lots in single processing stations and may not perform the best for different scenarios.

Overhead monorail systems are widely used in wafer manufacturing systems due to their efficiency in lean designs, gentle handling of wafers and efficient use of overhead space [12]. It has been demonstrated that they can significantly improve the inventory storage and material handling system reliability [13]. Christopher et al. indicate that both processing tool and vehicle dispatching rules and their interaction have a significant impact on Fab performance if the AMHS is not extremely over or under utilized. It also has been shown that the combination of dispatching rules is highly dependent on the specific Fab [14].

In order to solve the scheduling problem that integrates processing tools and vehicles, Lin et al. propose a hybrid push/pull dispatching rule, including the procedures for vehicle selecting lot (VSL) and lot selecting vehicle (LSV), to improve the photobay performance [15]. Min and Yih propose a combined simulation and neural network approach in which simulation is used to collect data related with the system status, performance and change in dispatching rules, and then its results are fed into a neural network to obtain related knowledge [16]. Tyan et al. propose a state-dependent policy for achieving better system performance [17]. However, these methods are not capable of handling complex stochastic semiconductor manufacturing environments.

Through a survey on the use of discrete event simulation for manufacturing system design and operation, Smith shows that simulation has proven to be an extremely useful analysis tool for developing dispatching rules [18]. Um et al. gives a simulation design and analysis of a flexible manufacturing system with an automated guided vehicle system (AGVs) [19]. Hung and Chen explore a simulation-based dispatching rule and a queue prediction dispatching rule for searching better dispatching rules to reduce flow times, while maintaining a high machine utilization [20]. Kim and Jeong present a simulation-based real-time scheduling methodology which considers both simulation models and decision time points for reselecting new rules [21]. Sivakumar also develops a discrete event simulation-based on-line multi-objective scheduling approach, which includes the use of a linear optimization algorithm and automatic simulation model generation, in a complex manufacturing environment [22]. Kim et al. present a simulation-based real-time scheduling (SBRTS) methodology in which lot scheduling rules and batch scheduling rules are selected from candidate rules based on information obtained from the simulation [23]. Jelong et al. present a hybrid approach that combined GA with the simulation [24]. However, only single performance indicator, namely the maximum completion time for the last job, is optimized in their study.

The contribution of this paper is twofold. First, it mainly concentrates on the "combinatorial dispatching" which has rarely been studied. Different from previous study on single performance indicator or weighted composite dispatching rules, a representative Pareto optimal solution subset is explored to handle multiple objectives. A GA-simulation procedure is applied to obtain this solution subset. This methodology could provide users the opportunities to select appropriate solutions according to their preferences. Second, a GA parameters optimization process is developed to handle complex environments. This process can improve the performance results by optimizing GA parameters even with the dramatically changing environments.

The remainder of this paper is organized as follows. In Section 2, the proposed methodology is given. Related dispatching rules, GA parameters optimization by response surface methodology and simulator parameters optimization will be presented in this section. Sections 3 and 4 describe GA controller and the simulation model in details, respectively. In Section 5, numerical experiments used to show that the proposed methodology can efficiently improve the overall system performance. Finally, Section 6 concludes the paper.

2. The GA based simulation optimization methodology

The proposed methodology that integrates processing tools and the AMHS is applied to real-time multiple-objective scheduling problems in semiconductor wafer fabrication systems. It mainly consists of two parts: the on-line real-time scheduler and the off-line scheduler, as illustrated in Fig. 1. The on-line scheduler is mainly employed to monitor and convey the information between users, semiconductor wafer fabrication system and off-line scheduler. It interacts with the off-line scheduler in ways of initializing the off-line scheduler and accepting the combinatorial dispatching rules optimized by the off-line scheduler. Then it applies the accepted combinatorial dispatching rules to the real system. Additionally, it can also be used to determine the points of time when new rules are reselected.

Kim and Jeong [21] provide three types of points of time for reselecting dispatching rules: the beginning of each planning horizon, the time when a major system disturbance occurs and the time when the differences between actual and estimated performance values exceed a pre-determined limit. They also categorize the system disturbances into two levels: major disturbances and minor disturbances. Major disturbances include arrivals of urgent jobs and major machine breakdowns that require a long repair time or for which the repair time cannot be estimated. On the other hand, the machine breakdowns for which the repair time is estimated to be short are considered as minor disturbances. In this paper, this method is adopted to determine the points of time when rules are reselected.

The off-line scheduler consists of four modules: GA parameters optimization module, simulation parameters optimization module, GA controller and the simulation model. The former two modules are used to search for better GA parameters (such as population size, crossover rate, mutation rate and stopping criterion) and simulation parameters (simulation length and number of replications), respectively. They are often conducted independently of the GA controller after receiving trigger signals from users or the on-line scheduler. The triggering signals mainly come from the system environment changes (such as material plan change and processing tool breakdown) or user requests.

GA controller and the simulation model are the most important modules of the off-line scheduler. Iterative runs of GA controller and the simulation model compose the iterative GA-simulation procedure which aims at finding the optimal combinatorial dispatching rules. The procedure is as follows. First, the GA controller produces different chromosome individuals which encode specific code numbers and meanwhile the simulation model selects the corresponding dispatching rules according to the code numbers.

Please cite this article in press as: Chang X, et al. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. J Manuf Syst (2013), http://dx.doi.org/10.1016/j.jmsy.2013.07.001

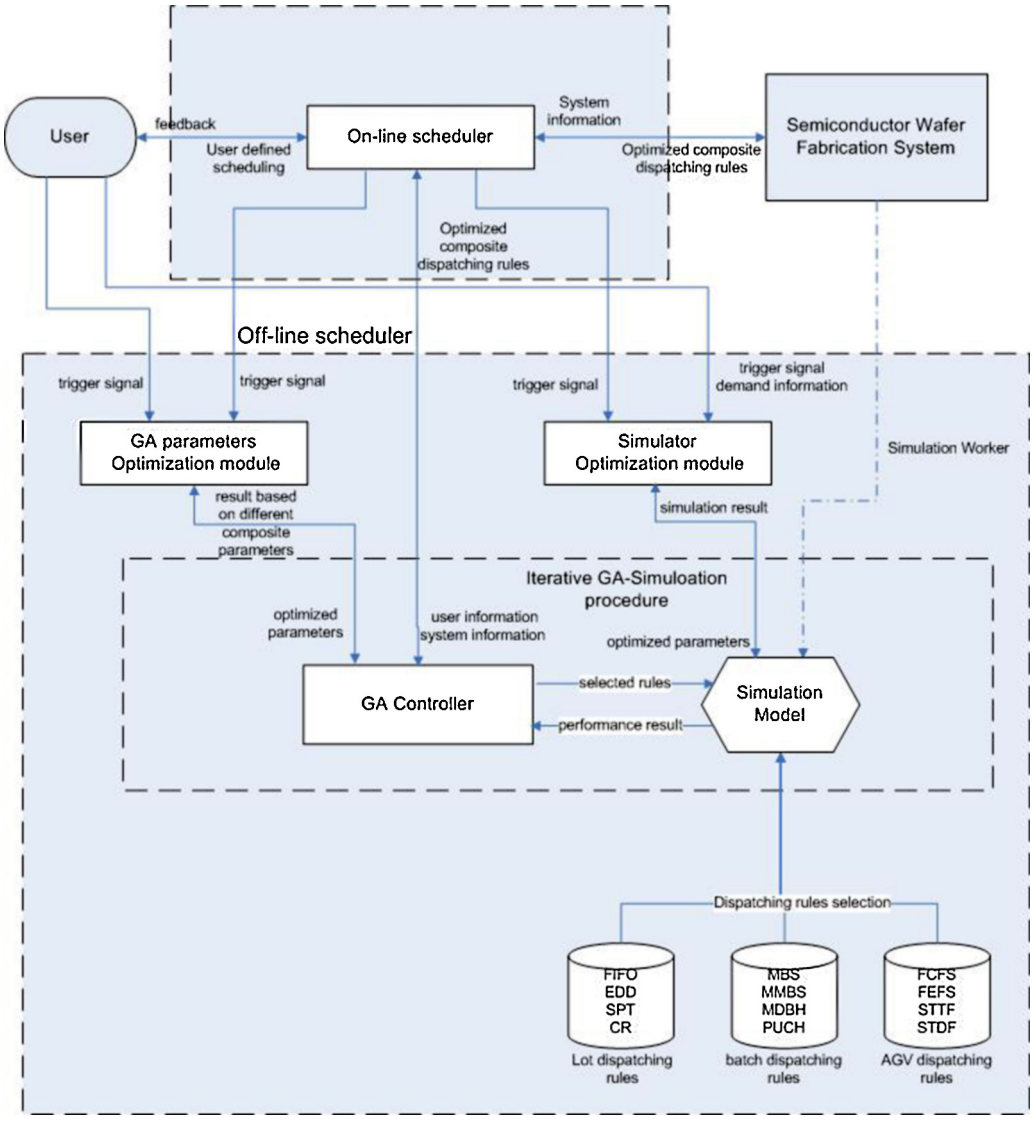

Fig. 1. GA based simulation optimization methodology.

Second, the simulation model will run for a period of time to estimate the performance results and pass them to the GA controller. Then, the GA controller will evaluate the fitness of each individual. Finally, the GA controller produces new individuals in terms of the fitness until the stopping criterion is satisfied. Individuals with better fitness represent better combinatorial dispatching rules. In this study, four performance indicators are considered to evaluate the fitness including WIP level, average cycle time, average delay and average hot lot delay. These fitness criteria are closely related to the throughput, satisfaction level of customers and capability for dealing with urgent jobs. The dispatching rules for single processing tools, batch processing tools and the AGVs are discussed below.

2.1. Dispatching rules: lot dispatching, batch dispatching and AGV dispatching

Effective dispatching mechanism is a widely used strategy to achieve high performance due to low computational requirements, ease of implementation and intuitive appeal. Chen and Cochran investigate the manufacturing rules for production planning in terms of multiple performance metrics and provide their effectiveness based on different factory conditions [25]. In practice, a good mechanism to dynamically select good rules is required. In this study, three categories of dispatching rules are discussed: lot dispatching, batch dispatching and AGV dispatching.

The commonly used lot dispatching rules include both single-attribute rules and multi-attribute rules. The single-attribute rules consists of first in first out (FIFO), time in system (TIS), EDD, SPT, shortest remaining processing time (SRPT) and others, while the multi-attribute rules include critical ratio (CR), slack (SL), process time/time in system (PT/TIS), etc. Considering the potential for offering a better solution to the problem, ten candidate rules, including FIFO, Pri-FIFO, SPT, Pri-SPT, SRPT, Pri-SRPT, EDD, CR, Pri-CR and SL, are selected as lot dispatching rules in this study. Here, the prefix "Pri-" is used to indicate a corresponding prioritized dispatching rule for hot lots. For example, Pri-FIFO means that the hot lots should be processed prior to normal lots firstly and meanwhile both the prioritized lots and the normal lots are dispatched by FIFO. The rules without the prefix "Pri-" indicate that both prioritized lots and normal lots are handled equivalently.

About the batch dispatching, four kinds of rules are taken as candidates in this study including minimum batch size rule (MBS), modified minimum batch rule (MMBS), modified dynamic batching heuristic (MDBH) and process urgency classification heuristic (PUCH) [26]. In addition, four kinds of AGV dispatching rules are

Please cite this article in press as: Chang X, et al. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. J Manuf Syst (2013), http://dx.doi.org/10.1016/j.jmsy.2013.07.001

chosen: first come first serve (FCFS), first encounter first serve (FEFS), shortest travel time first (SATTF) and EDD.

2.2. GA parameters optimization by response surface methodology

The results from GA are often sensitive to the search parameters like population size, number of generations, crossover rate and mutation rate. Therefore, a proper combination of these parameters often makes sense when computational time is limited. In this paper, a set of designed experiments are performed by the response surface methodology to obtain these proper GA parameters.

Both the first-order and the second-order models are widely used in response surface methodology to obtain an optimal response. This methodology can be adopted even when little information is known about the relationship between explanatory variables and response variables. Considering that the response of the first-order model is a linear function of explanatory variables and it may show a significant lack of fit on estimating the GA parameters, the second-order model including interaction terms is selected in this study. Pongcharoen et al. indicate that the second-order model can efficiently explore the effect of different levels of GA parameters [27]. It has been shown that three factors affect the performance mostly: *P/G* (population/generation), *C* (crossover rate), *M* (mutation rate). In this paper, these three factors are used to construct the response surface. The steps of the response surface method are given in the following:

Step 1: Select appropriately three levels for each of the three factors while satisfying the constraint $P_1 \times G_1 = P_2 \times G_2 = P_3 \times G_3$ to make sure the computation time is fixed.
Step 2: Conduct 27 ($27 = 3^3$) experimental treatments from the combination of the three levels of the three factors, respectively.
Step 3: Implement the regression analysis according to the results obtained in Step 2 and the regression model given in Eq. (1). Keep the factors with the value $P \leq 0.05$, which are statistically significant with a 95% level of confidence, and meanwhile ignore the factors with the value $P > 0.05$ to obtain the objective prediction function (the resulting response surface).

$$\bar{Y}_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1^2 + \beta_5 x_2^2 + \beta_6 x_3^2 + \beta_7 x_1 x_2 + \beta_8 x_2 x_3 + \beta_9 x_1 x_3 + \beta_{10} x_1 x_2 x_3 \quad (1)$$

where $\beta_{10} x_1 x_2 x_3$ is the interaction term, $x_1 = P/G$, $x_2 = C$, $x_3 = M$, β_i is the coefficient of the ith regression term, and $\bar{Y}_i$ is the mean of n replications.
Step 4: Optimize the objective with new optimal GA parameters.

2.3. Simulation parameters optimization

Simulation length and number of replications are main parameters affecting the accuracy and efficiency of the simulation. Generally, a longer-time simulation run gives more reliable estimates of system performances. However, in a dynamic manufacturing environment, a long simulation run may not be applicable due to frequently reselecting combinatorial dispatching rules. Considering that simulation is a time-consuming process, simulation length is mainly determined according to both time cost and the time intervals of reselecting dispatching rules.

Due to random variations in a discrete-event simulator, a large number of replications are needed to obtain reliable results. However, the simulation optimization is often time-consuming. In order to effectively to estimate the performance results in an acceptable time level, an adequate confidence level is required. Given an initial number of replications, N_0, and the corresponding half-width of the 95% confidence interval, H_0, a simple and commonly used way to estimate the required numbers of replications is given by Kelton et al. [28], which is as follows:

$$N = N_0 \times \left(\frac{H_0}{H}\right)^2 \quad (2)$$

where N and H are the required number of replications and the desired half-width, respectively. Taking the sample variance into consideration, Banks et al. provide another more precise way showed below [29].

$$\hat{N}^* = \left\lceil \frac{100 t_{N-1,\alpha/2^{s(N)}}}{H_{\text{required}} \bar{X}(N)} \right\rceil \quad (3)$$

where H_{required} is the confidence interval of a specified precision and $\hat{N}^*$ is the estimated number of replications. Hoad et al. test this method and show that it is stable only for larger value of N [30]. So they set a threshold for $\hat{N}^*$ and propose a heuristic framework for automated selection of the number of replications. The same heuristic framework is adopted in this paper to determine the number of replications.

The simulation parameter optimization module performs the function of determining the simulation length based on the time intervals of reselecting dispatching rules. It is also employed to determine the number of replications.

3. GA controller and the GA-simulation procedure

Some performance indicators, like throughout, WIP, cycle time, variance of cycle time, delay time and resource utilization rate, play important roles in evaluating various aspects of the semiconductor wafer fabrication system. To avoid evaluating the system performances partially, four commonly used performances, including WIP, average cycle time, average delay and average hot lot delay, are adopted in this study.

Konak et al. indicate that there are three general approaches for multi-objective optimization [31]. The first one is to combine the individual objective functions into a single combinatorial function, such as utility theory and weighted sum method. The second one is to move all but one objective to the constraint set. The third one is to determine an entire Pareto optimal solution set or a representative subset. The first approach requires decision-makers to select appropriate weights and the second one need users to establish the constraint values. In practice, it could be very difficult to precisely and accurately select these weights and values. And different weights or values can sometimes lead to quite different solutions. To overcome the weakness, in this paper, a third approach that evaluates the fitness of chromosomes with different random combinatorial weights in different generations is adopted in the proposed GA controller to find a Pareto optimal solution subset. This approach is similar to a specific MOGA method [32]. Compared with the former two approaches, it can be used to increase the diversity of the population, which is helpful to improve the results.

The iterative GA-simulation procedure is a random search technique. The GA controller is applied to search for better combinatorial dispatching rules by making the initial population evolves toward a population that is expected to contain the best solution while the simulation is used to estimate the objective performances. In this paper, the GA controller also plays an important role in initializing, controlling, modifying and running the simulation model. Once a simulation run is finished, the objective parameter values are recorded by the GA controller and analyzed during the fitness evaluation process. A merit of this infrastructure comes from the mutual independence of the GA controller and the simulation model. A new alternative GA-simulation procedure can be easily obtained by modifying the simulation model or the GA

Please cite this article in press as: Chang X, et al. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. J Manuf Syst (2013), http://dx.doi.org/10.1016/j.jmsy.2013.07.001

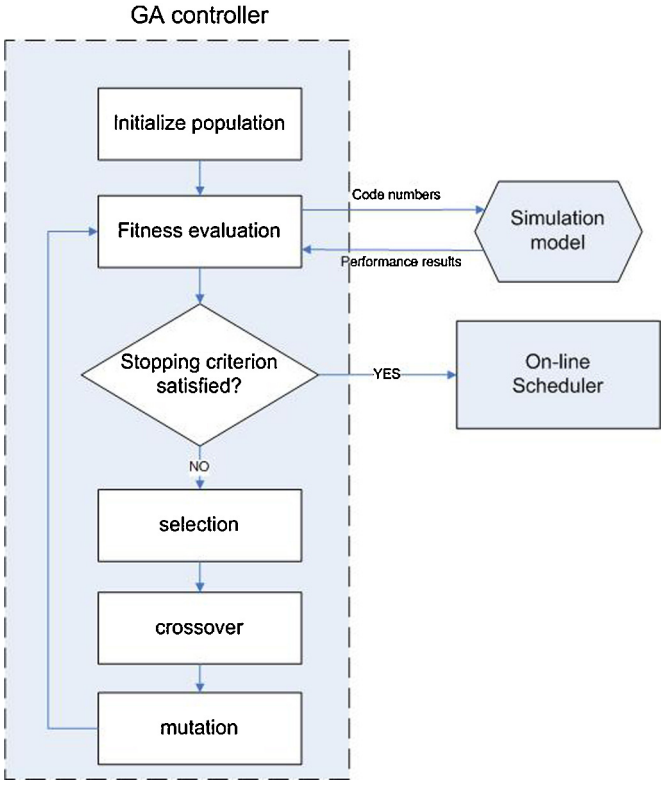

Fig. 2. The GA controller.

controller, respectively. At the same time, new parameters for the GA controller can also be selected while keeping the simulation model unchanged. The main processes of the GA-simulation procedure are shown in Fig. 2.

3.1. Population initialization and gene coding

A chromosome is represented as follows: $X=(x_1, x_2,\ldots, x_m)$, where m equals to the total numbers of processing tools and AGVs. Each gene in a chromosome, x_i, represents a specific dispatching rule from the relative candidate rules that are illustrated in Section 2.1. The first generation of population is initialized by randomly generating genes for each chromosome. In this paper, a gene will have an integer value ranging from 1 to 10 representing lot dispatching rules. Batch dispatching rules and AGV dispatching rules will be represented by codes ranging from I to IV and codes ranging from ① to ④, respectively. The candidate dispatching rules and the corresponding codes are shown in Table 1.

In real time environment, it is reasonable for processing tools to apply different dispatching rules. However, it is not easy to fulfill different AGV dispatching rules for all vehicles. Therefore, in this paper, all AGVs are set to fulfill the same dispatching rule, which means only one gene is occupied by the AGV dispatching rules in an individual chromosome.

3.2. Fitness evaluation

A fitness evaluation method is developed to obtain a Pareto optimal solution subset for the multi-objective optimization. The procedure of this method is given in the following.

Step 1: In order to deal with the unit difference between the four objectives, an appropriate assignment of fitness value for each objective is implemented. Take the CT as an example, the fitness

Table 1
The candidate dispatching rules.

Lot dispatching rules		Batch dispatching rules		AGV dispatching rules	
No.	Rules	No.	Rules	No.	Rules
1	FIFO	I	MBS	①	FCFS
2	Pri-FIFO	II	MMBS	②	FEFS
3	SPT	III	MDBH	③	SATTF
4	Pri-SPT	IV	PUCH	④	EDD
5	SRPT				
6	Pri-SRPT				
7	EDD				
8	CR				
9	Pri-CR				
10	SL				

can be assigned as $f=(T_{max}-T_c)/(T_{max}-T_{min})$, where T_{max}, T_{min} are the estimated maximum and minimum value of CT in real time environment, respectively. T_c is the performance result estimated by the simulation model. Fitness values for the other three objectives are assigned in a similar way.

Step 2: Assign a fitness value to each chromosome by performing the following steps:

Step 2.1: Generate a new series of random numbers u_k in [0,1] for each objective k, $k=1, 2,\ldots, K$ (in this study, $K=4$).

Step 2.2: Compute the random weight of each objective k as $w_k = (1/u_k)\sum_{i=1}^{k} u_i$.

Step 2.3: Compute the fitness of chromosome j as $F_j = \sum_{k=1}^{K} w_k f_{kj}$, $j=1, 2,\ldots, J$. Where J is the number of chromosomes in one generation and f_{kj} represents the fitness of objective k for chromosome j.

Step 3: Find the maximum and minimum value of F in this generation, namely, F_{max} and F_{min}.

Step 4: Check if there is any chromosome in this generation that is non-dominated by each other and any chromosome in set S, which is the set used to store non-dominated chromosomes found during the search so far. If so, update S by adding this chromosome and removing the corresponding dominated one.

Note that the set S is the desired Pareto optimal solution subset. As combinatorial weights are random for each generation, it is possible for the GA controller to search any solution space of the problem, which increases the population diversity.

3.3. Selection

The basic idea of selection is that the stronger individual with larger fitness should be selected with a larger probability. A most common used method is the roulette-wheel principle given in Eq. (4).

$$P_j = \frac{F_j}{\sum_{i=1}^{J} F_i} \tag{4}$$

where J is the number of chromosomes. P_j represents the probability that the jth chromosome will be selected and F_j represents the fitness of the jth chromosome.

Considering that the elitism mechanism improves performances according to Konak et al., an integrated strategy of the roulette-wheel principle and the elite strategy is adopted in this paper [31]. The specific method of the elite strategy in this study is to randomly remove one chromosome from the new generation and meanwhile to add one from set S after the mutation process.

Please cite this article in press as: Chang X, et al. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. J Manuf Syst (2013), http://dx.doi.org/10.1016/j.jmsy.2013.07.001

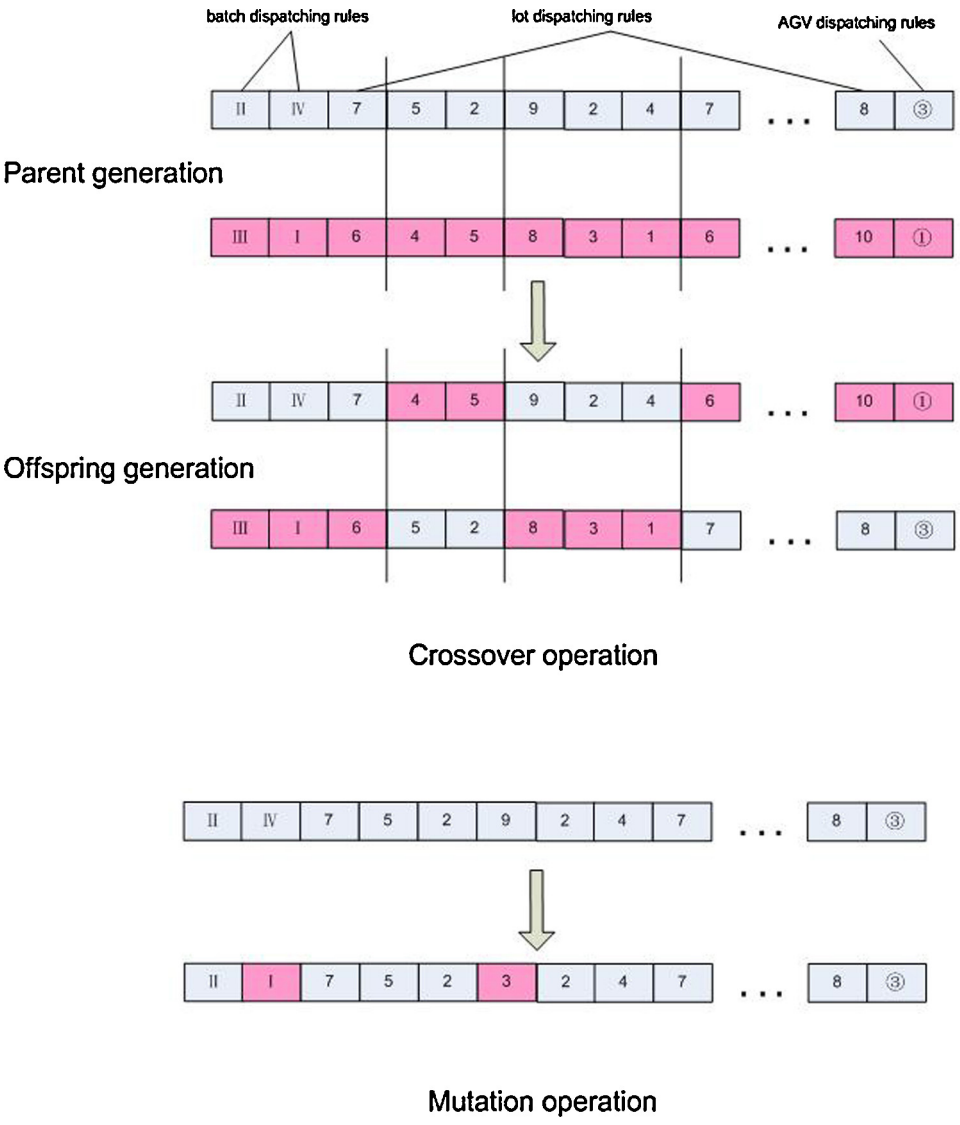

Fig. 3. The crossover and mutation operation.

3.4. Crossover operation and Mutation operation

The crossover operation is applied to the selected chromosomes with a probability of crossover rate. A three-point crossover strategy is adopted in this paper (see Fig. 3). The crossover operation is used to guide the direction in searching for better solutions because the short good ranges are much easier to preserve and therefore it is easy to reproduce a growing number of better solutions. The following step is the mutation operation. Each gene in the chromosome will randomly be assigned to another dispatching rule with the probability of mutation rate (see Fig. 3). Because of the mutation operation, any solution space of the problem could be randomly reached from any other solution.

4. The simulation model

The simulation model evaluates the individual chromosome by measuring the system performance in terms of WIP level, average cycle time, average delay and average hot lot delay. It simulates the operating processes of the integrated delivery system over a specified period of time.

In the simulation, each lot is represented as an entity with various attributes that reflects the specified information on the entity. For example, the time that a lot staying in the system can be reflected by the attribute "time in system". AGVs and processing tools are viewed as resources which require special time to transfer or process entities. System variables, such as NQ (Number of Queues), NQOU (NQ of upstream processing tools), NQOD (NQ of downstream processing tools), reflect the related information of the resources. The rules of selecting entities to process or transfer are based on entity attributes and system variables. For example, EDD for lot dispatching rules is implemented in the way of selecting the entity that contains the lowest value of attribute "due date" first.

The flows of lots and information, which motivate the implementation of the simulation model, are shown in Fig. 4. First, lots, represented by entities in the simulation model, are transferred to the current processing station *i* by AGVs and moved to the queues of the single processing tool or the batch processing tool. Meanwhile, related attributes are initialized and stored. Second, the idle processing tools select one or more lots to process based on the dispatching rules and the corresponding attributes or variable values. Whenever a process is finished, both attributes for the lots that have not been selected and the system variables are required to be updated. Then, the processing tool can select another one or more lots to conduct the next processing. The course of transfer is similar to that of processing. Lots waiting for transfer are selected

Please cite this article in press as: Chang X, et al. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. J Manuf Syst (2013), http://dx.doi.org/10.1016/j.jmsy.2013.07.001

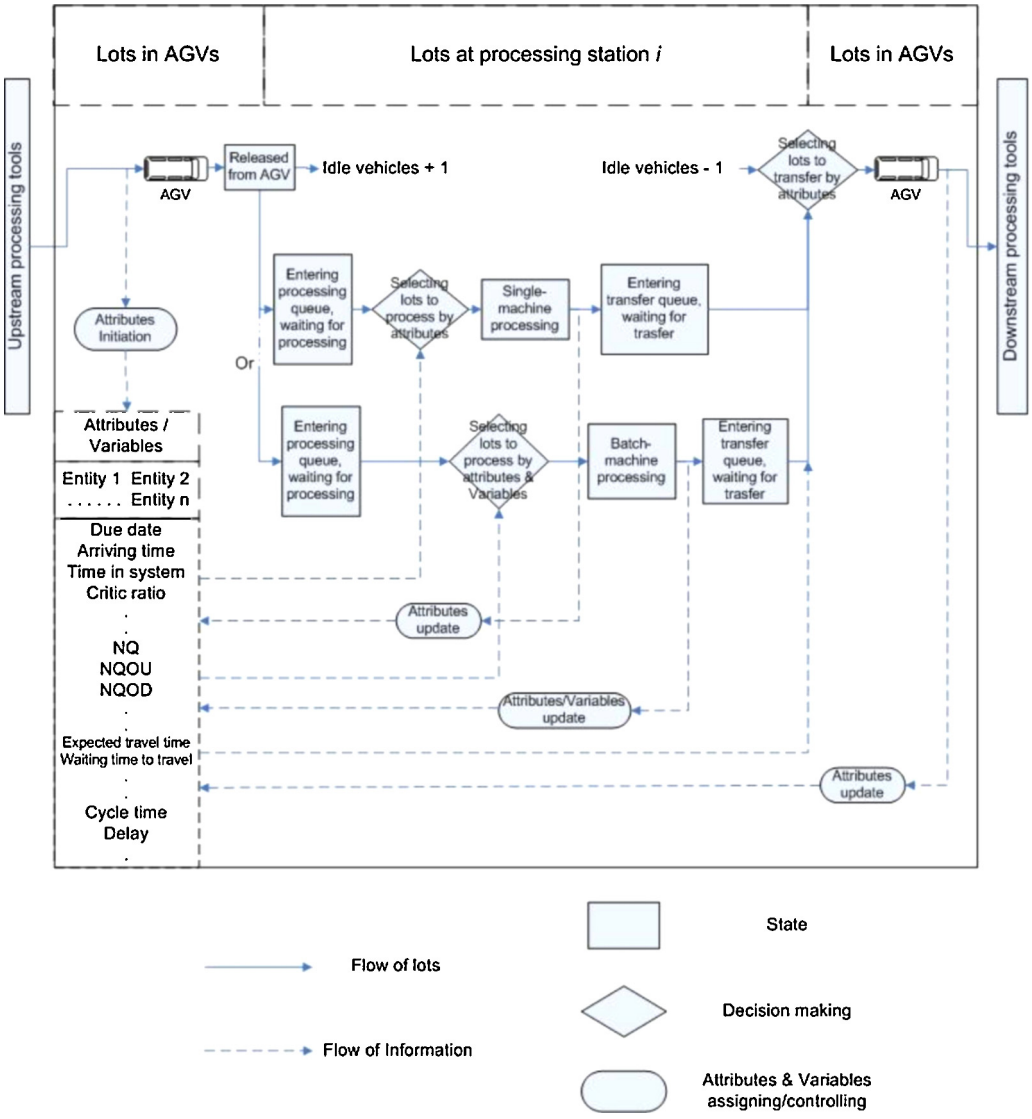

Fig. 4. The structure of simulation model at processing station *i* the simulation model with two hierarchical levels.

by idle AGVs. Attributes and variables need to be updated if one lot is selected. Simultaneously, the corresponding attributes for the selected lot are required to be removed from the current processing station to the next processing station.

5. Numerical experiments

In this section, a numerical experiment is designed for a bay system consisting of 4 types of lots (including a hot lot), 16 processing tools (including 2 batch tools) and the AMHS (see Fig. 5). Most parameters (such as processing times and vehicle speed) are same as those from SEMATECH's phase II report [33]. Due to the existence of crossover turntables in the AMHS of semiconductor wafer fabrication systems, the structure of AMHS becomes more complicated. Take the studied bay as example, there are three possible ways to convey lots from U2 to L4: U2 → U8 → stock → L4, U2 → U6 → T4 → stock → L4 and U2 → U3 → T2 → stock → L4, respectively. In the experiments, a discrete-event simulation software, Arena, is selected as the simulation tool for its strength of automatically selecting shortest-distance ways.

5.1. The stopping criterion

Different from other methods, simulation is much more time-consuming. Considering the simulation time is generally determined by number of populations in one generation and number of generations, a proper selection of these two parameters is required to make a trade-off between "accuracy" and "efficiency". The next simulation experiment shows how to select these two parameters properly. The results are shown in Fig. 6, in which the number of chromosomes of each generation is set as 12. Each *Y*-value represents an objective value for a chromosome with best fitness in current generation. It can be seen that three stages exist in the whole evolutionary life. The initial generation to about the 15th generation concludes the first stage. In this stage, the whole population evolves to the optimal direction with a fast speed. Stage 2 ranges from the 16th generation to about the 35th generation. This stage plays an important role on searching non-dominated solutions. So the Pareto optimal subset is gradually explored during this stage. The remaining generations conclude the last stage. This stage only generates few new non-dominated points and most of best points in this stage can simultaneously be found in stage 1 or 2. Therefore, it can be shorten or removed to save time. In this study,

Please cite this article in press as: Chang X, et al. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. J Manuf Syst (2013), http://dx.doi.org/10.1016/j.jmsy.2013.07.001

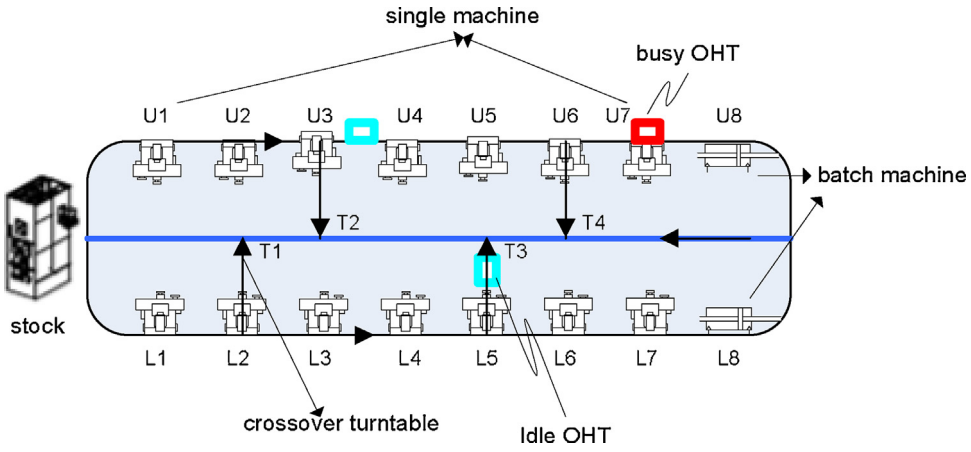

Fig. 5. The experimental bay with processing tools and AMHS.

the total number of chromosomes generated by the GA controller is fixed at $12 \times 40 = 480$.

5.2. Evaluation criteria for the Pareto optimal solutions subset

As stated in Section 2.2, *P*/*G*, *C* and *M* are the GA parameters that affect the performance most. They are all considered at three different levels to construct the response surface. With the total number of chromosomes of 480, the factors and levels used in this experiment are shown in Table 2. Since multiple performance objectives are considered, the quality of Pareto optimal solution subsets, generated by different composite parameters, needs to be assessed by quality metrics. These metrics can generally be categorized into unary performance metrics and binary performance metrics.

Table 2
Experimental factors.

Factors	Levels		
	1	2	3
P/*G*	12:40	16:30	20:24
C	0.4	0.6	0.8
M	0.05	0.15	0.25

The first type of unary performance metric, concerning about assessing the number of Pareto optimal solutions or the range of objective values, includes ratio of non-dominated individuals (RNI) [34], error ratio (ER) [35], overall non-dominated vector generation and ratio (ONVG) [35], Pareto dominance indicator (NR) [36], number of distinct choices (NDC_{μ}) [37], overall Pareto spread (OS) [37],

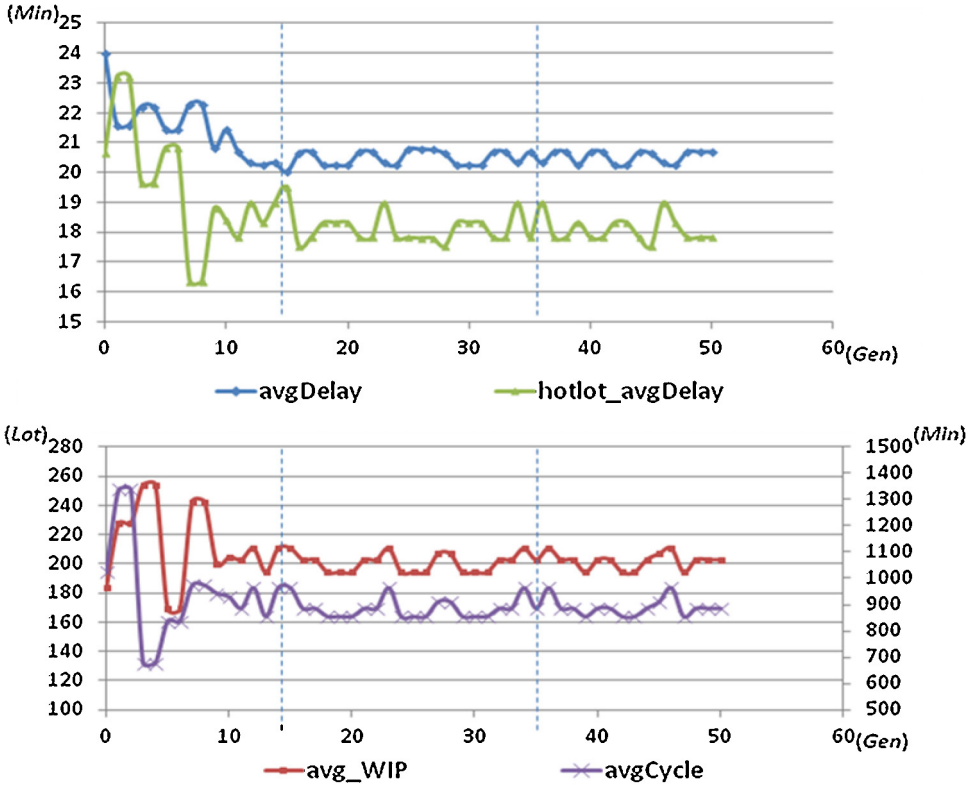

Fig. 6. The results of the GA-simulation procedure.

Please cite this article in press as: Chang X, et al. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. J Manuf Syst (2013), http://dx.doi.org/10.1016/j.jmsy.2013.07.001

etc. The second type of unary performance metric, which focuses on measuring the closeness of the estimated solution set to the true Pareto front, includes generational distance (GD) [35], maximum Pareto front error (MPFE) [35], uniform distribution (UD) [34], hypervolume difference (HD) [38], etc. The binary performance metrics, which are based on unary quality indicators or are direct comparison metrics, include ε-indicator I_ε [39], enclosing hypercube indicator [39], D-metric (coverage different metrics) [38], etc.

Wu and Azarm present five metrics that reflect various aspects of the estimated set, namely, HD, OS, AC (Accuracy of the Observed Pareto Frontier), number of distinct choices and cluster [37]. In this experiment, a measure R which combines metric OS and AC is adopted to qualify each subset. The measure R, transforming each subset into one real number, is described by Eq. (5).

$$R_t = \frac{OS_t}{OS_{avg}} + \frac{AC_t}{AC_{avg}} \quad \forall t = 1, 2, \ldots, T \tag{5}$$

where R_t represents the R value of the tth set. T is the total number of subsets generated by different combinatorial GA parameters. OS_{avg} and AC_{avg} are the average OS and AC values among all subsets, respectively.

To compute the OS and AC values, estimates of the ideal and maximal points, which are generally referred to as good and bad points, respectively, are required. In this experiment, the good point can be estimated as:

$$P^g = \left\{ (O_1^g, O_2^g, O_3^g, O_4^g) \middle| O_i^g = \min_{t=1}^{T} \min_{k=1}^{np_t} O_{t,k,i}, \quad \forall i = 1, 2, 3, 4 \right\} \tag{6}$$

where $O_{t,k,i}$ represents the ith objective value of the kth solution of the tth set and np_t is the number of points (or solutions) in the tth set. Similarly, the bad point can be assigned with maximum objective values among all solutions.

For computational convenience, all of the objective values are scaled by using Eq. (7) so that the scaled good point becomes $\tilde{P}^g = (0, 0, 0, 0)$ and the scaled bad point becomes $\tilde{P}^b = (1, 1, 1, 1)$.

$$\overline{O_{t,k,i}} = \frac{O_{t,k,i} - O_i^g}{O_i^b - O_i^g} \quad \forall t = 1, 2, \ldots, T; \quad k = 1, 2, \ldots, np_t; \quad i = 1, 2, 3, 4 \tag{7}$$

The metric OS quantifies how widely an estimated set spreads over the objective space. The OS value of the tth set is designed as follows:

$$OS_t = \frac{\prod_{i=1}^{4} \left| \max_{k=1}^{np_t} O_{t,k,i} - \min_{k=1}^{np_t} O_{t,k,i} \right|}{\prod_{i=1}^{4} \left| O_i^b - O_i^g \right|} = \prod_{i=1}^{4} \left| \max_{k=1}^{np_t} \overline{O_{t,k,i}} - \min_{k=1}^{np_t} \overline{O_{t,k,i}} \right| \quad \forall t = 1, 2, \ldots, T \tag{8}$$

The metric AC indicates how good an estimated set is. It can be explained by the concept "impreciseness". If there exist additional Pareto solutions that are undetected, then such solutions could not belong to either the dominant region (S_{do}) or the inferior region (S_{in}). The impreciseness of the approximation comes from the fact that not all the points can be simultaneously on the Pareto frontier. Fig. 7 gives an example of the dominant region and inferior region of a set with four points and two objective values. The AC value of the tth set is obtained as follows:

$$AC_t = \frac{1}{1 - S_{do}(t) - S_{in}(t)} \quad \forall t = 1, 2, \ldots, T \tag{9}$$

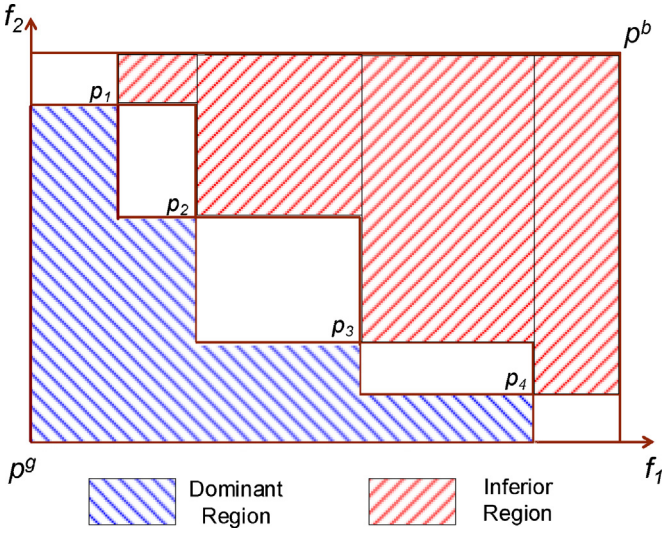

Fig. 7. The dominant and inferior region of a set $t = \{P_1, P_2, P_3, P_4\}$.

Meanwhile, the dominant region of the tth set can be estimated by Eq. (10). The inferior region can also be estimated in a similar way.

$$S_{do}(t) = \sum_{r=1}^{np_t} \left\{ (-1)^{r+1} \sum_{k_1=1}^{np_t - r + 1} \cdots \sum_{k_l = k_{l-1}+1}^{np_t - (r-l+1)+1} \cdots \sum_{k_r = k_{r-1}}^{np_t} \prod_{i=1}^{4} \left[1 - \min_{j=1}^{r} \overline{O_{t,k_j,i}} \right] \right\} \quad \forall t = 1, 2, \ldots, T \tag{10}$$

5.3. GA parameters optimization

Since measure R has been defined, it can be used to optimize the GA parameters by the response surface method according to Eq. (1). The treatment results and the corresponding regression results are shown in Tables 3 and 4, respectively.

Factors with the value $P \leq 0.05$ are statistically significant with a 95% level of confidence. It can be seen that C, C^2 and M^2 are the significant terms. None of the other higher order interactions is significant. So the objective prediction function is: $R = 1.127 + 3.257C - 2.789C^2 - 5.129M^2 = 2.078 - 2.789 \times (C - 0.58)^2 - 5.129M^2$. The model suggests that the R value can be maximized with smallest level of M, namely 0.05, and with C equaling to 0.58. Fig. 8 shows that average R measure value can be improved when C is close to 0.58.

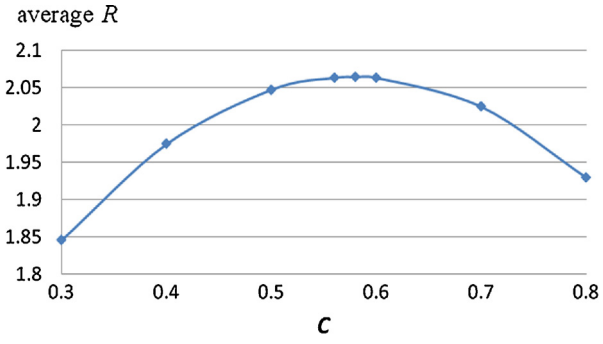

Fig. 8. Average R values versus different C.

Table 3
Simulation results for the experimental design.

No.	P/G	C	M	R_{avg}	σ
1	12:40	0.4	0.05	1.9933	0.0113
2	12:40	0.4	0.15	2.0325	0.0094
3	12:40	0.4	0.25	1.9664	0.0087
4	12:40	0.6	0.05	2.0529	0.0143
5	12:40	0.6	0.15	2.1542	0.0082
6	12:40	0.6	0.25	2.181	0.012
7	12:40	0.8	0.05	1.9157	0.0068
8	12:40	0.8	0.15	1.9801	0.0113
9	12:40	0.8	0.25	1.9376	0.0149
10	16:30	0.4	0.05	1.9474	0.0118
11	16:30	0.4	0.15	2.0856	0.0087
12	16:30	0.4	0.25	2.0698	0.0111
13	16:30	0.6	0.05	2.0685	0.0163
14	16:30	0.6	0.15	2.071	0.0086
15	16:30	0.6	0.25	2.0144	0.0112
16	16:30	0.8	0.05	1.8549	0.0131
17	16:30	0.8	0.15	1.9237	0.0079
18	16:30	0.8	0.25	1.8892	0.0132
19	20:24	0.4	0.05	1.9831	0.0124
20	20:24	0.4	0.15	2.102	0.0175
21	20:24	0.4	0.25	2.0716	0.013
22	20:24	0.6	0.05	1.9895	0.0112
23	20:24	0.6	0.15	2.0838	0.0084
24	20:24	0.6	0.25	2.0549	0.0119
25	20:24	0.8	0.05	1.8323	0.0054
26	20:24	0.8	0.15	1.8756	0.0073
27	20:24	0.8	0.25	1.8713	0.0104

Table 4
Regression analysis.

Predictor	Coefficient	Std. Error	T	P
(Constant)	1.127	.214	5.275	0.000
P/G	−.158	.365	−0.434	0.670
C	3.257	.526	6.191	0.000
M	1.158	1.032	1.121	0.279
$(P/G)^2$	.216	.220	0.983	0.340
C^2	−2.789	.384	−7.261	0.000
M^2	−5.129	1.537	−3.338	0.004
$(P/G) \times C$	−.324	.425	−0.762	0.457
$C \times M$	.809	1.485	0.544	0.594
$(P/G) \times M$	1.504	1.548	0.972	0.346
$(P/G) \times C \times M$	−2.121	2.489	−0.852	0.407

5.4. Effectiveness of the proposed methodology

Six scenarios are used to conduct an experiment for comparing the proposed GA based simulation optimization methodology with the single dispatching rules methods that are widely used in practice. The experimental results are given in Table 5, which indicate that the proposed methodology can significantly improve the overall performances of the semiconductor wafer fabrication system.

Another experiment is conducted to compare the proposed methodology with Jelong's hybrid approach [24]. Since this hybrid approach is used to solve single-objective problems, weighted sum method is adopted to combine the individual objective functions into a single combinatorial function. A Pareto subset can be constructed by replicating this hybrid approach with different random composite weights and by selecting non-dominated solutions. The experiment shows that the performance of the hybrid approach is close to that of the proposed methodology if the number of replications is nearly 120. The hybrid approach can just improve the final result a little bit, namely less than 2%, if the number reaches 300. However, the running time is almost 300 times than the proposed methodology. Therefore, the proposed methodology is applicable in practices.

Table 5
Average improvement of the proposed methodology.

Scenario (dispatching rules)	CT improvement (%)	WIP improvement (%)	Delay improvement (%)	Hot lot delay improvement (%)
FIFO-MBS-FCFS	17.7	24.7	22.7	88
EDD-MMBS-SATTF	18.1	35.6	16.4	30
SPT-MDBH-FEFS	20.2	17.1	33.9	62
SRPT-MMBS-FEFS	15.9	24.3	27.8	55
CR-MDBH-FCFS	17.8	19.5	9.2	29
SL-MBS-EDD	16.4	11.7	20	39

In this study, it takes about 1.5 h to implement a GA-simulation procedure on a personal computer with Core2, 2.4 GHz processor and 2 GB RAM when the replication number of simulation is set as 10 times. Considering that most of the processing times for each lot at each station are above 30 min, the proposed methodology can be considered to be adequate to solve practical problems. However, it may consume much more time when implementing the GA parameters optimization module because at least 27 GA-simulation procedures are required. So servers with better processors and RAM or parallel computers, which can simultaneously been used to execute GA-simulation procedures with different composite parameters, are needed to speed up the implementation.

6. Conclusions

This paper proposes a GA based simulation optimization methodology to solve a real-time multi-objective dispatching decision problem for the "integrated delivery" in a 300-mm semiconductor wafer fabrication Fab. The methodology, consisting of the on-line scheduler and the off-line scheduler, can optimize the combinatorial dispatching rules including lot dispatching, batch dispatching and AGV dispatching. Four modules (GA parameters optimization module, simulation parameters optimization module, GA controller and simulation model) compose of the off-line scheduler. They are used to optimize the parameters for searching better combinatorial dispatching rules. A numerical experiment is designed to verify if the GA parameters optimization module is able to improve the performances and another experiment is designed to compare the proposed methodology with the widely used single dispatching rule and weighted sum method. It has been seen that the proposed methodology outperforms single dispatching rule in overall performance indicators and can save much more time compared with weighted sum method. In order to deal with the multi-objective problems, a representative Pareto optimal solution subset is required. Since it may be hard for users to determine a solution from the subset, a good mechanism for finding a satisfied subset or selecting appropriate solutions for users will be the future research direction.

Acknowledgements

The work presented in this paper has been supported by grants from the National High-Tech Research and Development Program (863 Program) of China (2008AA04Z104), National Natural Science Foundation of China (71131005, 70871077) and Ph.D. Programs Foundation of Ministry of Education of China (20120073110029). The authors are grateful to the referees for their constructive comments and suggestions. The presentation of the paper has been significantly improved with their inputs.

References

[1] Gupta AK, Sivakumar AI. Job shop scheduling techniques in semiconductor manufacturing. The International Journal of Advanced Manufacturing Technology 2006;27(11–12):1163–9.

[2] Yang J, Chang TS. Multi objective scheduling for IC sort and test with a simulation testbed. IEEE Transactions on Semiconductor Manufacturing 1998;11(2):304–15.
[3] Li L, Qiao F, Jiang H, Wu QD. The research on dispatching rule for improving on-time delivery for semiconductor wafer Fab. 8th International Conference on Control, Automation, Robotics and Vision (ICARCV) 2004:494–8.
[4] Wu MC, Huang YL, Chang YC, Yang KF. Dispatching in semiconductor fabs with machine-dedication features. International Journal of Advanced Manufacturing Technology 2006;28(9):978–84.
[5] Monch L, Driebel R. A distributed shifting bottleneck heuristic for complex job shops. Computers and Industrial Engineering 2005;49(3):363–80.
[6] Sarin SC, Varadarajan A, Wang L. A survey of dispatching rules for operational control in wafer fabrication. Production Planning & Control 2011;22(1):4–24.
[7] Pierce NG, Yurtsever T. Value-based dispatching for semiconductor wafer fabrication – VBD. Proceedings of the 2000 IEEE/SEMI advanced semiconductor manufacturing conference 2000:245–9.
[8] Hsieh S, Hou KC. Production-flow-value-based job dispatching method for semiconductor manufacturing. International Journal of Advanced Manufacturing Technology 2006;30(7–8):727–37.
[9] Dabbas RM, Fowler JW. A new scheduling approach using combined dispatching criteria in wafer fabs. IEEE Transactions on Semiconductor Manufacturing 2003;16(3):501–10.
[10] Bahaji N, Kuhl ME. A simulation study of new multi-objective composite dispatching rules, CONWIP, and push lot release in semiconductor fabrication. International Journal of Production Research 2008;46(14):3801–24.
[11] Yang KM, Park JH, Kang KS. A study on the weighted composite dispatching rule in modular production systems. International Journal of Advanced Manufacturing Technology 2007;33:803–11.
[12] Cardarelli G, Pelagagge PJ. Simulation tool for design and management optimization of automated interbay material-handling and storage systems for large wafer fab. IEEE Transactions on Semiconductor Manufacturing 1995;8(1):44–9.
[13] Davis J, Weis M. Addressing automated materials handling in an existing wafer Fab. Semiconductor International 1995:125–8.
[14] Christopher J, Kuhl ME, Hirschman K. Simulation analysis of dispatching rules for automated material handling systems and processing tools in semiconductor fabs. In: Proceedings of the 2005 IEEE international symposium on semiconductor manufacturing. 2005. p. 84–7.
[15] Lin JT, Wang FK, Chang YM. A hybrid push/pull-dispatching rule for a photobay in a 300 mm wafer Fab. Robotics and Computer-Integrated Manufacturing 2006;22(1):47–55.
[16] Min HS, Yih Y. Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system. International Journal of Production Research 2003;41(16):3921–41.
[17] Tyan JC, Du TC, Chen JC, Chang IH. Multiple response optimization in a fully automated FAB: an integrated tool and vehicle dispatching strategy. Computers & Industrial Engineering 2003;46(1):121–39.
[18] Smith JS. Multiple response survey on the use of simulation for manufacturing system design and operation. Journal of Manufacturing Systems 2003;22(2):157–71.
[19] Um I, Cheon H, Lee H. The simulation design and analysis of a flexible manufacturing system with automated guided vehicle system. Journal of Manufacturing Systems 2009;28(4):115–22.
[20] Hung YF, Chen IR. A simulation study of dispatch rules for reducing flow times in semiconductor wafer fabrication. Production Planning & Control 1998;9(7):714–22.
[21] Kim YD, Jeong KC. A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules. International Journal of Production Research 1998;36(19):2609–26.
[22] Sivakumar A. Multiobjective dynamic scheduling using discrete event simulation. International Journal of Computer Integrated Manufacturing 2001;14(2):154–67.
[23] Kim YD, Shim OH, Choi B, Hwang H. Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility. IEEE Transactions on Semiconductor Manufacturing 2003;16(2):290–8.
[24] Jeong SJ, Lim SJ, Kim KS. Hybrid approach to production scheduling using genetic algorithm and simulation. International Journal of Advanced Manufacturing Technology 2006;28(1–2):129–36.
[25] Chen HN, Cochran JK. Effectiveness of manufacturing rules on driving daily production plans. Journal of Manufacturing Systems 2005;24(4):339–51.
[26] Kim YD, Kim JG, Choi B, Kim HU. Production scheduling in a semiconductor wafer fabrication facility producing multiple product types with distinct due dates. IEEE Transactions on Robotics and Automation 2001;17(15):589–98.
[27] Pongcharoen P, Hicks C, Braiden PM, Stewardson DJ. Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products. International Journal of Production Economics 2002;78(3):311–22.
[28] Kelton WD, Sadowski RP, Sturrock DT. Simulation with arena. third ed. New York: McGraw-Hill; 2004.
[29] Banks J, Carson JS, Nelson BL, Nicol DM. Discreteevent system simulation. 5th ed. Upper Saddle River, NJ: Prentice Hall; 2009.
[30] Hoad K, Robinson S, Davies R. Automated selection of the number of replications for a discrete event simulation. Journal of the Operational Research Society 2009:1632–44.
[31] Konak A, Coit D, Smith A. Multi-objective optimization using genetic algorithms: a tutorial. Reliability Engineering and System Safety 2006;91(9):992–1007.
[32] Murata T, Ishibuchi H. MOGA: multi-objective genetic algorithms. In: Proceedings of the 1995 IEEE international conference on evolutionary computation. 1995.
[33] Campbell E, Ammenheuser J. 300 mm Factory Layout and Material handling Modeling: Phase II report, International SEMATECH, Technology transfer #99113848B-ENG; 2000.
[34] Tan KC, Lee TH, Khor EF. Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons. Artificial Intelligence Review 2002;17(4):253–90.
[35] Van Veldhuizen DA, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Ph.D. Thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999.
[36] Goh CK, Tan KC. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Transactions on Evolutionary Computation 2009;13(1).
[37] Wu J, Azarm S. Metrics for quality assessment of a multiobjective design optimization solution set. Transactions of the ASME 2001;18(123).
[38] Zitzler E, Evolutionary algorithms for multiobjective optimization: Methods and applications, Ph.D. Thesis, Swiss Federal Institute of Technology Zurich; 1999.
[39] Zitzler E, Thiele L, Laumanns M, Foneseca CM, da Fonseca VG. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation 2003;7(2):117–32.

Please cite this article in press as: Chang X, et al. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. J Manuf Syst (2013), http://dx.doi.org/10.1016/j.jmsy.2013.07.001