Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

SIMULATION BASED MULTIOBJECTIVE SCHEDULE OPTIMIZATION IN SEMICONDUCTOR MANUFACTURING

Amit K. Gupta
Appa Iyer Sivakumar

School of Mechanical and Production Engineering
Nanyang Technological University
Nanyang Avenue
639798, SINGAPORE

ABSTRACT

In semiconductor manufacturing, it requires more than one objective such as cycle time, machine utilization and due date accuracy to be kept in focus simultaneously, while developing an effective scheduling. In this paper, a near optimal solution, which is not inferior to any other feasible solutions in terms of all objectives, is generated with a combination of the analytically optimal and simulation based scheduling approach. First, the job shop scheduling problem is modeled using the discrete event simulation approach and the problem is divided in to simulation clock based lot selection sub-problems. Then, at each decision instant in simulated time, a Pareto optimal lot is selected using the various techniques to deal with multiobjective optimization such as weighted aggregation approach, global criterion method, minimum deviation method, and compromise programming. An illustration shows how these techniques work effectively in solving the multiobjective scheduling problem using discrete event simulation.

1 INTRODUCTION

In semiconductor industry, the primary challenge is to maximize the throughput of the facility while responding rapidly to customer demands through low cycle times. This is also important in order to face the response of a highly dynamic market characterized by rapidly changing demands and product mixes with sometimes very brief product life cycles. The need for higher utilization is created by the capital-intensive nature of the constraint equipments such as testers. Therefore the effective scheduling of the semiconductor back-end is one of the key aspects in achieving these improvements.

There are four main stages in a typical IC manufacturing process: wafer fabrication, wafer sort, assembly cycle and final test. The flow between these processing stages is illustrated in Figure 1. Among these stages, the wafer fabrication and wafer sort are usually known as the front-end and the IC assembly and testing are known as the back-end. As these back-end equipment are very highly capital intensive in nature, their effective utilization is very crucial for the factory performance. Also, the efficient scheduling of these test operations play an important role in the on-time delivery of the products and thus on the customer satisfaction.

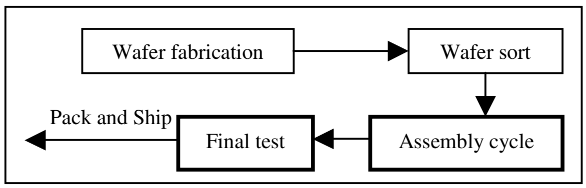

Figure 1: IC Manufacturing Process Flow

The focus of this paper is on the optimization of multiobjective scheduling of these test operations based on discrete event simulation. The paper is organized as follows: Section 2 describes the simulation based scheduling and its applications; Section 3 formulates the scheduling problem in semiconductor back-end, using discrete event simulation; Section 4 briefs about Pareto optimal solution in multiobjective optimization; Section 5 describes various approaches, with their merits and limitations, to deal with multiobjective optimization; Section 6 formulates the multiple objectives in semiconductor manufacturing; Section 7 describes this application by an illustrative example of a typical situation; Section 8 discusses the impact of the proposed techniques; and Section 9 concludes the paper with future work.

2 SIMULATION BASED SCHEDULING

In recent developments, simulation is not just a tool to mimic the real-world system for analyzing it, but it has become a popular technique for developing production schedules and dispatch lists in manufacturing environments

(Mazziotti and Horne 1997, Morito and Lee 1997, Sivakumar 1999). An important aspect of simulation-based scheduling is that it uses actual customer orders and WIP information, and not arrival/demand data estimated from statistical distributions. Thus, keeping update with the real information, simulation offers the advantage of developing a feasible and accurate schedule in shorter computation times compared to some of the other techniques (Kiran 1998, Mazziotti and Horne 1997), even for the job shop scheduling problems which are considered as NP-hard (Pinedo, 1995). Davis (1998) and Sivakumar (2001) report the application of online simulation in complex manufacturing scheduling. In online simulation, one of the major advantage is that the simulator mimics the behavior of the actual system in an intuitive manner that enables the users to understand the logic of manufacturing systems (Hopp and Spearman, 1996). Further, the simulation based scheduling also serve as a tool for handling exceptions in the production plan, such as machine break-down, hot lots, etc., by generating the "what now" scenarios.

However, the simulation based scheduling differs from the typical simulation studies in scope of application, approach of modeling, conduct of experimentation and analysis of output as reported by Koh et al. (1996). Table 1 describes the summary of differences between the typical simulation studies and the simulation based scheduling of a shop.

Table 1: Summary of Differences

	Typical simulation studies	*Simulation based scheduling*
Scope	Design and analysis.	Operational planning.
Model	Stochastic model (random processes).	Deterministic flexible model.
Experiment	Extensive multiple runs for statistical variance according to the design of experiment.	Fewer shorter runs, experimenting different scheduling rules (or strategies).
Output	Statistical estimates of effects of various factors.	Operational plan with system performance parameters.

The application of simulation based scheduling at a semiconductor wafer fabrication of AMD Inc., using real time dispatch (RTD™) tool from AutoSimulation Inc. has been reported by Rippenhagen and Krishnaswamy (1998). They implemented the combination of Hunger Ratio and Critical Ratio as dispatching rules, in prior which were analyzed in an off-line mode using simulation modeling. Rules deemed beneficial via simulation were then transferred to the Manufacturing Execution Systems (MES) for controlling the order of processing, in order to avoid the starvation of possibly reoccurring bottleneck equipment.

Another similar type of application at Sony semiconductor wafer fabrication, using AutoSched package from AutoSimulation Inc. is reported by Watt (1998). In this work, the shop floor scheduling system was integrated with the fab MES and a number of other data sources including an integrated machine standards database, preventive maintenance scheduling and a Kanban stage calculation worksheet. Also, the simulation was used to verify Sony production rules and determine the most effective Kanban strategy for the fab.

3 PROBLEM FORMULATION

The scheduling problem in semiconductor manufacturing is considered as one of the complex job shop scheduling problems, which are generally formulated by conventional approaches like Branch and Bound method, Lagrangian relaxation approach or other optimization methods using Tabu search, Simulated annealing, Genetic algorithm etc (Kiran 1998, Pinedo 1995). But, these conventional search and optimization methods are generally intensive in computation time as even the simple manufacturing scheduling problems are NP-hard. The complexity of scheduling problem increases more in semiconductor manufacturing because of 1) the presence of different types of work-centers, 2) very large and changing varieties of the products, 3) sequence dependent set-up times, 4) re-entrant process flow, 5) dynamic nature of the problem, and 6) contradicting multiple objective functions, etc.

In modeling the scheduling problem of semiconductor manufacturing, the use of discrete event simulation method helps in overcoming many of the limitations of the conventional approaches (Sivakumar 2001). The most distinguished advantage of this method is that it avoids even formulating an NP-hard optimization problem and thus provides the optimum solution at that instant within a limited time. The operations are simulated on jobs using defined resources in discrete time. Figure 2 illustrates this concept (Sivakumar 2001).

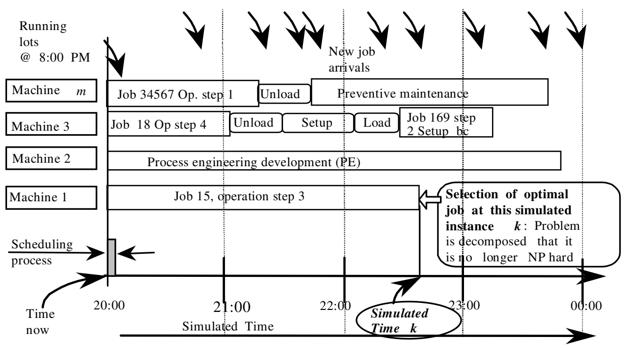

Figure 2: Simulation Based Scheduling

For example, in simulation, each machine selects its activity and operation in future time and as a result, all jobs available at that instance can be considered and all activities such as PM and sequence dependant setups can be scheduled. In this formulation, machine *m* at simulated time k has access to all the work in process (WIP) expected to be in the queue at time k, including those that would have arrived after the start of the simulation. Thus, the overall NP-hard scheduling problem gets reduced to the selection of a suitable lot on machine m at simulation instant k, which no longer remains a NP-hard problem. In other words, the issue of NP-hardness gets resolved when the overall scheduling problem is decomposed in to the sub-problems of lot selection for each local work center at each decision instance, using the deterministic discrete event simulation method. At every decision instant in simulated time, the resources, jobs (lots), and supporting information are considered by taking a snap shot of the shop floor and support systems prior to each scheduling run (Sivakumar 2001). The near term scheduling is achieved by considering the system progress in a deterministic manner up to the next stochastic event or the elapse of a predetermined time. In this paper, our focus is on the selection of a suitable lot on machine m at decision instant k in simulation clock, keeping the interest on the demand of multiple contradicting objectives.

4 PARETO OPTIMAL SOLUTION

The subject of multiobjective optimization is widely researched and published (Goicoechea et al. 1982, Sawaragi et al. 1985, Steuer 1986, Tabucanon 1988, Yu 1985, Zeleny 1982). A multiobjective optimization problem can be denoted as:

$$Min\ f(x) \underline{\underline{\Delta}} (f^1(x), f^2(x), ..., f^P(x)),\ \text{for}\ x \in X,$$

where each $f^j(x)$, j =1…P, is a scalar objective function. An optimal solution in the classic sense is one which attains the minimum value of all the objectives simultaneously. The solution x* is optimal to the problem defined if and only if $x^* \in S$ and

$$f^j(x^*) \leq f^j(x)\ \text{for all j and for all}\ x \in S,$$

where S is the feasible region. In general, there may not be an particular optimal solution to a multiobjective problem, as one objective function gains only at the deterioration of the other objectives, due to their conflicting nature. In other words, optimality is an illusion when the objectives are conflicting. Therefore, one must be satisfied with obtaining the Pareto optimal solutions. A Pareto optimal solution is one in which no decrease can be obtained in any of the objectives without causing a simultaneous increase in at least one of the other objectives. A Pareto optimal solution is also called as efficient, non-dominated, or non-inferior solution (Tabucanon 1988). The solution x* is efficient to the problem defined if and only if there does not exist any $x \in S$ such that $f^j(x) \leq f^j(x^*)$ for all j and $f^j(x) < f^j(x^*)$ for at least one j.

Suppose that there are two objective functions, $f^1(x)$, and $f^2(x)$, where $x \in X$. A single objective problem can be formulated as $Z \underline{\underline{\Delta}} (f^1(x), f^2(x))$. We can calculate Z^1, where $Z^1 \underline{\underline{\Delta}} (f^1(x^1), f^2(x^1))$, for a point $x^1 \in X$, and the value of Z^1 can be plotted on the Cartesian coordinates as shown in Figure 3. Suppose our goal is to minimize both f^1 and f^2. Point Z^1 resulting from x^1 is certainly not the choice as point Z^2 is better than point Z^1 in terms of both f^1 and f^2.

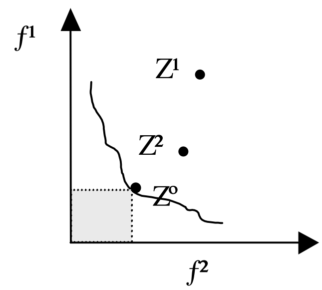

Figure 3: Pareto Optimization

Pareto Optimal or efficient solutions are defined as the boundary line of solutions that are better solutions than the others in the operating region. An operating solution $x^0 \in X$ is a Pareto optimal or efficient solution if no other operating point $x \in X$ exists such that $f(x) \leq f(x^0)$ and this implies that we will not find another operating solution x^a such that $f^1(x^a) \leq f^1(x^0)$ and $f^2(x^a) \leq f^2(x^0)$. In this example, point Z^o can be considered as a Pareto optimal point since there are no solution in the shaded area. In general, real life problems have more than two objective functions, usually with tradeoff between these. The primary objective of our research is therefore to identify solutions that are always on the Pareto optimal boundary and use various approaches on conflicting objectives to locate the desired solutions at every job selection event in simulated horizon.

5 APPROACHES FOR MULTIOBJECTIVE OPTIMIZATION

The multiobjective optimization problems are generally solved by combining the multiple objectives into one scalar objective, whose solution is a Pareto optimal point for the original MOP (Multiobjective Optimization Problem). Most of these combinations are either in a linear fashion or in form of the distance derivatives (Goicoechea et al. 1982, Gupta and Sivakumar 2002, Tabucanon 1988, Yu 1985, Zeleny 1982). Some of the prominent techniques in this relation are Weighted Aggregation, Global Criteria, Minimum Deviation, and Compromise Programming, which are described in the following sections.

5.1 Weighted Aggregation Method

A standard technique in multiobjective optimization is to minimize a positively weighted convex sum of the objectives. It is easy to prove that the minimizer of this combined function is Pareto optimal (Steuer 1986). But, the problem is up to the user to choose appropriate weights. Until recently, considerations of computational expense forced users to restrict themselves to performing only one such minimization, considering just one set of weights chosen with care. Newer, more ambitious approaches aim to minimize convex sums of the objectives for various settings of the convex weights, therefore generating various points in the Pareto set. Though computationally more expensive, this approach gives an idea of the shape of the Pareto surface and provides the user with more information about the trade-off among the various objectives. In this, weighted aggregation approach, different objectives are weighted and summed up to one single objective. The problem then becomes as:

$$Min.F = \sum_{j=1}^{P} w_j f^{j}(x),$$

where w_j are non-negative weights with $\sum w_j = 1$. By varying these weights, the whole Pareto surface can be found out as each Pareto optimal solution point on a convex surface correspond to a set of w_j.

This method is the simplest possible approach to solve the multi objective problem, but from application point of view, the user may be having only an intuition of the importance of one objective over the other, without having any knowledge of an exact set of weights for their various objectives, as it is very tough to establish a relationship between these weights and the real outcome in terms of objective functions values. To deal with this complexity, the researchers came up with the idea of finding a Pareto boundary by assigning varying weights to the objectives. Here, the difficulty is that the user has the full set of Pareto optimal points, defining a Pareto boundary, but out of this whole set which one to choose for application. Another problem is that of dimensional inconsistency among various objectives. So, often it is observed that due to the different units of the objectives, one objective functional value seem to be becoming dominant in the overall weighted aggregation of objective functions, even in various combinations of the weights. As a result, this approach becomes misleading, always deciding in favor of a particular objective, unless normalization is performed. One of the simple normalization technique to overcome this problem is to divide the each objective function by its maximum value and using the weighted ratio sum.

However, this method suffers from two more serious drawbacks (Das and Dennis 1997). First, the relationship between the vector of weights and the Pareto curve is such that a uniform spread of weight parameters rarely produces a uniform spread of points on the Pareto set. Often, all the points found are clustered in certain parts of the Pareto set with no solution in the interesting "middle part" of the set, thereby providing little insight into the shape of the trade-off curve. The second drawback is that non-convex parts of the Pareto set cannot be obtained by minimizing convex combinations of the objectives, though the existence of a non-convex part in a Pareto boundary is a very rarely occurring phenomenon.

5.2 Global Criterion Method

In this method, a global objective function is formed as the sum of derivations of the values of the individual objective functions from their respective singular objective optimum values as a ratio to that of the singular optima. Thus, from the original P objective functions, a single function is formulated and the problem becomes tantamount to solving a single objective optimization (Tabucanon 1988). The modified problem is:

$$Min.F = \sum_{j=1}^{P} \left[\frac{f^{j}(x^*) - f^{j}(x)}{f^{j}(x^*)} \right]^{r},$$

where $f^{j}(x^*)$ is optimum value of singular objective function j at its optima point x^*, $f^{j}(x)$ is the function value itself, and r is an integer valued exponent that serves to reflect the importance of the objectives. Since, the individual terms in the global objective function are expressed in ratios that are necessarily dimensionless, there is no need to worry regarding the problem of dimensional consistency among various objective functions. In addition, explicit information on the relative importance of the objectives is also not necessary to know.

However, the value of exponent r has to be defined by the user, which is generally set as greater than or equal to two in order to give more and more weight to the largest of deviations from the theoretical ideal solution (Tabucanon 1988). One positive thing about it is that as the value of r increases beyond a particular value, the solution set becomes consistent with further increase in the value of r. Therefore, the selection of r does not remain a complex problem in this case. One difficulty with this method arises when the individual optimum of an objective is very small or close to zero. In such case, this objective becomes dominant over all other objective functions as in the overall minimizing function, the deviation from this particular objective function is divided by a quantity approaching to zero, making the corresponding term extremely large. Then, the overall minimizing function would try to make this term as minimum as possible by making the deviation very small and thus resulting in the favor of this single particular objective function.

5.3 Minimum Deviation Method

This method is also applicable when the Pareto optimal values of the objectives are known but their relative importance is not known. It aims at finding the best compromise solution, which minimizes the sum of individual objective's fractional deviation. The fractional deviation of an objective refers to a ratio between the deviation of a value of that objective from its individual solution and its maximum deviation. The maximum deviation of an objective is obtained from the difference between its individual optimal solution and its least desirable solution, which corresponds to the individual optimal solution of one of the other objectives (Tabucanon 1988, Steuer 1986).

5.3.1 Developing A Payoff Table

For each objective function, its optimal value is first determined and values of other objective functions are calculated corresponding to this individual optimum. After computing it for all the objectives, a payoff table is formed as shown in Table 2. Column j correspond to the solution vector $x^{j}*$, which optimizes the jth objective, $f^{j}(x)$. f^{ij} is the corresponding value taken by the objective $f^{j}(x)$ when $f^{j}(x)$ reaches its individual optimum value $f^{j}(x*)$. The individual optimum value of each objective function is on the diagonal elements of the payoff table (i.e., when $i=j$).

Table 2: Payoff Table

	X^{1*}	X^{2*}	…	X^{j*}	…	X^{P*}
Z^1	$f^1(x^*)$	f^{12}	…	f^{1j}	…	f^{1P}
Z^2	f^{21}	$f^2(x^*)$	…	f^{2j}	…	f^{2P}
:	:	:	:	:	:	:
Z^i	f^{i1}	f^{i2}	…	f^{ij}	…	f^{iP}
:	:	:	:	:	:	:
Z^P	f^{P1}	f^{P2}	…	f^{Pj}	…	$f^P(x^*)$

Let $x*$ denote the ideal solution, which gives the vector of the optimum value of each objective function. Thus $F*(x*) = [f^1(x*), f^2(x*),\ldots, f^P(x*)]$ is the ideal objective vector. This vector can not be obtained unless all objectives are non-conflicting i.e., this ideal objective vector is only a hypothetical solution, which never exists in the practical real-life problems.

5.3.2 Computational Procedure

The best compromise solution is defined as the solution that will give the minimum of the sum of the fractional deviation of all objectives (Tabucanon 1988). The fractional deviation of each of the objectives is expressed as a fraction of its maximum deviation. Let $f^{j}(x^{o})$ be the least desirable value of $f^{j}(x)$. The minimum deviation problem is therefore formulated as:

$$Min.F = \sum_{j=1}^{P}\left[\frac{f^{j}(x^*) - f^{j}(x)}{f^{j}(x^*) - f^{j}(x^{o})}\right]$$

This formulation is justified by giving various reasons. Firstly, the objectives may be different in units of measurement. The fractional conversion will help in eliminating the effect of the dimension differences in computation. Secondly, in the event of any significant difference in magnitude of the objective function, the total deviation in absolute terms will be dominated by the objective which has a greater magnitude. The fractional term will help by normalizing the magnitude of each objective. Lastly, it helps to avoid the difficulty when the individual optimum of an objective is very small or close to zero.

5.4 Compromise Programming

The concept of compromise programming is similar to other distance based techniques (Goicoechea et al. 1982, Tabucanon 1988, Zeleny 1982). The method of compromise programming identifies solutions which are closest to the ideal solution (described in section 5.3.1) as determined by some measure of distance. The solutions identified as being closest to the ideal solution are called compromise solutions and constitute the compromise set.

In compromise programming , the point of interest is the comparison of distances of different efficient points $(f^{j}(x), j = 1,2,\ldots,P)$ from the ideal solution which is the point of reference. Since the objectives may be of different dimensions, so the distance measure needs to be corrected to make the individual objectives mutually commensurable. It is therefore necessary to use relative rather than absolute deviations. The individual relative deviations can be raised to any power $(r=1,2,\ldots,\infty)$ before these are summed, and also the weights w_j $(0<w_j<1$ and $\sum_j(w_j)=1)$ can be attached to the different relative deviations (Tabucanon 1988). For a multiobjective problem, with ideal point $F*(x*) = [f^1(x*), f^2(x*),\ldots, f^P(x*)]$, the overall multiobjective minimizing objective function can be expressed as follows:

$$Min\ d_r = \left[\sum_{j=1}^{P}\left[w_j \cdot \frac{f^{j}(x^*) - f^{j}(x)}{f^{j}(x^*)}\right]^{r}\right]^{1/r}$$

Operationally, three points of the compromise set are usually calculated, that is, those corresponding to $r=1$, 2, and ∞ (Goicoechea et al. 1982). It should be noted that when $r=1$ and equal weights w_j, the compromise programming technique is equivalent to the global criterion discussed in section 5.2. When $r=2$, the equation becomes simply the distance between two points in P dimension space, where each relative deviation is weighted in proportion to its magnitude. As r becomes larger and larger, the largest deviation receives more and more weight. For r approaching ∞, the distance measure reduces to the following expression:

$$d_{r\to\infty} = \underset{j}{Max}\left(\left|w_j \cdot \frac{f^j(x^*) - f^j(x)}{f^j(x^*)}\right|\right)$$

This is because the relative contribution of the largest relative deviation when raised to a large exponent would be extremely larger than all the rest combined, and thus will dominate the distance determination. Therefore, the choice of r reflects the user's concern with respect to the maximal deviation.

Introduction of w_j allows the expression of the user's intuition concerning the relative importance of the various objectives. Thus a double-weighting scheme exists. The parameter r reflects the importance of the maximal deviation and the parameter w_j reflects the relative importance of the j^{th} objective. From application point of view, both these parameters have to be decided by the user.

6 APPLICATION

In semiconductor manufacturing, more than one objective such as cycle time, machine utilization and due date accuracy are considered as the performance criteria. The objective is to provide feasible approaches for the selection of a suitable lot at each decision instant in simulation clock, to optimize amongst the contradicting needs of delivery accuracy, machine utilization and cycle time (Sivakumar et al. 2001). High machine utilization can be considered equivalent to low machine idle time, especially the setup change time.

6.1 Due Date Priority D_{ik}

Due date priority is computed based on the slack time of each job at the lot scheduling instance k with D_{ik} as positive or negative value as described in Figure 4 and therefore defined as:

$$D_{ik} = d_i - k - RP_{ik},$$

where RP_{ik} is the remaining total pure process time of lot i at time instance k and d_i is the due date and time of lot i at the start of simulation (Sivakumar et al. 2001).

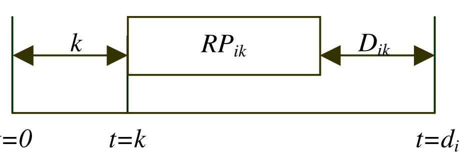

Figure 4: Due Date Priority (D_{ik})

6.2 Relative Cost Factor of a Setup SC_{imk}

Relative cost factor of a setup, SC_{imk} is defined as:

$$SC_{imk} = S_{imk}/R_{imk}, \quad \text{with } 0 < R_{imk} \le R_x,$$

where R_{imk} is the potential run time of available work in process that can make use of the new setup on machine m after processing the operation step on lot i. R_x , is the run time of remaining customer demand. S_{imk} is the sequence dependent setup time on machine m from its previous setup to process current lot i at an instance k. If a job can be processed without any setup i.e., if setup time = 0 then the relative setup cost factor is also zero.

Value of a setup also depends upon the number of other machines in the machine family that has identical setup as lot i at the particular instance k. We define the integer variable δ_{imk} to denote the quantity of equipment available (but busy at instance k) of the identical setup as required by lot i at the instance k. Let the integer variable $\delta_{imk} \in [0,1]$ equal to one if machine m is identical setup to lot i and zero otherwise. Therefore the number of machines, Q_k that have identical setup to that of the setup of lot i at the instance k can be expressed for each instance of k as $\sum_{m=1}^{M}\delta_{imk} = Q_k$, where $\delta_{imk} \ge 1$, as at least the machine at decision point satisfies the criteria. Potential run time, R_{imk}, is computed by $R_{imk} = \sum P_{imk}/Q_k$, where $\sum P_{imk}$ is the cumulative process time of all jobs available at instance k to satisfy customer demand that are capable of being processed on machine m using the new setup of lot i.

6.3 Cycle time priority $T'_{iy_n k}$

Cycle time priority is computed based on a modified cycle time, which is the sum of actual cycle time at simulated instance k and remaining process time so that we get a relative measure of cycle time at completion. We define the remaining process time of lot i at simulated instance k as P_{iyk} and actual cycle time i.e., the duration from arrival time of lot i at operation y to time k as T_{iyk} . If the arrival time of lot i at operation y is represented by b_{iy}, then the actual cycle time will be defined as: $T_{iyk} = k - b_{iy}$.

Therefore, the modified cycle time is $P_{iyk} + T_{iyk}$ and the cycle time priority, T'_{iyk}, is defined as (Sivakumar 2001):

$$T'_{iyk} = \left[\frac{1}{\left(P_{iyk} + T_{iyk}\right)^2}\right],$$

The square value of the modified cycle time provides a polynomial increase in priority for cycle time as it gets aged. Lower the value of T'_{iyk} denotes a higher priority.

6.4 Pareto Optimization

Each of the above factors with a lower value denotes a higher priority. The objective functions therefore are:

$$\textit{Minimize } D_{ik},$$
$$\textit{Minimize } SC_{imk}, \textit{ and}$$
$$\textit{Minimize } T'_{iyk}.$$

A programming formulation can therefore be established to represent the scheduling problem using the weighted aggregation method, global criterion approach, minimum deviation technique and compromise programming. Thus, it has been formulated that the scheduling problem in the semiconductor manufacturing can be solved by the proposed approaches, with out even formulating the problem as NP-hard.

7 AN ILLUSTRATION

The example in Table 3 shows the snap shot detail of the lots, in a typical lot selection scenario on a particular machine. There are 10 lots, belonging to two different families F1 and F2, in the queue during this selection instance. The due-date priority, cycle time priority and relative set-up cost factors are generated using fictitious data in the proper range based on the factory data of previous work (Sivakumar 2001). The testing of this methodology is done manually in the excel-sheet environment. The simulation clock is advanced manually on ten consecutive decision instances. In this example, the sequence dependent setup time, S_{imk}, consists of two things: one, the normal setup of lots, which is taken as 30, 60, 90, or 120 min. respectively and second, an additional setup time of 60 min, if family change from F1 to F2, and of 120 min if family changes from F2 to F1. The potential run time, R_{imk}, is the sum of the processing times of all those lots which can be loaded with same setup. The remaining processing time, P_{iyk}, is taken between 40 to 240 min., arrival time of lot i at operation y, b_{iy}, as less than 500 min., due date and time, d_i, between 4 to 7 days, remaining total pure process time RP_{ik} as between 1 to 3 days, etc. The initial instant is considered at k=500 min with already existing setup of family F1.

Then, the due date priority, $D_{ik,}$ the relative set-up cost actor, SC_{imk}, and the cycle time priority, T'_{iyk}, are calculated from the data, shown in Table 3. Further, all the four methods are applied to select a lot for loading, with equal weights and power three wherever necessary. Table 4 shows the lots, their due-date priority D_{ik}, cycle time priority, T'_{iyk}, relative set-up cost factor, SC_{imk}, and their joint multiobjective function value in various approaches, discussed in section 5, at this initial lot selection instant.

Table 3: Snap Shot Detail of the Lots.

Initial INSTANT 1:			at k=500min.=0.35 days			With F1 family	
Lot no.	d_i	RP_{ik}	SetupT	S_{imk}	R_{imk}	b_{iy}	P_{iyk}
	4-7 days	1-3 days	0-120			(0-500)	(40-240)
J1,F1	5.9	2.4	90	90	995	220	62
J2,F1	4.9	1.7	60	60	995	122	169
J3,F1	5.3	2.1	30	30	995	45	211
J4,F1	4.1	2.8	120	120	995	318	124
J5,F1	6.5	1.3	30	30	995	149	235
J1,F2	6.9	1.1	120	180	455	85	185
J2,F2	5.0	1.4	90	150	455	134	239
J3,F2	6.2	2.6	30	90	455	187	97
J4,F2	4.4	1.9	60	120	455	256	220
J5,F2	5.4	2.9	60	120	455	423	138

Table 4: Individual Function Priorities and Multiobjective Values in Different Approaches.

Lot no.	D_{ik}	SC_{imk}	T'_{iyk}	Wtd.Av.	Glo.Cri.	Min.Dev.	Com.Prog.
				equal wts.	r=3		eq. wts., r=3
J1,F1	3.14	0.1122	9E-06	0.506	42.72	1.258	1.165
J2,F1	2.88	0.0748	3E-06	0.349	10.14	0.710	0.721
J3,F1	2.77	0.0374	2E-06	0.265	7.64	0.407	0.656
J4,F1	0.93	0.1496	1E-05	0.465	79.25	1.105	1.432
J5,F1	4.85	0.0374	3E-06	0.402	73.92	0.901	1.399
J1,F2	5.45	0.2048	3E-06	0.709	202.91	2.027	1.959
J2,F2	3.25	0.1706	3E-06	0.519	60.49	1.334	1.308
J3,F2	3.25	0.1024	6E-06	0.457	24.95	1.092	0.974
J4,F2	2.15	0.1365	5E-06	0.425	22.01	0.985	0.934
J5,F2	2.15	0.1365	2E-05	0.687	657.17	1.862	2.898
Min.=	0.93	0.0374	2E-06	0.265	7.64	0.407	0.656
Lot J3,F1 will be selected at k = 500 min.							

From the values of these objective functions, it is very clear that the weighted average method will mislead unless normalization of values is performed, due to dimensional inconsistency. So, while computing, the simple normalization is performed by taking the ratio of each function value and its maximum available value in the given lots. From theses values, all four methods result in the choice of same lot J3,F1 at this particular instant. After selecting a lot at each decision instant, the simulation clock is forwarded by the added time of setup and processing. For next instant consideration, the potential run time, R_{imk}, is reduced by the processing time of last selected lot and new setup times are computed based on the change in the family of lots. Thus, the manual simulation experiments are done for all the four approaches to find out the sequence of the lots. Table 5 shows the sequence of these lots for processing on the particular machine with all these four methods and with common heuristics used as thumb rules such as FCFS (First-Cum-First-Serve), SPT (Shortest Processing Time) and EDD (Earliest Due Date).

8 DISCUSSION

Based on the sequence of lots under different methods as shown in Table 5, various performance parameters such as completion time of last lot, make-span time, average cycle time of all the lots and their standard deviation, average

Table 5: Sequence of Lots under Different Methods

Wtd.Av.	Glo.Cri., Com.Prog.	Min.Dev.	FCFS	SPT	EDD
J3,F1	J3,F1	J3,F1	J3,F1	J1,F1	J4,F1
J2,F1	J4,F2	J2,F1	J1,F2	J3,F2	J4,F2
J4,F2	J5,F2	J4,F2	J2,F1	J4,F1	J2,F1
J5,F2	J2,F1	J2,F2	J2,F2	J5,F2	J2,F2
J3,F2	J4,F1	J3,F2	J5,F1	J2,F1	J3,F1
J2,F2	J3,F2	J5,F2	J3,F2	J1,F2	J5,F2
J4,F1	J2,F2	J5,F1	J1,F1	J3,F1	J1,F1
J5,F1	J1,F1	J4,F1	J4,F2	J4,F2	J3,F2
J1,F1	J5,F1	J1,F2	J4,F1	J5,F1	J5,F1
J1,F2	J1,F2	J1,F1	J5,F2	J2,F2	J1,F2

Table 6: Comparison of Various Performance Parameters

	Wtd. Av.	Glo. Cri., Com. Pro.	Min. Dev.	FCFS	SPT	EDD
Completion Time (min) =	3110	3290	3230	3650	3650	3650
Make-span time (min) =	3065	3245	3185	3605	3605	3605
Average CT (min) =	1697	1786	1758	2061	1852	2032
Std. Dev. of CT (min) =	795.6	854.2	807.6	861.8	1079.7	995.9
Average Waiting time (min) =	1436	1507	1485	1746	1537	1717
Std.Dev. of Waiting Time (min) =	777.7	829.3	785.1	869.5	1014.2	988.6
M/C Utilization =	0.6437	0.6022	0.6154	0.5333	0.5333	0.5333

waiting time in the queue and their standard deviation, and machine utilization are computed for each method and are shown in Table 6. From these results, it is very clear that these four methods which take in to consideration the multiple objectives are much better than heuristics which consider only one objective at a time, in relation to almost all the performance parameters for this particular example. There is quite significant improvement in the performance of the primary objectives (machine utilization, cycle time and delivery within due time). Especially, in this particular case, the equal weighted aggregation method gives the best results in all the performance parameters, among all the shown methods. To test this methodology on the real factory data in a complex job shop environment, with the help of a simulation engine is in the consideration of the future work of this ongoing research.

9 CONCLUSION AND FUTURE WORK

This paper presented the concepts, development and application methodology of simulation based multiobjective schedule optimization in semiconductor manufacturing. The methodology reported can be applied easily to the complex job shop scheduling problems such as in semiconductor manufacturing and significant benefits can be achieved in terms of cycle time distribution, on-time delivery and utilization of the shop. Pareto optimal solutions can be consistently achieved in dynamic manufacturing environment, using the proposed approaches. The process of testing these approaches on a simulation engine constitutes the currently going on research in further direction.

Future work includes experimenting the proposed methodology on real factory data using an on-line simulation engine, comparing it with the common heuristic methods such as First-Come-First-Serve, Shortest Processing Time, Earliest Due Date etc, which are generally used in the industry as thumb rules. However, the confirmation of the point, that selection of a Pareto optimal lot at each decision instant in simulation clock will provide an overall Pareto optimal solution of the scheduling problem, needs further research. Further research issues also include the development of the methodology for operating the shop in a user controlled trade off between the objectives i.e., the selection of a lot in such a way that it will give the overall schedule according to the defined emphasis on the particular objectives.

REFERENCES

Das, I., and J. Dennis. 1997, A closer look at the drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1): 63-69.

Davis, W.J. 1998, On-line Simulation: Need and Evolving Research Requirements. In *Handbook of Simulation*, eds. J. Banks, John Wiley and Sons, Inc., New York: 465-516.

Goicoechea, A., D.R. Hansen, and L. Duckstein. 1982, *Multiobjective decision analysis with engineering and business applications*, John Wiley and Sons, New York.

Gupta, A.K., and A.I. Sivakumar. 2002, Approaches to Multiobjective scheduling optimization in semiconductor back-end. In *Proceedings of International Conference on Modeling and Analysis of Semiconductor Manufacturing (MASM) April 10-12, 2002*, Tempe, Arizona: 223-228.

Hopp, W.J., and M.L. Spearman. 1996, *Factory Physics: foundations of manufacturing systems*, Irwin, Times Mirror, USA.

Kiran, A.S. 1998, Simulation and scheduling. In *Handbook of Simulation*, ed. J.Banks, John Wiley and Sons, Inc., New York: 677-717.

Koh, K.-H., R. de Souza, and N.C. Ho. 1996, Database driven simulation / simulation-based scheduling of a job-shop. *Simulation Practice and Theory,* 1996, Vol. 4: 31-45.

Mazziotti, B.W., and R.E. Horne, Jr. 1997, Creating a flexible, simulation-based finite scheduling tools. In *Proceedings of the 1997 Winter Simulation Conference*. ed. S. Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson, IEEE, Piscataway, New Jersey: 853-858.

Morito, S., and K.H. Lee. 1997, Efficient simulation / optimization of dispatching priority with "fake" processing time. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson, IEEE, Piscataway, New Jersey: 872-879.

Pinedo, M. 1995, *Scheduling: theory, algorithms and systems*. Prentice Hall, Englewood Cliffs.

Rippenhagen, C., and S. Krishnaswamy. 1998, Implementing the theory of constraints philosophy in highly reentrant systems. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D.J. Medeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan, IEEE, Piscataway, New Jersey: 993-996.

Sawaragi, Y., H. Nakayama, and T. Tanino. 1985, *Theory of multiobjective optimization*, Academic Press Inc., Orlando, Florida.

Sivakumar, A.I. 2001. Multiobjective dynamic scheduling using discrete event simulation, *International Journal of Computer Integrated Manufacturing*, 2001, Vol 14, No. 2: 154-167.

Sivakumar, A.I., A.K. Gupta, and C.S. Chong. 2001, Multiobjective Pareto optimal dynamic scheduling of semiconductor backend using on-line discrete event simulation, under review in *IIE Transactions*.

Sivakumar, A.I. 1999. Optimization of Cycle time and Utilization in semiconductor test manufacturing using simulation based, on-line near-real-time Scheduling System, In *Proceedings of the 1999 Winter Simulation conference*, eds. P.A. Farington, H.B. Nembhard, D.T. Sturrok, and G.W. Evans, IEEE, Piscataway, New Jersey: 727-735.

Steuer, R. E. 1986, *Multiple criteria optimization: theory, computation, and application*. Krieger Publishing Company, Malabar, Florida.

Tabucanon, M. T. 1988, *Multiple criteria decision making in industry*. Elsevier Science Publishers B.V., Amsterdam, The Netherlands.

Watt, D.G. 1998, Integrating simulation based scheduling with MES in a semiconductor FAB. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D.J. Medeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan, IEEE, Piscataway, New Jersey: 1713-1715.

Yu, P.-L 1985*, Multiple-criteria decision making: concepts, techniques, and extensions*, Plenum press, New York.

Zeleny, M. 1982, *Multiple criteria decision making*, McGraw-Hill Book Company, New York.

AUTHOR BIOGRAPHIES

AMIT KUMAR GUPTA is a Ph.D. research student in the School of Mechanical and Production Engineering at Nanyang Technological University, Singapore. He obtained his bachelor degree in Mechanical Engineering (1st class Hons.) from the Indian Institute of Technology (IIT), Delhi. He is a research scholar in the field of multiobjective scheduling optimization in the dynamic environment of semiconductor manufacturing**.** His email address is <pa7032927@ntu.edu.sg>.

APPA IYER SIVAKUMAR (Senior member IIE) is an Associate Professor in the School of Mechanical and Production Engineering at Nanyang Technological University (NTU), Singapore and a Fellow of Singapore Massachusetts Institute of Technology(MIT) Alliance (SMA). Prior to this he was at Gintic Institute of Manufacturing Technology, Singapore. His research interests are in the area of simulation based optimization of manufacturing performance, supply chain, and dynamic schedule optimization. Prior to joining Gintic in 1993, he held various management positions including technical manager and project manager for nine years at Lucas Systems and Engineering and Lucas Automotive, UK. He received a Bachelors of Engineering from University of Bradford, UK and a Ph.D. in Manufacturing Systems Engineering from University of Bradford, UK. He has been the technical chair and coedited the proceedings of the 3rd and 4th International Conference on Computer Integrated Manufacturing (ICCIM '95 and ICCIM'97), Singapore. His email address is <msiva@ntu.edu.sg>.