J Sched (2008) 11: 29–47
DOI 10.1007/s10951-007-0049-1

A multi-criteria approach for scheduling semiconductor wafer fabrication facilities

Michele E. Pfund · Hari Balasubramanian · John W. Fowler · Scott J. Mason · Oliver Rose

Published online: 27 November 2007

Abstract In this research, we model a semiconductor wafer fabrication process as a complex job shop, and adapt a Modified Shifting Bottleneck Heuristic (MSBH) to facilitate the multi-criteria optimization of makespan, cycle time, and total weighted tardiness using a desirability function. The desirability function is implemented at two different levels of the MSBH: the subproblem solution procedure level (SSP level) and the machine criticality measure level (MCM level). In addition, we suggest two different methods of choosing the critical toolgroup at the MCM level: (1) the Local MCM approach, which chooses the critical toolgroup based on local desirability values from the SSP level and (2) the Global MCM approach, which chooses the critical toolgroup based on its impact on the desirability of the entire disjunctive graph. Results demonstrate the desirability-based approaches' ability to simultaneously minimize all three objectives.

M.E. Pfund
Supply Chain Management Department, Arizona State University, Tempe, AZ 85287, USA

H. Balasubramanian
Department of Health Sciences Research, Mayo Clinic, Rochester, MN 55905, USA

J.W. Fowler (✉)
Department of Industrial Engineering, Arizona State University, PO Box 875906 Tempe, AZ 85287-5906, USA
e-mail: john.fowler@asu.edu

S.J. Mason
Department of Industrial Engineering, University of Arkansas, 4207 Bell Engineering Center, Fayetteville, AR 72701, USA

O. Rose
Institute for Applied Computer Science, Technical University of Dresden, 01062 Dresden, Germany

Keywords Multicriteria · Shifting bottleneck · Complex job shop

1 Introduction

The Factory Operations portion of the 2005 International Technology Roadmap for Semiconductors (SIA 2005) indicates that there is increasing pressure on semiconductor manufacturers to maximize throughput, reduce cycle times and improve on-time delivery (OTD) of products to customers. This section of the ITRS also contains a list of potential solutions to the cost per function (e.g., transistor) and cycle time requirements. The potential solutions are classified into planning decision support tools at the strategic level and tools for running the factory at the tactical or execution level. The ITRS identifies real-time scheduling as one of the execution-level potential solutions. In this paper, we discuss the development of a new approach for scheduling semiconductor wafer fabrication facilities ("wafer fabs") that attempts to optimize an aggregation function that combines throughput (equivalent to makespan ($C_{\max}$)), average cycle time (CT, which is similar to the sum of completion times), and OTD (minimize total weighted tardiness (TWT)).

In a typical wafer fab, there often are dozens of process flows. Process flows are routes that the wafer lots have to follow in the factory. Each process flow contains 200–500 processing steps and more than 100 machines. These machines are expensive, ranging in price from $50,000 to over $14 million per tool. Frequently, groups of identical machines process lots in parallel, thereby forming a *toolgroup*. The economic necessity to reduce capital spending dictates that such expensive machines be shared by all lots requiring the particular processing operation(s) provided by the machine, even though they may be at different stages of their manufacturing process flow. In fact, a given part may visit a toolgroup many times as part of its process flow; this is called re-entrant flow. This results in a manufacturing environment that is different in several ways from both traditional flow shops as well as job shops. The main consequence of this re-entrant flow is that wafers at different stages in their manufacturing cycle have to compete with each other for the same machines. The manner in which this competition is resolved has a clear impact on wafer fab objectives.

Furthermore, the nature and duration of the various operations in a wafer fab process flow differ significantly. Some operations require 30 minutes or less to process a *lot* of 25 wafers, while others may require over twelve hours. Many of these long operations involve batch processing of lots and it is not uncommon for one-third of all wafer fab operations to involve batch processing. Batch processing machines tend to off-load multiple lots (1 to 6) onto tools that are capable of processing only one lot at a time. This leads to the formation of long queues in front of these serial (non-batch) tools and ultimately a non-linear flow of products in the factory. The probabilistic occurrence of unplanned tool failures results in a great deal of variability inherent in the time a lot spends in process in the wafer fab. This variability prevents accurate prediction of production cycle times, resulting in longer lead-time commitments. There are some wafer fab machines, such as ion implanters, that require significant sequence-dependent setups. If not scheduled appropriately, these tools can become wafer fab bottlenecks. In order to understand the scheduling approaches currently being used in the semiconductor industry, a survey instrument was created and sent to each of the Semiconductor Research Corporation and International Sematech member companies. The survey was designed to ask specific questions regarding the types of scheduling methodologies currently implemented, the limitations of these methodologies, and the needs for future generation scheduling systems. In total, 16 respondents from 14 companies participated in the survey, representing wafer fabs from Europe, Asia, and North America.

Survey results indicate that many dispatching systems are in place within wafer fabs, most of which have been installed for more than five years (Fowler and Pfund 2001). These systems are considered "satisfactory" in that benefits are being received, but the majority of survey respondents believe that more benefits are possible. Specifically, respondents indicate that better scheduling/dispatching rules, test environments, and reporting tools are needed. The survey asked for the top three objectives used in wafer fabs today. The top three responses were cycle time, factory throughput, and on-time delivery. Maximizing factory throughput is similar to minimizing makespan, while TWT can be thought of as a surrogate measure for OTD. Compared to the results of the 1994 Sematech survey, cycle time and OTD have gained significant importance in wafer fabs (Neacy et al. 1994). The primary goal of this research effort is to develop a solution approach that provides good performance for makespan, average cycle time, and TWT for semiconductor wafer fabs. We use the well known Shifting Bottleneck approach as the basis for our approach.

2 Related work

2.1 The shifting bottleneck heuristic

The disjunctive graph formulation and the Shifting Bottleneck procedure to solve the job shop scheduling problem to minimize the makespan ($Jm||C_{\max}$ in the notation $\alpha|\beta|\gamma$ of Graham et al. 1979) was first proposed by Adams et al. (1988). Since then research has been focused on the algorithmic improvement of the procedure (papers that focus on this include Dauzère-Pérès and Lasserre 1993, Balas et al. 1995, and Balas and Vazacopoulos 1998) and also on extending the Shifting Bottleneck procedure to more complicated objectives and augmenting the job shop environment with features that commonly occur in practice. Ovacik and Uzsoy (1992) use an adapted Shifting Bottleneck procedure for the scheduling of semiconductor testing operations. They included sequence dependent setups and used the maximum lateness as an objective ($Jm|r_j, s_{jk}|L_{\max}$). Ivens and Lambrecht (1996) and Schutten (1998) discuss the extension of the disjunctive graph formulation to accommodate practical features such as due-dates, release dates, setup times, transportation times, parallel machines, and beginning inventory.

Pinedo and Singer (1999) develop a disjunctive graph formulation and used the Shifting Bottleneck procedure to minimize total weighted tardiness ($Jm|r_j|\sum w_j T_j$). Wafer fabs are modeled as complex job shops by Mason et al. (2002), who extend the classical job shop work of Pinedo and Singer (1999) to develop a disjunctive graph formulation and modified Shifting Bottleneck heuristic (MSBH) for the wafer fab scheduling problem which we represent as $FJc|r_j, s_{jk}, p\text{-}batch, recrc|\sum w_j T_j$. Mason et al. (2002) account for toolgroups consisting of multiple identical machines in a given work center which perform the same function (which is why the environment is a flexible job shop represented as *FJc*), batch processing tools (*p-batch*), different arrival times of job (r_j), sequence-dependent setups (s_{jk}), and recirculating product flow (*recrc*); all of these are key features that characterize manufacturing in wafer fabs.

2.2 Multicriteria scheduling

Multicriteria scheduling research arose from the need to address real world scheduling problems, which seldom have a single objective function. A schedule that is good for one objective function may in fact be quite poor for another. Decision makers must carefully evaluate the trade-offs involved in considering several different criteria in practical scheduling applications.

Multicriteria problems can be considered in many ways and therefore it is important to point out the nature of optimization being performed. Consider, for example, a bicriteria scheduling problem with objectives of minimizing γ_1 and γ_2. If X is a feasible schedule, then let $X(\gamma_1)$ and $X(\gamma_2)$ be the γ_1 and γ_2 objective function values of X. While our discussion is on bicriteria problems, the ideas are easily extended to problems with more criteria.

Sometimes the decision-maker has a priori information regarding the nature of optimization to be performed. For instance, criteria γ_1 may be of primary importance and γ_2 of secondary interest. In other words, a solution best in γ_2 may be desired among all solutions that are best in γ_1. This is called hierarchical or lexicographic optimization. In other cases, the decision-maker may have a composite linear function of the form $F_l(\gamma_1, \gamma_2) = \alpha\gamma_1 + (1-\alpha)\gamma_2$, $0 \le \alpha \le 1$ in mind that needs to be minimized. The weighted sum translates multiple objectives into a single objective value for a proposed schedule. In this way, alternative schedules can be compared easily using only a single objective or fitness value. Some difficulty can be experienced in setting the weighting factors in practice, primarily due to the dimensionality of the objective function criteria in practice. Care must be taken to ensure the scale of γ_1 (e.g., $\sum w_j C_j$) does not dominate γ_2 (e.g., $C_{\max}$). Recently, Kacem et al. (2002) introduced a homogenization approach for dealing with this potential scale difference. Cochran et al. (2003) discuss several schemes for weighting the objectives.

A more complex problem is to generate the set of Pareto optimal or non-dominated points for the decision-maker to choose from. In a bicriteria context, a schedule X is called Pareto optimal or non-dominated if there exists no other feasible schedule X'' such that $X''(\gamma_1) \le X(\gamma_1)$ and $X''(\gamma_2) \le X(\gamma_2)$ where at least one of the inequalities is strict. The decision maker can now choose from this set of non-dominated solutions, the schedule that is most preferred. This approach is called the a posteriori approach and is generally the most difficult.

For comprehensive surveys on multicriteria scheduling we refer the reader to Foote et al. (1988), Nagar et al. (1995), T'kindt and Billaut (2006) and for a more recent study to Hoogeveen (2005). Of other papers, we refer here only to the papers that involve the job shop environment. Very few researchers have dealt with multi-criteria scheduling in job shops. The complexity of the problem has a major role to play in this. Esquivel et al. (2002) and Kacem et al. (2002) investigate the generation of Pareto optimal schedules in classical and flexible job shops. Iima et al. (1999) and Itoh et al. (1993) consider the minimization a function that involves all the objectives under study in classical job shops. Balas et al. (1998) address the job shop scheduling problem with deadlines ($J|\tilde{d}| \in (C_{\max}, T_{\max})$ in the terminology of T'kindt and Billaut 2006). This can be viewed as a bicriteria problem involving minimax objectives. The $\in$-constraint approach involves generating the set of Pareto optimal solutions by solving a series of subproblems in which one criterion is optimized while not exceeding a certain prefixed value of the other criterion. Recently, Balas et al. (2005) consider the same problem but with the inclusion of sequence dependent setups ($J|\tilde{d}, s_{jk}| \in (C_{\max}, T_{\max})$).

In this paper, we consider multicriteria scheduling problem in which we combine makespan ($C_{\max}$), average cycle time ($\sum C_j$), and TWT into a single aggregation function. Our aggregation function, however, is different from the linear combination of objectives described earlier. We use, instead, the desirability function to aggregate the objectives. We assume the decision maker has a prefixed goal/target value and an upper/worst-case value for every criterion. We also assume the decision-maker, just as in the linear combination case, is aware of the priorities on the objectives. The complex job shop environment has only been studied for TWT (Mason et al. 2002). Here we present a solution methodology for the multicriteria optimization of makespan, average cycle time, and TWT in a complex job shop.

Before we proceed with describing our approach, we note that minimizing any of the three criteria considered individually in a complex job shop is strongly NP-hard (via reduction from the classical job-shop case), and therefore the multicriteria problem involving the three criteria is strongly NP-hard as well. Our approach, therefore, focuses on the development of a heuristic approach that is computationally feasible for the complex job shop environment. We also note that while approaches other than the Shifting Bottleneck heuristic (enumerative approaches such as branch and bound algorithms, and meta heuristics) have been used for classical job shops, these approaches are difficult to apply, given the additional complicating features of the complex job shop (Mason et al. 2005). We therefore choose to build on the MSBH of Mason et al. (2002) to extend it for the multicriteria problem.

3 Aggregation using the desirability function

The desirability function approach in optimizing multiple criteria of interest was originally suggested by Derringer and Suich (1980). The approach transforms each objective into a value between 0 and 1. Thus, each criterion is converted into an individual desirability function δ_i that varies over the range zero to one. If is outside the user's defined acceptable range, then $\delta_i = 0$. However, if y_i meets the goal, then $\delta_i = 1$.

In our research, all three objectives (makespan, cycle time, and TWT) are to be minimized. Let U be the maximum allowable value for the response and let G be the goal

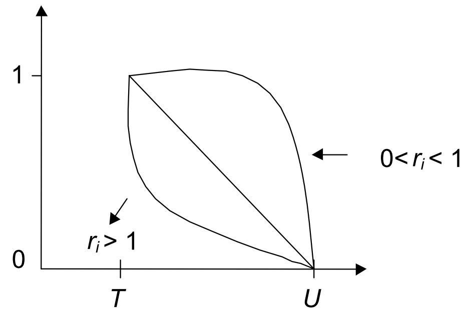

Fig. 1 The desirability function for measure i (Myers and Montgomery 1995)

value for y_i. When minimizing the response y_i, $\delta_i = 1$ if $y_i < G$. Further, if $G \le y_i \le U$,

$$\delta_i = \left(\frac{U - y_i}{U - G}\right)^{z_i}. \tag{1}$$

Otherwise, if , then $\delta_i = 0$. In (1), z_i is a real number known as the weight on the desirability function. When $z_i = 1$ for each objective i, the desirability function is linear (Fig. 1). Choosing $z_i > 1$ places more emphasis on being close to the goal value, while setting $0 < z_i < 1$ decreases importance on proximity to the goal value.

Once the individual desirabilities have been calculated, the combined desirability D, which is to be maximized, is computed as the geometric mean of the individual desirabilities:

$$D = (\delta_1 \delta_2 \cdots \delta_m)^{1/m}. \tag{2}$$

In (2), m is the total number of responses. For our research, $m = 3$, as we represent the desirabilities of makespan, cycle time and TWT as δ_{cmax}, δ_{ct}, and δ_{twt}. Each desirability value δ_i in our experiments will have goal value G_i and maximum (upper) limit U_i, $\forall i \in \{\text{C}\max, \text{CT}, \text{TWT}\}$.

It is important to link the properties of a solution that is optimal with respect to a desirability function, and its connection to the classical multicriteria idea of a Pareto optimal or non-dominated solution. From the definition and discussion of the desirability function, it is clear that the optimal solution depends upon the goals (G values) and upper targets (U values) assumed for each criterion. We have listed below two specific cases below with regard to this. While these cases highlight situations where optimality with respect to the desirability function and Pareto optimality may not always match with each other, they also show that these differences are more due to the framing of the problem with respect to upper and lower limits than any inherent issues in the desirability approach.

1. The way the desirability function is structured, any solutions that are over the pre-specified upper limit (for a minimization problem) for *any* of the criteria are assigned a value of 0. If the upper limit is fairly strict (aggressive), this could exclude non-dominated solutions that are at the ends of the efficient frontier. But in practice this simply means that these solutions not acceptable because they perform poorly for at least one of the criteria. Thus, a non-dominated solution may not be optimal from a desirability point of view as certain constraints on individual criteria have to be met.
2. Consider now the set of feasible solutions X with non-zero desirability values. Thus, for each solution in X, the criteria values will be strictly less than the upper limits. Let x^* be a dominated solution in X. Let ND_{x^*} be the set of non-dominated solutions that dominate x^*. Also, let $x^*(j)$ represent the value of criterion j for solution x^* and G_j represent the goal or target value for criterion j. Then, for a given set of desirability weights, x^* is an optimal solution with respect to the desirability function if: There exists no criterion j and no x^{**} in ND_{x^*} such that $x^{**}(j) <= G_j <= x^*(j)$.

   This situation arises because the desirability function does not distinguish between solutions that are below G_j for criterion j: it assigns them a value of 1 even though there are differences in criteria values. This reflects the idea that a "satisfaction level" for that criterion has been met. We note that while x^* is optimal for the desirability function (despite being dominated), there exists a solution(s) in ND_x which is (are) non-dominated and also (alternately) optimal. Also, as a special case if we were to set G_j to 0 (or to really low value), x^* would no longer be optimal for the desirability function. This leads to our decision for choice of G_j values in our methodology (see Sect. 4.1).

4 Methodology

We first provide a brief overview of the Shifting Bottleneck heuristic. Using the disjunctive graph representation, the SB procedure of Adams et al. (1988) decomposes the $J_m||C_{\max}$ problem into multiple instances of the $1|r_j|L_{\max}$ problem ('subproblems'). The subproblems are solved according to some specified 'subproblem solution procedure' (SSP) (a heuristic or an exact procedure depending on the computational requirements), and then evaluated in terms of a specified performance or 'machine criticality' measure. A computational study of machine criticality measures and sub-problem solution procedures is provided in Holtsclaw and Uzsoy (1996). The 'most critical' machine is then scheduled at each iteration of the procedure.

The MSBH procedure of Mason et al. (2002) builds on the SB procedure. It decomposes the complex job shop scheduling problem into individual toolgroup scheduling instances (each toolgroup scheduling instance is a sub-problem). The toolgroups (or the set of identical parallel machines) are then scheduled using a sub-problem solution procedure (SSP), and are then evaluated based on a machine criticality measure (MCM). The most critical toolgroup is then identified and scheduled in that iteration. Since our goal is to develop an approach that takes into account multiple criteria, we use the desirability function aggregation of the three objectives in our sub-problem solution procedure as well as our machine criticality measure. We now describe the details of our approach.

The MSBH of Mason et al. (2002) is used for analyzing the $FJc|r_j, s_{jk}$, *p-batch*, *recrc*$|\sum w_j T_j$ problem. We now present the steps of the MSBH:

1. Let M denote the set of all m toolgroups. Initially the set M_0 the set of toolgroups that have been sequenced or scheduled is empty.
2. Form and solve the subproblems for each toolgroup $i \in M \backslash M_o$ (SSP level)
3. Identify the critical or bottleneck toolgroup $k \in M \backslash M_o$ (MCM level)
4. Sequence tool group k using the subproblem solution from Step 2. Set $M_0 \cup \{k\}$.
5. Re-optimize the schedule for each toolgroup $m \in M_0$ considering the newly added disjunctive arcs for toolgroup k.
6. If $M = M_0$ stop. Otherwise, go to step 2.

At the SSP level (Step 2), each toolgroup is scheduled using some SSP, and an objective function evaluation is obtained each time a proposed toolgroup schedule is inserted into the underlying disjunctive graph. At the MCM level (Step 3), each toolgroup's objective function is then used to determine the most critical toolgroup in the current iteration of the MSBH. At both the SSP and MCM levels, we propose to use the desirability function of Derringer and Suich (1980), which seeks to optimize a single aggregation function that combines several different objective functions. We combine makespan, cycle time (average flow time), and TWT into a single desirability function, which is then optimized.

4.1 SSP level

We use the Apparent Tardiness Cost with Setups and Ready Times (ATCSR) heuristic of Gadkari (2003) to schedule jobs on each toolgroup at the SSP level. The ATCSR is a composite dispatching rule. It combines four different priority rules—Weighted Shortest Processing Time (WSPT), Least Slack, Shortest Setup, and Ready Time—into a single function. Except for the WSPT rule, all other rules are raised to an exponent so that they appropriately discount the index value for a job. In the ATCSR heuristic, an index $I_j(t,l)$ is calculated for every unscheduled job at time t as follows:

$$I_j(t,l) = \frac{w_j}{p_j} \exp\left(\frac{-\max(d_j - p_j - \max(r_j,t), 0)}{k_1 \bar{p}}\right) \times \exp\left(-\frac{s_{lj}}{k_2 \bar{s}}\right) \exp\left(-\frac{\max(r_j - t, 0)}{k_3 \bar{p}}\right), \quad (3)$$

where w_j is job j's weight or priority, and r_j is the ready time of job j with d_j and r_j set by finding the critical path of the disjunctive graph. We do not use the $d^k_{i,j}$ variables of Pinedo and Singer (1999) as they require increased computations and need to be calculated at each subproblem. This is especially true for the large models we consider in our experiments. Further, s_{lj} is the setup time incurred when changing from job l to job j, $\bar{p}$ is the average processing time of all remaining jobs, and $\bar{s}$ is the average setup time. The three scaling parameters k_1, k_2, and k_3 determine the relative importance of the exponential terms in relation to each other and to the WSPT term. The heuristic works as follows: we first identify the machine j that is available to process jobs the earliest (t denotes the time at which the machine is available). Next, we calculate the $I_j(t,l)$ value for all unscheduled jobs and schedule, at time t on machine j, the job that has the highest $I_j(t,l)$ value. The time on the machine is updated and the procedure is repeated.

ATCSR has traditionally been used only for the TWT objective (Lee and Pinedo 1997 and Gadkari et al. 2007). However, some of its components may be beneficial to the other objectives such as makespan and total completion time considered in this research. In Table 1 a check signifies a positive contribution by the ATCSR function's priority rule component to the corresponding objective function criterion.

Varying the scaling parameters and therefore varying the relative importance of the different terms in ATCSR could lead to good multicriteria schedules. As mentioned before the multicriteria complex job shop problem we consider in this paper is strongly NP-hard. Even at the subproblem solution procedure level, multicriteria problems are generally strongly NP-hard. It is therefore difficult to make a comment on how close the solutions generated by our technique are to being Pareto optimal or non-dominated. However, it can be easily shown that the solutions that will be picked at the SSP level using the desirability function will be the non-dominated amongst the schedules that are generated by

Table 1 Components of the ATCSR rule and the criteria they impact

	WSPT	Least slack	Shortest setup	Ready time
Makespan			✓	✓
CT	✓		✓	✓
TWT	✓	✓	✓	✓

varying the scaling parameters of ATCSR. Using ATCSR in this manner for multicriteria purposes is based on Balasubramanian et al. (2006) that empirically explore the performance of a composite dispatching rule similar to the ATCSR for single machine bicriteria scheduling.

At the SSP level, the scaling parameters are varied using a grid search approach in order to generate a wide range of schedules for subsequent consideration by the desirability function for the three objectives of interest. We use five different values for each scaling parameter and thus test 125 different combinations. Parameter k_1 is incremented from 0.1 to 2.1 in steps of 0.4; k_2 is incremented from 0.1 to 1.1 in steps of 0.2; and k_3 is incremented from 0.001 to 0.011 in steps of 0.002. These values are chosen based on some empirical pilot runs; the values also draw upon research by Chen et al. (2007a), in which parameterization of the composite dispatching rules is discussed at length.

Let l represent the total number of k_1, k_2, and k_3 scaling parameter combinations evaluated over the grid space at the SSP level. Schedule $i\,(i = 1 \ldots l)$ is characterized by its corresponding objective function values for the three performance metrics of interest, $Cmax^{(i)}$, $CT^{(i)}$, and $TWT^{(i)}$. Further, let $Cmax_{(\min)}(Cmax_{(\max)})$, $CT_{(\min)}(CT_{(\max)})$, and $TWT_{(\min)}(TWT_{(\max)})$ denote the corresponding grid space's minimum (maximum) objective function values over all l schedules evaluated at the SSP level. Let $D_{(\max)} = \max_{i=1..l} D^{(i)}$, where $D^{(i)}$ denotes the combined desirability of schedule i. We identify the schedule corresponding to $D_{(\max)}$ as S^*. Finally, let $r_{\text{Cmax}}, r_{\text{ct}}$, and r_{twt} signify the desirability weights for makespan, cycle time, and TWT, respectively.

Procedure ***SetGoals*** below determines the upper and goal values for each objective of interest by generating l schedules over the scaling parameter grid space for the toolgroup currently under study in the SSP. Then, Procedure ***FindMostDesirable*** identifies the most desirable schedule within the set of l generated schedules.

Procedure SetGoals
For $i = 1$ to l
  Create schedule i using the i-th combination of smoothing parameters k_1, k_2, and k_3 on the subproblem.
  Record $Cmax^{(i)}$, $CT^{(i)}$, and $TWT^{(i)}$.
  If $\alpha^{(i)} < \alpha_{(\min)}$ Then $\alpha_{(\min)} = \alpha^{(i)}$, $\forall \alpha \in \{Cmax, CT, TWT\}$
  If $\alpha^{(i)} > \alpha_{(\max)}$ Then $\alpha_{(\max)} = \alpha^{(i)}$, $\forall \alpha \in \{Cmax, CT, TWT\}$
Next i
Goal value $G_\alpha = \alpha_{(\min)}$, $\forall \alpha \in \{Cmax, CT, TWT\}$
Upper value $U_\alpha = \alpha_{(\max)}$, $\forall \alpha \in \{Cmax, CT, TWT\}$

Procedure FindMostDesirable
$D_{(\max)} = 0$
For $i = 1$ to l
  Calculate $D^{(i)}$ using (1) and (2)
  If $D^{(i)} > D_{(\max)}$ Then
    $D_{(\max)} = D^{(i)}$
    $S^* = i$
  End If
Next i

The upper (goal) limit is thus set to the worst (best) value observed for each objective over the l different schedules considered at the SSP level. Clearly, the upper and goal limits could be fixed, pre-determined values, as knowledge of a particular toolgroup and its performance may be known a priori in a real world setting, thereby making it easier to decide upon appropriate values for these variables. However, in the more general framework that is proposed in this research, it is intuitive to equate the worst result observed as the upper limit for a given performance, and the best result observed as the goal value.

Considering the fact that combined desirability is the geometric mean of all component desirabilities, we mandate that the schedule with the worst (highest) objective function value over all l schedules will *not* result in a component desirability of zero. Clearly, the schedule that performs worst for one objective may not necessarily perform poorly for the other objectives of interest. Alternatively, we assign a desirability value of 0.0001. Letting any $\delta_i = 0$ causes the combined desirability of schedule $i\,D^{(i)}$ to equal 0, which disqualifies schedule i for selection even when its performance for other objectives may be quite good.

4.2 MCM level

At the MCM level, the critical toolgroup is identified and then scheduled. One way of identifying the critical toolgroup is to determine each toolgroup's contribution to the overall complex job shop's TWT ("MCM-TWT"). Another technique suggested by Pinedo and Chao (1999) considers the deviation of job completion times if machine i were scheduled at the current iteration ("MCM-PC"):

$$\sum_{k=1}^{n} w_k (C_k'' - C_k') \exp\left(\frac{-\left(d_k - C_k''\right)^+}{K}\right). \tag{4}$$

In (4), $C_k'(C_k'')$ denotes the completion time of the job k before (after) machine i is scheduled and K is a scaling parameter. While the MCM-PC approach considers job completion time deviations, it does not lend itself easily to the inclusion of multiple objectives using the desirability approach. In addition, the K value used can dramatically affect which toolgroup is identified as critical at a given iteration, depending on the scheduler's sensitivity to job due date. Therefore, we consider only MCM-TWT as our base criticality measure and use the desirability approach to blend objectives other than TWT into it.

Before the desirability function can be used at the MCM level, the upper limits and goal values for each of the three objectives of interest must be determined. We employ a procedure similar to the one used at the SSP level. The only difference is that at the MCM level, we are interested in identifying and scheduling the toolgroup with the least (lowest) combined desirability value. Since the desirability value aggregates three different criteria, a low desirability value for a given toolgroup indicates that it is the most critical with respect to all the criteria under consideration and therefore should be scheduled first.

We consider two different approaches for identifying the critical toolgroup at the MCM level. First, each toolgroup schedule's impact on the makespan, cycle time, and TWT of the entire complex job shop is assessed when identifying the critical toolgroup ("Global MCM") by inserting the schedule of the toolgroups into the disjunctive graph. Alternatively, the critical toolgroup can be identified using only SSP level performance metrics ("Local MCM") (i.e., do not consider the toolgroup's impact on the rest of the complex job shop). Our goal in the proposition of these two approaches is to test whether a difference is noticeable in the global and local approaches. In a practical setting, if the MSBH procedure were to be used for scheduling, the computation time for the Local MCM approach for large problem sizes would be less that the time for the Global MCM approach. But intuitively, it would seem that the Global MCM approach would reflect the critical machine more accurately, since it takes into account conflicts between jobs in the entire wafer fab. Regardless of which MCM approach is used, the end result of this step of the MSBH is the scheduling of the critical toolgroup via the insertion of arcs into the corresponding problem's disjunctive graph.

5 Testing and experimentation

Table 2 shows the different combination of approaches at the SSP and MCM levels that we investigate in this paper. Approach 1 is the MSBH of Mason et al. (2002) that seeks to minimize TWT. The other five approaches use some combination of desirability ("Des") and TWT minimization. We employ the naming convention "SSP-γ_MCM-ω" to describe an approach that uses $\gamma \in \{\text{TWT, Des}\}$ at the SSP level and $\omega \in \{\text{TWT, Des(Local), Des(Global)}\}$ at the MCM level. For example, Approach 6 in Table 2 corresponds to SSP-Des_MCM-Des(Global). We also compare the six approaches in Table 2 with the following pure dispatching-based approaches: Critical Ratio (CR), Earliest Due Date (EDD), and First In First Out (FIFO).

Table 2 Different approaches

No.	SSP level	MCM level
1	Only TWT	Only TWT
2	Only TWT	Desirability (Local MCM)
3	Only TWT	Desirability (Global MCM)
4	Desirability	Only TWT
5	Desirability	Desirability (Simple MCM)
6	Desirability	Desirability (Global MCM)

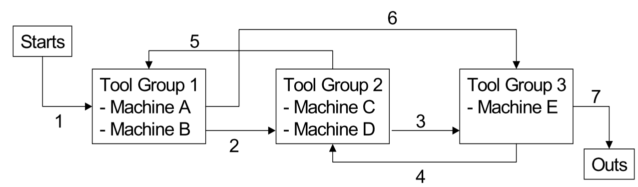

Fig. 2 The Minifab model (El Adl et al. 1996; Mason et al. 2002)

5.1 Experimental testbed

We examine two different complex job shop models in our experimental testbed. The first model, the "Minifab" model of El Adl et al. (1996), is perhaps the most succinct representation of a wafer fab in the open literature. The Minifab consists of three toolgroups and two job types that require reentrant flow during their processing (Fig. 2). Toolgroup 1 consists of two batch-processing machines with maximum batch size of three jobs, operating in parallel (e.g., a diffusion oven). Toolgroup 2 consists of two identical serial processing machines operating in parallel (e.g., a photolithography stepper), while Toolgroup 3 consists of a single machine characterized by sequence-dependent setups (e.g., an ion implanter).

We first consider 20-job static instances of the Minifab wherein $r_j = 0\ \forall j$. There are two part types in this model: 10 jobs of Part Type A and 10 jobs of Part Type B. The weights (priorities) of the jobs, regardless of part type, are generated using a discrete uniform distribution $w_j \sim DU[1, 100]$. The due date for job j is assigned using the notion of a flow factor f_j used to represent some multiple of job j's theoretical (raw) processing time RPT_j: $d_j = r_j + f_j RPT_j$, where $r_j = 0$ in the Minifab instances. In order to generate reasonable values for f_j, we examine a single static instance using FIFO dispatching at all toolgroups. From each job's completion time C_j we estimate $f_j = (C_j - r_j)/RPT_j$. Let $f_{\max} = \max_j f_j$ and $f_{\min} = \min_j f_j$. Since FIFO is independent of due-date and weight settings, only one instance is necessary to obtain $f_{\max}$ and $f_{\min}$ values. For our experiments involving the MSBH, we generate each flow factor f_j uniformly over the range $[f_{\min}, f_{\min} + f_{\max}/2]$. We generate 20 different instances of the Minifab model, each with its own unique flow factor and job weight values, using common random numbers (Law and Kelton 2000) across the scheduling approaches to better discriminate between each approach. In our initial Minifab experiments, we focus on optimizing C_{max} and TWT, disregarding CT in an attempt to illustrate the difference between each approach's performance more clearly.

The second model we examine is based on Testbed Dataset 1 of Fowler et al. (1995). The full dataset, which contains 83 toolgroups, is reduced by Mason et al. (2005) to an 11-toolgroup factory containing all bottleneck toolgroups of the original dataset with purchase prices in excess of $100,000 ("Modified Testbed Dataset 1"). Of the 11 toolgroups in Modified Testbed Dataset 1 (MTD1), three contain batch-processing machines, while two toolgroups are characterized by sequence-dependent setups. Two products types exist in MTD1: Product 1 and Product 2. Product 1 requires 73 processing steps while the second product involves 97 steps. Twenty-five jobs of each product type are considered, but unlike the Minifab model, $r_j \geq 0\,\forall j$. As was the case with the Minifab experiments, $w_j \sim DU[1, 100]$, while $f_j \sim U[1, 1.3]$ for the MTD1 instances under study. All three objectives of interest (C_{max}, CT, and TWT) are investigated for the MTD1 experimental instances.

5.2 Desirability function weights

Previously, we discussed the procedure for setting the upper and goal limits for the three different objectives at the MCM and SSP levels. However, we still must determine how desirability weights z_α will be set for $\alpha \in (Cmax, CT, TWT)$. Since the three weights are blended together into a single objective function, their ratios relative to each other are important. Experimentation with these weights involves testing the weight values between zero and one, subject to the constraint that $\sum_\alpha z_\alpha = 1$ (Myers and Montgomery 1995 and Dabbas et al. 2003), i.e., this is a mixture experiment. Table 3 describes the desirability weight settings used in our experimentation. Note that the first combination of weights in both the 2-criteria (part a) and 3-criteria (part b) optimization cases represents the typical TWT optimization associated with the MSBH of Mason et al. (2002). These weight setting combinations are obtained from an augmented simplex centroid design of either 2 or 3 variables in a mixture experiment (Myers and Montgomery 1995).

In the presentation of our results for the two models, we assume initially that the decision maker has placed equal emphasis on all the objectives being considered (i.e., $z_{Cmax} = z_{ct} = z_{twt} = 0.333$). However, as the desirability function used to choose either the schedule at the SSP level or the critical toolgroup at the MCM level is a just a heuristic procedure, it is necessary to test a number of possible combinations of weights to get a "good" final solution. If we perform an exhaustive search using all desirability weight combinations, the MSBH must be run each time a combination of weights is tested at the SSP or MCM levels. While the computational effort associated with running the MSBH for the Minifab model is insignificant, it is extremely high for the MTD1 model that contains 50 jobs and more than 70 processing steps for each product type.

To counter this problem, we use the simulation environment developed by Rose et al. (2002). The structure of the environment is shown in Fig. 3. The main purpose of the simulation environment is to emulate the behavior of a real wafer fabrication facility under different schedules provided by the Shifting Bottleneck heuristic. The environment allows for jobs arriving continuously over time in a wafer fab and the MSBH develops schedules for pre-defined time intervals. In essence this method is similar to that used by Singer (2001), which decomposes large job shops instances into smaller instances that fit in time windows and applies the Shifting Bottleneck procedure independently each time window. In the MTD1 instances considered in this paper (where the number of jobs to be scheduled is fixed) we use the Rose et al. (2002) environment as a temporal de-

Table 3 Desirability weight settings

Cmax	*TWT*
1	0
0	1
0.5	0.5
0.67	0.33
0.33	0.67
0.83	0.17
0.17	0.83

(a) 2-criteria weights

Cmax	*CT*	*TWT*
1	0	0
0	1	0
0	0	1
0.5	0.5	0
0.5	0	0.5
0	0.5	0.5
0.33	0.33	0.33
0.67	0.17	0.17
0.17	0.67	0.17
0.17	0.17	0.67

(b) 3-criteria weights

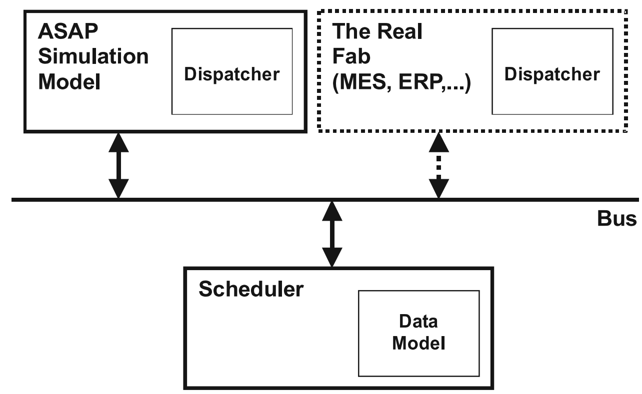

Fig. 3 Structure of the simulation and testing environment (Rose et al. 2002)

composition to reduce computational effort. The MSBH is used to schedule the complex job shop every four hours until all jobs have completed their processing. Only the uncompleted processing steps of both ready and in-process jobs are considered for scheduling during this time horizon. This rolling horizon-based decomposition procedure considerably reduces the computational effort involved with invoking the MSBH.

6 Experimental results

6.1 Bicriteria optimization for minifab model

For each of the 20 experimental instances, we use the seven different combinations of desirability weights in Table 3(a) at both the SSP and MCM levels. For the Minifab model, pilot runs suggested that SSP-TWT_MCM-Des (Global), SSP_TWT-MCM-Des (Simple), and SSP-TWT_MCM-TWT produced the same results over all 20 instances (i.e., the sequence in which the toolgroups were scheduled in the MSBH was the same for all three approaches). This is not surprising, as the Minifab contains only three toolgroups. Toolgroup 3, which contains only one machine and is subject to sequence-dependent setups (Fig. 2) is always determined to be the most critical toolgroup, followed by toolgroup 1 (the batch-processing toolgroup), then toolgroup 2. Thus, the impact of using desirability at the MCM level is not significant for the Minifab model. The MTD1 model with its 11 toolgroups provides greater potential for measuring the impact of desirability at the MCM level.

However, using desirability at the SSP level produced significantly different results for the Minifab model instances. As the SSP-level results are independent of the approach used at the MCM level, the approaches to be compared reduce to SSP-TWT and SSP-Des. The former represents the first combination in Table 3(a), while the latter represents the "best" of the remaining six combinations of desirability weights. To determine the best combination, we use the desirability function again, but now externally when the complex job shop has been fully scheduled and all objective functions have been realized.

Let $TWT_{(best)}$ denote the best (i.e., lowest) value of TWT obtained from using the following three dispatching rules: CR, EDD, and FIFO. As these dispatching rules are expected to perform poorly for TWT as they do not explicitly consider job due dates and/or weights, we use $TWT_{(best)}$ as our upper bound on TWT. Therefore, let $G_{\text{Cmax}} = Cmax_{(\min)}$, $G_{\text{twt}} = TWT_{(\min)}$, $U_{\text{Cmax}} = Cmax_{(\max)}$, and $U_{\text{twt}} = TWT_{(best)}$. At the final output level we assume initially that the decision maker places equal emphasis on both objectives ($z_{\text{Cmax}} = z_{\text{twt}} = 0.5$). Table 4

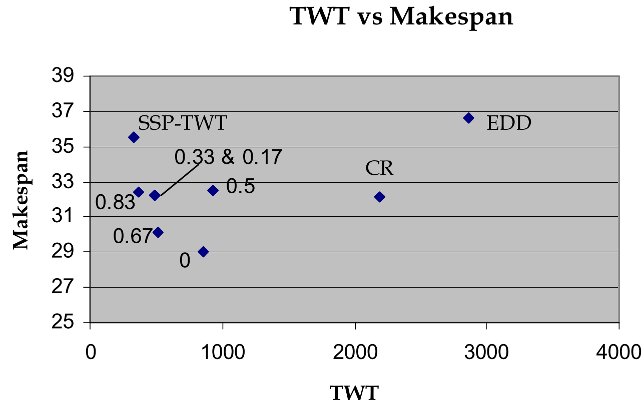

Fig. 4 Solutions in the objective space (I) for instance 1 of the Minifab model. Numbers adjacent to each point indicate the weight of total weighted tardiness criterion

shows the combined, $C_{\max}$, and TWT desirabilities for each of the 20 Minifab model instances for three different scheduling approaches (SSP-TWT, SSP-Des (with the combination of weights being chosen via pilot runs as described above), and SSP-Des* (the schedule generated using $z_{\text{Cmax}} = z_{\text{twt}} = 0.5$)) and the three competing dispatching rules (CR, EDD, and FIFO). The bold number in each row is the best desirability value of the corresponding instance.

It is clear that while SSP-TWT is superior in terms of TWT performance, its $C_{\max}$ desirability is very poor when compared to CR and SSP-Des. Therefore, its combined desirability is not good. CR and FIFO have high $C_{\max}$ desirability, but their respective TWT desirabilities over all 20 instances are poor. SSP-Des performs reasonably well for both objectives, and therefore has the best combined desirability for the 20 Minifab problem instances. SSP-TWT shares the best desirability with SSP-Des in only one of the 20 instances, instance 17. A comparison between SSP-Des and SSP-Des* reveals that SSP-Des has a desirability roughly 9% better than SSP-Des*. Further, the former performs more consistently than the latter (see standard deviation results). However, SSP-Des is computationally more expensive due to the exhaustive desirability weight search. Therefore, if a quick solution that is "good" in both objectives is desired, SSP-Des* should be used for Minifab problem instances.

Figures 4 and 5 show objective space plots of total weighted tardiness and makespan for two experimental MiniFab instances. All solution points that have been labeled by a fraction between 0 and 1 are those obtained by using the different weight combinations at the SSP level listed in Table 4. The fraction indicates the weight of total weighted tardiness used in the desirability function. The weight of makespan, of course, is the difference between 1 and the weight of total weighted tardiness. In Fig. 4 it is possible to see the tradeoff between total weighted tardiness and makespan: while the SSP-TWT solution produces

Table 4 Minifab model: desirability values for the approaches with weights $r_{\text{twt}} = 0.5$ and $r_{\text{cmax}} = 0.5$ for the combined desirability

	Combined desirability						Makespan desirability						TWT desirability					
Instance	SSP-TWT	SSP-Des	SSP-Des*	CR	EDD	FIFO	SSP-TWT	SSP-Des	SSP-Des*	CR	EDD	FIFO	SSP-TWT	SSP-Des	SSP-Des*	CR	EDD	FIFO
1	0.5995	**0.8658**	0.8217	0.1038	0.0084	0.0084	0.3697	0.7704	0.7839	**1.0000**	0.7124	0.7483	0.9721	**0.9731**	0.8614	0.0108	0.0001	0.0001
2	0.8643	**0.9729**	0.9688	0.0916	0.0015	0.0015	0.7471	**1.0000**	**1.0000**	0.7541	0.0215	0.7541	**1.0000**	0.9465	0.9386	0.0111	0.0001	0.0001
3	0.8621	**0.8640**	**0.8640**	0.1105	0.0015	0.0015	0.7432	0.7880	0.7880	**1.0000**	0.0214	0.7501	**1.0000**	0.9472	0.9472	0.0122	0.0001	0.0001
4	0.9062	**0.9213**	**0.9213**	0.0100	0.0965	0.0965	0.8486	0.8501	0.8501	**0.9967**	0.8144	0.8501	0.9678	**0.9985**	**0.9985**	0.0001	0.0114	0.0001
5	0.1343	**0.8735**	**0.7362**	0.0100	0.0932	0.0932	0.0180	0.9535	0.6739	**0.9934**	0.8087	0.8469	**1.0000**	0.8002	0.8043	0.0001	0.0108	0.0001
6	0.7056	**0.9581**	**0.9581**	0.0099	0.0174	0.0174	0.4979	0.9817	0.9817	0.9817	0.0283	**0.9909**	**1.0000**	0.9352	0.9352	0.0001	0.0107	0.0001
7	0.6149	**0.9308**	0.7623	0.0087	0.0015	0.0015	0.3781	**0.9225**	0.7331	0.7625	0.0219	0.7625	**1.0000**	0.9391	0.7927	0.0001	0.0001	0.0110
8	0.1718	**0.9792**	**0.9792**	0.0096	0.0898	0.0898	0.0307	**1.0000**	**1.0000**	0.9290	0.7290	0.9408	**0.9614**	0.9588	0.9588	0.0001	0.0111	0.0001
9	0.5360	**0.9527**	**0.9527**	0.0092	0.0014	0.0014	0.2974	**1.0000**	**1.0000**	0.8446	0.0185	0.8446	**0.9662**	0.9076	0.9076	0.0001	0.0001	0.0106
10	0.5672	**0.8677**	0.8178	0.0086	0.0152	0.0152	0.3217	0.7707	**0.7840**	0.7463	0.0213	0.7463	**1.0000**	0.9769	0.8530	0.0001	0.0108	0.0001
11	0.8845	**0.9210**	0.8924	0.0089	0.0160	0.0160	0.8008	**0.9421**	0.7963	0.7963	0.0229	0.7963	0.9770	0.9004	**1.0000**	0.0001	0.0112	0.0001
12	0.1510	**0.8413**	0.1300	0.1069	0.0080	0.0080	0.0228	0.8086	0.0228	**1.0000**	0.6346	0.6986	**1.0000**	0.8752	0.7413	0.0114	0.0001	0.0001
13	0.1376	**0.9674**	0.8583	0.0094	0.0046	0.0046	0.0189	**0.9913**	0.8863	0.8863	0.2161	0.8904	**1.0000**	0.9441	0.8311	0.0001	0.0001	0.0107
14	0.1335	**0.9142**	0.6689	0.0094	0.0941	0.0941	0.0178	**0.9606**	0.8628	0.8772	0.8416	0.8772	**1.0000**	0.8701	0.5185	0.0001	0.0105	0.0001
15	0.7947	**0.8828**	0.8489	0.1128	0.0014	0.0014	0.6315	0.9680	0.8471	**0.9965**	0.0185	0.8446	**1.0000**	0.8051	0.8507	0.0128	0.0001	0.0001
16	0.7171	**0.8811**	0.8521	0.0100	0.0015	0.0015	0.7501	**1.0000**	0.7262	0.9948	0.0214	0.7501	0.6855	0.7763	**1.0000**	0.0001	0.0001	0.0114
17	**0.8451**	**0.8451**	0.8220	0.0090	0.0015	0.0015	0.7945	0.7945	0.6757	**0.8021**	0.0231	**0.8021**	0.8989	0.8989	**1.0000**	0.0001	0.0001	0.0116
18	0.1516	**0.9166**	0.8205	0.0083	0.0077	0.0077	0.0230	**1.0000**	0.6967	0.6822	0.5925	0.6913	**1.0000**	0.8401	0.9662	0.0001	0.0001	0.0115
19	0.5274	**0.8879**	0.8595	0.0087	0.0015	0.0015	0.2781	**1.0000**	0.7561	0.7633	0.0220	0.7633	**1.0000**	0.7883	0.9771	0.0001	0.0001	0.0108
20	0.6139	**0.9431**	0.8503	0.0087	0.0015	0.0015	0.3769	**0.9520**	0.7501	0.7501	0.0214	0.7501	**1.0000**	0.9342	0.9638	0.0001	0.0001	0.0102
Average	0.5459	**0.9093**	0.8192	0.0332	0.0232	0.0232	0.3983	**0.9227**	0.7807	0.8778	0.2806	0.8049	**0.9715**	0.9008	0.8923	0.0030	0.0039	0.0045
Std.Dev	0.2923	0.0436	0.1802	0.0428	0.0364	**0.0364**	0.3106	0.0885	0.2081	0.1125	0.3477	**0.0790**	0.0717	**0.0675**	0.1178	0.0052	0.0053	0.0055

Table 5 Minifab model: performance of the approaches for a range of decision-maker's priorities

(*Cmax*, *TWT*)	SSP-Des	SSP-TWT	CR	EDD	FIFO	SSP-DM*
(0, 1)	7	13	0	0	0	13
(1, 0)	17	0	3	0	0	15
(0.5, 0.5)	19	1	0	0	0	18
(0.83, 0.17)	18	2	0	0	0	14
(0.17, 0.83)	20	0	0	0	0	20
(0.67, 0.33)	20	0	0	0	0	16
(0.33, 0.67)	18	2	0	0	0	14
avg.	17	2.57	0.43	0	0	15.71

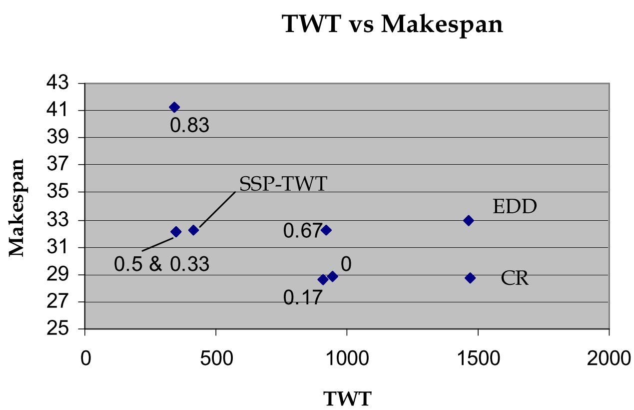

Fig. 5 Solutions in the objective space (II) for instance 4 of the Minifab model. Numbers adjacent to each point indicate the weight of total weighted tardiness criterion

the best total weighted tardiness, its makespan value is not very good. The solutions generated by the different weight combinations generate a variety of solutions, which produce slightly worse total weighted tardiness values (but still significantly better than CR or EDD solutions; FIFO results are not shown as they are really poor) but improve on the makespan. Figure 5 shows a graph where the using the desirability approach generates a solution better in both total weighted tardiness and makespan than SSP-TWT. These two different instances shown in Figs. 4 and 5 roughly classify the nature of solutions for all 20 of the instances: either a trade-off exists between the two objective values or SSP-Des generates a solution better in both objective values.

The results for combined desirability in Table 4 assume that both performance measures are equally important. However, decision makers often have differing priorities for each objective. To consider this reality, we record the number of times the most desirable solution is generated by each scheduling approach for a given a set of desirability weights (Table 5). In Table 5, (a, b) signifies the decision maker places $a\%$ priority on makespan and $b = (1 - a)\%$ priority on TWT. It is clear that SSP-Des generates the most number of desirable solutions for the 20 instances irrespective of the decision-maker's priorities except for the case of $a = 0$. In this case, SSP-TWT produces the best solutions for 65% of the instances, while SSP-Des produces the best solutions for the remaining seven instances. For all other cases in which $a > 0$, SSP-Des produces the best solutions for a minimum of 85% of the Minifab problem instances. In addition we use the notation SSP-DM* to represent the solution that uses the same combination of desirability weights as that used by the decision-maker, unlike the exhaustive search procedure of SSP-Des. For the each of the decision-maker's 7 weight combinations considered, we count the number of times that SSP-DM* produces a better solution than SSP-TWT, CR, EDD, and FIFO. These results are shown in the last column of Table 5. SSP-DM* produces a better solution than SSP-TWT, CR, EDD, and FIFO in more than 80% of the twenty cases. It was also observed that the desirability of the solution generated by SSP-DM* (obtained from a single run of the MSBH) for each of the decision-maker's priorities is very close to the desirability of the solution generated from SSP-Des (obtained by exhaustive search).

6.2 3-criteria optimization for MTD1 model

The MTD1 model with 11 toolgroups provides an opportunity to test both the MCM approaches proposed in this paper and the approaches where desirability optimization at the MCM level is used in conjunction with desirability optimization at the SSP level. For setting the upper and goal limits for the desirability calculations, we use the same techniques employed for the Minifab model described above. Initially, we again assume equal importance of all three objectives. For SSP-TWT_MCM-Des (Global) and SSP-TWT_MCM-Des (Local), we test the nine of the 10 different combinations of weights in Table 3(b), as the first combination is again excluded as it represents the SSP-TWT_MCM-TWT case. Our goal in this testing is to identify the combination of γ (approach at the SSP level) and ω (approach at the MCM level) that produce the most desirable results. Similarly, nine different desirability weight combinations are tested for SSP-Des_MCM-TWT to identify the most desirable schedules.

Clearly, the three approaches mentioned above are computationally intensive considering the exhaustive desirability weight search. Therefore, we also check the performance of these approaches when only one combination of weights (i.e., equal importance to all objectives) is used at the SSP and MCM levels. These cases are denoted by SSP-TWT_MCM-Des (Global)*, SSP-TWT_MCM-Des (Local)*, and SSP-Des_MCM-TWT*. For SSP-Des_MCM-Des (Global) and SSP-Des_MCM-Des (Local) desirability optimization is used at both levels. Ideally, therefore, an exhaustive search would mean 100 (10×10) runs would be required before all weight combinations can be explored. Even a set of 5 desirability combinations at each level would require 25 runs for each instance before the best can be chosen. Since the number of runs is prohibitive, for these two approaches, at both SSP and MCM levels we use the combination $(0.333, 0.333, 0.333)$. This combination of equal weight importances is also motivated by our initial pilot run comparisons in which we assume equal emphasis is placed on all objectives.

Table 6 shows both the average and standard deviation of the combined desirability values of each approach while Tables 7, 8, and 9 show the individual desirability values of makespan, cycle time, and total weighted tardiness, respectively, for each of the 20 instances of the MTD1 model, assuming that all objectives have equal importance to the decision maker (i.e., $z_{\mathrm{Cmax}} = z_{\mathrm{ct}} = z_{\mathrm{twt}} = 0.333$). It is clear that SSP-Des_MCM-TWT, SSP-Des_MCM-TWT*, SSP-Des_MCM-Des (Global), and SSP-Des_MCM-Des (Local) perform well for all three measures. In fact, the TWT-desirability of SSP-TWT_MCM-TWT, though much better than that of CR on average, is still significantly lower than the approaches that use desirability at the SSP level. The results for combined desirability follow a similar pattern: SSP-Des_MCM-TWT has the highest mean combined desirability, closely followed by SSP-Des_MCM-Des (Global) and SSP-Des_MCM-Des (Local). Further, all three approaches have low standard deviations, an indication of their consistency. The approaches that use desirability optimization only at the MCM level perform well in one or two instances, but fall short in their combined mean desirabilities. Therefore, using desirability merely to choose a critical machine does not appear to be effective consistently unless the schedule implemented at the SSP level is also chosen using a desirability approach.

It is also clear from Table 6 that SSP-TWT_MCM-Des (Global)*, SSP-TWT_MCM-Des (Local)*, and SSP-Des_MCM-TWT* have solution qualities close to their computationally intensive counterparts (i.e., SSP-TWT_MCM-Des (Global), SSP-TWT_MCM-Des (Local), and SSP-Des_MCM-TWT). It is particularly encouraging to notice the performance of SSP-Des_MCM-TWT*, SSP-Des_MCM-Des (Global), and SSP-Des_MCM-Des (Local), as their solutions are generated by a single MSBH run requiring 25 minutes on a 2.0 GHz Pentium IV computer with 1 GB of RAM. On average, these three "single pass" approaches' desirabilities are within 6% of the desirability of SSP-Des_MCM-TWT, which requires 225 minutes to generate the best solution for a single problem instance.

Table 10 compares the performance of SSP-TWT_MCM-TWT (i.e., the original MSBH of Mason et al. 2002) with the approach that produced $D_{(\max)}$ for each of the 20 MTD1 problem instances via a performance ratio. For each problem instance, objective $\alpha \in (Cmax, CT, TWT)$ resulting from the approach that produced $D_{(\max)}$ is divided by the corresponding objective of the SSP-TWT_MCM-TWT approach for the same problem instance. This performance ratio allows us to analyze the quality of the most desirable schedule in terms of percentage gain/loss with respect to each objective α. Surprisingly, the average gain in TWT over the TWT-based MSBH is 28%. Further, the average gain in $C_{\max}$ performance is 4–5%, while cycle time is decreased by 2% on average using a desirability-based approach. Therefore, in the case of the MTD1 model, the desirability-based approaches generated schedules superior in all three objectives as compared to the MSBH. Although the gains in $C_{\max}$ and cycle time are relatively small, these small gains have contributed significantly to the large gains in TWT performance since these measures are not independent of each other.

Table 11 displays the performance of the desirability approaches for the MTD1 model when the decision maker may or may not place equal importance on each objective. In order to evaluate the performance of each approach for different priorities weightings, we count the number of times the best solution is produced by each approach for a given set of priorities. Table 11 shows 10 possible combinations of decision maker priorities (a, b, c) and the number of times that each approach produces the best solution for a given (a, b, c) weighting scheme. Under this weighting scheme, the decision maker places a% weight on $C_{\max}$, b% weight on cycle time, and $c = (1 - a - b)$% weight on TWT. Experimental results suggest that SSP-Des_MCM-TWT is the rule that performs well most often for a wide range of decision maker priorities, followed by SSP-Des_MCM-Des (Local) and SSP-Des_MCM-Des (Global). Additionally, it was also observed, just as in the Minifab cases, that for all approaches that use desirability at the SSP level, if only the corresponding decision-maker's weights were used (therefore requiring only 1 run of the MSBH), the desirability of the solutions were close in most cases to the desirability of the solutions generated by SSP-Des (which requires exhaustive search). Therefore, the clear conclusion from these experimental results is that it is important to implement schedules at the toolgroup level that take into account multiple criteria.

Table 6 Desirability Results for the MTD1 Model Across 20 Problem Instances for $r_{\mathrm{Cmax}} = r_{\mathrm{ct}} = r_{\mathrm{twt}} = 0.333$

Instance	SSP-TWT MCM-TWT	SSP-Des MCM-TWT	SSP-Des MCM-TWT*	SSP-TWT MCM-Des(G)	SSP-TWT MCM-Des(G)*	SSP-TWT MCM-Des(L)	SSP-TWT MCM-Des(L)*	SSP-Des MCM-Des(G)	SSP-Des MCM-Des(L)	CR	EDD	FIFO
1	0.759	0.759	0.044	0.041	0.026	0.039	0.039	0.883	**0.982**	0.418	0.041	0.018
2	0.851	**0.968**	**0.968**	0.944	0.889	0.897	0.897	0.887	0.937	0.044	0.431	0.011
3	0.800	0.978	0.978	0.834	0.816	0.920	0.774	**0.995**	0.984	0.393	0.043	0.011
4	0.730	0.979	0.970	0.859	0.797	0.865	0.781	0.953	**0.995**	0.043	0.403	0.011
5	0.721	0.973	0.973	0.874	0.776	0.628	0.469	**0.999**	0.925	0.341	0.038	0.010
6	0.599	0.992	0.929	0.745	0.609	**0.999**	0.938	0.973	0.864	0.378	0.044	0.011
7	0.041	**1.000**	0.989	0.041	0.041	0.038	0.021	0.958	0.959	0.366	0.040	0.018
8	0.040	**1.000**	**1.000**	0.658	0.040	0.040	0.040	0.899	0.894	0.370	0.041	0.010
9	0.041	0.876	0.876	0.896	0.878	0.900	0.900	**1.000**	0.917	0.384	0.039	0.010
10	0.699	0.962	0.937	0.763	0.681	0.716	0.645	0.983	**0.984**	0.356	0.039	0.011
11	0.826	**0.998**	0.990	0.793	0.793	0.726	0.671	0.908	0.966	0.041	0.337	0.018
12	0.799	**1.000**	0.970	0.901	0.808	0.662	0.662	0.863	0.739	0.357	0.039	0.017
13	0.041	**0.998**	0.997	0.041	0.041	0.042	0.042	0.971	0.952	0.358	0.041	0.011
14	0.733	0.933	0.933	0.878	0.799	**0.952**	0.712	0.893	0.870	0.039	0.349	0.011
15	0.039	0.950	0.946	0.043	0.022	0.042	0.042	0.952	**0.973**	0.042	0.396	0.015
16	0.823	0.944	0.926	0.836	0.656	0.827	0.827	**0.978**	0.931	0.404	0.041	0.011
17	0.798	0.992	0.987	0.852	0.852	0.836	0.825	**0.998**	0.915	0.041	0.351	0.010
18	0.717	0.902	0.902	0.674	0.640	0.817	0.817	0.955	**0.963**	0.041	0.356	0.018
19	0.772	**1.000**	**1.000**	0.944	0.885	0.895	0.832	0.974	0.896	0.370	0.043	0.011
20	0.722	0.868	0.857	0.777	0.777	0.787	0.787	0.908	**1.000**	0.039	0.304	0.010
Ave	0.578	**0.954**	0.908	0.670	0.591	0.631	0.586	0.947	0.932	0.241	0.171	0.013
Std. Dev	0.323	0.061	0.208	0.331	0.339	0.362	0.342	**0.045**	0.061	0.168	0.165	0.003

Table 7 MTD Model: Makespan-Desirability of the approaches

Instance	SSP-TWT MCM-TWT	SSP-Des MCM-TWT	SSP-Des MCM-TWT*	SSP-TWT MCM-Des(G)	SSP-TWT MCM-Des(G)*	SSP-TWT MCM-Des(L)	SSP-TWT MCM-Des(L)*	SSP-Des MCM-Des(G)	SSP-Des MCM-Des(L)	CR	EDD	FIFO
1	0.791	0.791	0.889	0.742	0.198	0.644	0.644	0.927	0.948	**1.000**	0.844	0.561
2	0.773	**0.998**	**0.998**	0.840	0.824	0.810	0.810	0.988	0.926	0.952	0.993	0.125
3	0.619	0.986	0.986	0.754	0.714	0.911	0.705	**1.000**	0.978	0.936	0.902	0.118
4	0.692	0.989	0.986	0.784	0.771	0.796	0.762	0.913	**1.000**	0.857	0.873	0.116
5	0.731	0.930	0.930	0.855	0.741	0.508	0.263	**1.000**	0.949	0.815	0.653	0.111
6	0.558	**1.000**	0.918	0.833	0.674	0.997	0.855	0.989	0.887	0.815	0.947	0.123
7	0.775	**1.000**	0.992	0.784	0.781	0.632	0.102	0.955	0.940	0.854	0.759	0.564
8	0.718	**1.000**	**1.000**	0.727	0.712	0.744	0.744	0.939	0.909	0.848	0.847	0.109
9	0.754	0.789	0.789	0.820	0.827	0.925	0.925	**1.000**	0.956	0.804	0.723	0.113
10	0.630	0.940	0.880	0.815	0.662	0.688	0.609	0.970	**1.000**	0.746	0.685	0.121
11	0.844	0.995	0.980	0.780	0.780	0.714	0.662	0.883	**1.000**	0.853	0.786	0.582
12	0.865	**1.000**	0.937	0.934	0.795	0.838	0.838	0.873	0.844	0.838	0.750	0.463
13	0.837	**1.000**	0.996	0.829	0.822	0.812	0.812	0.985	0.953	0.769	0.778	0.119
14	0.782	0.944	0.943	0.961	0.927	0.863	0.397	0.950	**1.000**	0.723	0.761	0.123
15	0.661	0.913	0.911	0.835	0.119	0.797	0.797	0.863	**1.000**	0.789	0.854	0.301
16	0.711	0.855	0.810	0.758	0.679	0.765	0.765	0.957	**1.000**	0.911	0.762	0.115
17	0.752	0.977	0.964	0.874	0.874	0.849	0.772	**1.000**	0.985	0.800	0.789	0.112
18	0.567	0.936	0.936	0.463	0.379	0.739	0.739	0.870	**1.000**	0.822	0.806	0.556
19	0.673	**1.000**	**1.000**	0.970	0.809	0.878	0.823	0.958	0.887	0.869	0.939	0.118
20	0.661	0.847	0.830	0.691	0.691	0.689	0.689	0.831	**1.000**	0.722	0.555	0.105
Ave	0.720	0.944	0.934	0.803	0.689	0.763	0.668	0.943	**0.958**	0.836	0.800	0.233
Std. Dev	0.087	0.071	0.065	0.109	0.213	0.131	0.208	0.053	**0.047**	0.073	0.105	0.191

Table 8 MTD model: CT-Desirability of the approaches

Instance	SSP-TWT MCM-TWT	SSP-Des MCM-TWT	SSP-Des MCM-TWT*	SSP-TWT MCM-Des(G)	SSP-TWT MCM-Des(G)*	SSP-TWT MCM-Des(L)	SSP-TWT MCM-Des(L)*	SSP-Des MCM-Des(G)	SSP-Des MCM-Des(L)	CR	EDD	FIFO
1	0.987	0.987	0.940	0.949	0.926	0.949	0.949	0.976	**1.000**	0.977	0.820	0.106
2	0.977	0.978	0.978	**1.000**	0.982	0.988	0.988	0.937	0.959	0.888	0.938	0.106
3	0.992	0.992	0.992	0.982	0.985	**0.997**	0.953	0.996	0.975	0.912	0.856	0.104
4	0.951	**1.000**	0.997	0.988	0.971	0.990	0.967	0.975	0.984	0.927	0.919	0.109
5	0.898	**1.000**	**1.000**	0.943	0.923	0.884	0.867	0.998	0.961	0.829	0.828	0.101
6	0.941	0.978	0.961	0.945	0.937	**1.000**	0.995	0.960	0.922	0.911	0.873	0.105
7	0.874	**1.000**	0.996	0.900	0.881	0.875	0.884	0.983	0.996	0.889	0.821	0.102
8	0.911	**1.000**	**1.000**	0.921	0.893	0.885	0.885	0.944	0.963	0.908	0.838	0.101
9	0.929	0.982	0.982	0.982	0.974	0.962	0.962	**1.000**	0.946	0.895	0.838	0.101
10	0.937	**1.000**	0.994	0.931	0.921	0.918	0.909	0.978	0.975	0.902	0.884	0.109
11	0.899	**1.000**	0.998	0.901	0.901	0.879	0.880	0.939	0.957	0.824	0.826	0.099
12	0.931	**1.000**	0.995	0.970	0.938	0.957	0.957	0.926	0.878	0.864	0.816	0.101
13	0.853	**1.000**	0.997	0.827	0.826	0.888	0.888	0.974	0.954	0.906	0.909	0.106
14	0.896	0.955	0.955	0.934	0.920	**1.000**	0.987	0.941	0.868	0.794	0.855	0.105
15	0.924	0.990	0.983	0.952	0.924	0.915	0.915	**1.000**	0.955	0.908	0.921	0.108
16	0.978	**1.000**	**1.000**	0.971	0.954	0.957	0.957	0.979	0.944	0.936	0.925	0.105
17	0.922	**1.000**	0.998	0.931	0.931	0.928	0.940	0.997	0.946	0.858	0.841	0.102
18	0.902	0.905	0.905	0.912	0.921	0.902	0.902	**1.000**	0.954	0.862	0.847	0.103
19	0.935	**0.999**	**0.999**	0.969	0.954	0.960	0.946	0.982	0.939	0.878	0.865	0.102
20	0.895	0.928	0.925	0.928	0.928	0.934	0.934	0.969	**1.000**	0.848	0.842	0.100
Ave	0.926	**0.985**	0.980	0.942	0.929	0.938	0.933	0.973	0.954	0.886	0.863	0.104
Std. Dev	0.038	0.026	0.028	0.040	0.037	0.044	0.039	**0.024**	0.035	0.043	0.039	0.003

Table 9 MTD Model: TWT-Desirability of the approaches

Instance	SSP-TWT MCM-TWT	SSP-Des MCM-TWT	SSP-Des MCM-TWT*	SSP-TWT MCM-Des(G)	SSP-TWT MCM-Des(G)*	SSP-TWT MCM-Des(L)	SSP-TWT MCM-Des(L)*	SSP-Des MCM-Des(G)	SSP-Des MCM-Des(L)	CR	EDD	FIFO
1	0.5611	0.5611	0.0001	0.0001	0.0001	0.0001	0.0001	0.7610	**1.0000**	0.0745	0.0001	0.0001
2	0.8148	0.9288	0.9288	**1.0000**	0.8684	0.9018	0.9018	0.7547	0.9252	0.0001	0.0860	0.0001
3	0.8329	0.9567	0.9567	0.7838	0.7730	0.8572	0.6902	0.9904	**1.0000**	0.0712	0.0001	0.0001
4	0.5920	0.9491	0.9290	0.8173	0.6757	0.8203	0.6467	0.9719	**1.0000**	0.0001	0.0818	0.0001
5	0.5712	0.9912	0.9912	0.8294	0.6823	0.5505	0.4523	**1.0000**	0.8673	0.0586	0.0001	0.0001
6	0.4091	0.9977	0.9096	0.5256	0.3578	**1.0000**	0.9716	0.9695	0.7896	0.0727	0.0001	0.0001
7	0.0001	**1.0000**	0.9774	0.0001	0.0001	0.0001	0.0001	0.9373	0.9422	0.0644	0.0001	0.0001
8	0.0001	**1.0000**	1.0000	0.4250	0.0001	0.0001	0.0001	0.8193	0.8156	0.0655	0.0001	0.0001
9	0.0001	0.8669	0.8669	0.8927	0.8415	0.8194	0.8194	**1.0000**	0.8525	0.0789	0.0001	0.0001
10	0.5802	0.9472	0.9390	0.5855	0.5184	0.5817	0.4840	**1.0000**	0.9765	0.0673	0.0001	0.0001
11	0.7424	**1.0000**	0.9919	0.7090	0.7090	0.6100	0.5189	0.9021	0.9410	0.0001	0.0589	0.0001
12	0.6330	**1.0000**	0.9799	0.8072	0.7066	0.6222	0.6222	0.7956	0.5446	0.0628	0.0001	0.0001
13	0.0001	**0.9941**	0.9987	0.0001	0.0001	0.0001	0.0001	0.9544	0.9488	0.0656	0.0001	0.0001
14	0.5630	0.9013	0.9018	0.7547	0.5974	**1.0000**	0.9205	0.7964	0.7577	0.0001	0.0655	0.0001
15	0.0001	0.9484	0.9441	0.0001	0.0001	0.0001	0.0001	**1.0000**	0.9652	0.0001	0.0790	0.0001
16	0.8008	0.9849	0.9797	0.7952	0.4349	0.7717	0.7717	**1.0000**	0.8547	0.0774	0.0001	0.0001
17	0.7331	**1.0000**	0.9987	0.7599	0.7599	0.7411	0.7741	0.9985	0.8217	0.0001	0.0653	0.0001
18	0.7198	0.8662	0.8662	0.7241	0.7517	0.8184	0.8184	**1.0000**	0.9342	0.0001	0.0659	0.0001
19	0.7319	**1.0000**	**1.0000**	0.8968	0.8981	0.8494	0.7409	0.9834	0.8629	0.0661	0.0001	0.0001
20	0.6365	0.8306	0.8193	0.7317	0.7317	0.7576	0.7576	0.9294	**1.0000**	0.0001	0.0599	0.0001
Ave	0.496	**0.936**	0.899	0.602	0.515	0.585	0.545	0.928	0.890	0.041	0.028	0.000
Std. Dev	0.311	0.103	0.218	0.334	0.332	0.367	0.350	**0.090**	0.111	0.035	0.036	0.000

7 Model parameters

Our model and solution approach assume information on the due-dates and weights of the jobs. It also requires specifying the scaling parameters for the ATCSR heuristic at the SSP level and desirability weights that need to be used at both SSP and MCM levels. In this section, we discuss the setting of these parameters.

Table 10 MTD model: Comparison of SSP-TWT_MCM-TWT with the best of the desirability approaches

Instance	Ratios		
	MkSp	CT	TWT
1	0.9703	0.9964	0.7904
2	0.9930	0.9940	0.9206
3	0.9525	0.9988	0.8689
4	0.9584	0.9884	0.8761
5	0.9543	0.9714	0.5509
6	0.9556	0.9844	0.7510
7	0.9474	0.9665	0.5322
8	0.9500	0.9739	0.6144
9	0.9594	0.9792	0.7803
10	0.9664	0.9851	0.7687
11	0.9616	0.9696	0.6352
12	0.9673	0.9791	0.6614
13	0.9747	0.9664	0.6452
14	0.9908	0.9740	0.6845
15	0.9786	0.9820	0.7230
16	0.9820	0.9939	0.8918
17	0.9569	0.9785	0.7453
18	0.9627	0.9735	0.7450
19	0.9579	0.9811	0.7547
20	0.9368	0.9687	0.6040
Ave	0.9638	0.9802	0.7272
Std. Dev	0.0144	0.0101	0.1114

7.1 Job due-dates and weights

The due-dates could be vendor's promises or the customer's requirements; in either case, these due-dates can be used in our model. Companies also have customers that are more important than others, and this information can be used to set weights for the products such that they reflect the relative importance of a product to another.

7.2 Desirability weights

It is important to make a distinction between the weights the decision maker chooses to assign to each criterion and the weights that are assigned at the SSP level or MCM level. We refer to Sect. 5.2 for a description of how these sets of weights are identified. While it is intuitive to set the desirability weights at the SSP and MCM level to be equal to the decision maker's desirability weights, for better solutions a more exhaustive search for desirability weight combinations may be required at the SSP and MCM levels. Provided that there is a simulation model of the fab, it would be reasonable to state that this process would take 1–3 days of computing time and analysis. We expect that the results would have to be reassessed and the experimentation carried out again when there are changes in fab with respect to technology, processes, and product mix.

7.3 Scaling parameters for ATCSR

At the SSP level, we use a grid approach to set the scaling parameters. The scaling parameters are dependent on

Table 11 MTD model: Performance of the approaches for a range of decision-maker's priorities, indicated weights are ($Cmax$, CT, TWT)

($Cmax$, CT, TWT)	SSP-TWT MCM-TWT	SSP-Des MCM-TWT	SSP-TWT MCM-Des(G)	SSP-TWT MCM-Des(L)	SSP-Des MCM-Des(G)	SSP-Des MCM-Des(L)	CR	EDD	FIFO
(0, 0, 1)	0	7	1	2	6	4	0	0	0
(0, 1, 0)	0	7	0	0	4	8	1	0	0
(1, 0, 0)	0	12	1	2	3	2	0	0	0
(0, 0.5, 0.5)	0	8	0	1	5	6	0	0	0
(0.5, 0, 0.5)	0	8	1	2	6	3	0	0	0
(0.5, 0.5, 0)	0	9	0	1	4	5	1	0	0
(0.33, 0.33, 0.33)	0	7	0	2	5	6	0	0	0
(0.17, 0.17, 0.66)	0	6	1	2	8	3	0	0	0
(0.17, 0.66, 0.17)	0	8	0	1	5	6	0	0	0
(0.66, 0.17, 0.17)	0	8	0	2	8	2	0	0	0
Average	0	8	0.4	1.5	5.4	4.5	0.2	0	0

the characteristics of the scheduling instance such as due-date tightness, weights, release date range, and setup severity factor. Several different approaches can be used to set these parameters. Lee and Pinedo (1997) use curve-fitting methods to set the scaling parameters for a similar composite dispatching rule; they present expressions for the scaling parameters in terms of the characteristics of the problem instance. More recently, regression based approaches have been used to set the parameters. Gadkari et al. (2007) present a regression model to determine the values of the scaling parameters. Chen et al. (2007a) improve upon this approach to reduce significantly the number of experiments needed for data collection. Finally, Chen et al. (2007b) present an efficient approach to determine the robust scaling parameters—i.e., the parameters that work for well for all scheduling instances. Such techniques can help find the ranges for the scaling parameters for our problem.

Formulas and values suggested in aforementioned papers could be used directly, or the analyst could redo the experimentation proposed in them. As with desirability weights, this will be part of a pilot study using a fab simulation model and historical data. The computational effort should be less than one day, given that the composite dispatching rule can be easily implemented and computes schedules quickly. As before, a periodic reassessment of these parameters is warranted with changes in the wafer fab with respect to technology, processes, and product mix.

8 Conclusions and future research

Semiconductor wafer fabrication is a complex process typically requiring hundreds of steps with unique features such as re-entrant flows, batch machines, and sequence dependent setups. These features can be modeled as a complex job shop. The MSBH heuristic of Mason et al. (2002) considers the minimization of TWT in a complex job shop. We build on this and propose a methodology for multicriteria optimization using the desirability function. Given the strongly NP-hard nature of this multicriteria problem, our approach provides a computationally feasible way of accommodating multiple criteria.

We use the desirability approach at two different levels of the MSBH, the SSP level and the MCM level, and propose five new approaches for scheduling complex job shops. Using two representative complex job shop models from the literature, we compare our approaches to the original MSBH (approach "SSP-TWT_MCM-TWT") as well as three dispatching rules. Twenty problem instances were generated and analyzed for each of the two representative complex job shop models (20-job instances of the Minifab and 50-job instances of MTD1). External to the MSBH, we use the desirability function to compare the schedules generated by the different competing approaches. Experimental results show that when equal emphasis is placed on all three objectives, the desirability approach performs significantly better than both the MSBH and dispatching rules. While a tradeoff between $C_{\max}$ and TWT was observed in the Minifab experiments, the desirability approaches performed the best in all three objectives in the MTD1 experiments. An important conclusion from our experimentation is that the use of a desirability approach at the SSP level consistently produces superior results. We are available by email (see corresponding author's contact information) to answer any questions that readers may have regarding the algorithms in this paper as well as the experimental results.

In the future, we plan to explore the use of the desirability approach to approximately generate the efficient frontier for the complex job shop environment, as this information could prove quite useful to decision makers to help to understand the inherent trade-off between competing objectives. Further, we will extend the approaches proposed in this paper to the practical case wherein dynamic job arrivals are present.

Acknowledgements This research was partially supported by the Semiconductor Research Corporation and International Sematech through Factory Operations Research Center (FORCe) grant 2001-NJ-880.

References

Adams, J., Balas, E., & Zawack, D. (1988). The Shifting Bottleneck procedure for job shop scheduling. *Management Science*, *34*, 391–401.

Balas, E., Lancia, G., Serafini, P., & Vazacopoulos, A. (1998). Job shop scheduling with deadlines. *Journal of Combinatorial Optimization*, *1*, 329–353.

Balas, E., Lenstra, J. K., & Vazacopoulos, A. (1995). The one-machine problem with delayed precedence constraints and its use in job shop scheduling. *Management Science*, *41*(1), 94–109.

Balas, E., & Vazacopoulos, A. (1998). Guided local search with shifting bottleneck for job shop scheduling. *Management Science*, *44*(2), 262–275.

Balas, E., Simonetti, N., & Vazacopoulos, A. (2005). Job shop scheduling with set-up times, deadlines, and precedence constraints. In *Proceedings of the 2nd multidisciplinary international conference on scheduling: theory and applications* (pp. 520–532). New York: MISTA.

Balasubramanian, H., Fowler, J. W., & Pfund, M. E. (2006). Single machine bicriteria scheduling using the apparent tardiness cost heuristic. In *Industrial engineering research conference*, Orlando, FL, USA.

Chen, Y., Fowler, J. W., Pfund, M. E., & Montgomery, D. C. (2007a). *Methodologies for parameterization of composite dispatching rules ASUIE-ORPS-2007-019*. Industrial Engineering, Arizona State University. http://ie.fulton.asu.edu/research/workingpaper/wps.php

Chen, Y., Pfund, M. E., Montgomery, D. C., Fowler, J. W., & Callarman, T. E. (2007b). *Robust scaling parameters for composite dispatching rules ASUIE-ORPS-2007-020*. Industrial Engineering, Arizona State University. http://ie.fulton.asu.edu/research/workingpaper/wps.php

Cochran, J. K., Horng, S. M., & Fowler, J. W. (2003). A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Computers and Operations Research*, *30*, 1087–1102.

Dabbas, R., Fowler, J., Rollier, D., & McCarville, D. (2003). Multiple response optimization using mixture-designed experiments and desirability functions in semiconductor scheduling. *International Journal of Production Research*, *41*, 939–961.

Dauzère-Pérès, S., & Lasserre, J. B. (1993). A modified Shifting Bottleneck procedure for the job shop scheduling. *International Journal of Production Research*, *31*, 923–932.

Derringer, G., & Suich, R. (1980). Simultaneous optimization of several response variables. *Journal of Quality Technology*, *12*, 214–219.

El Adl, M. K., Rodriguez, A. A., & Tsakalis, K. S. (1996). Hierarchical modeling and control of re-entrant semiconductor manufacturing facilities. In *Proceedings of the 35th conference on decision and control*, Kobe, Japan.

Esquivel, S. C., Ferrero, S. W., & Gallard, R. H. (2002). Parameter settings and representations in Pareto-based optimization for job shop scheduling. *Cybernetics and Systems*, *33*, 559–578.

Foote, B. L., Ravindran, A., & Lashine, S. (1988). Computational feasibility of multi-criteria models of production, planning and scheduling. *Computers and Industrial Engineering*, *15*, 129–138.

Fowler, J., & Pfund, M. (2001). *State of the art scheduling survey results*. Semiconductor Research Corporation Publication Number P003240

Fowler, J. W., Feigin, G., & Leachman, R. (1995). *Semiconductor manufacturing Testbed: data sets*. Arizona State University

Gadkari, A., Pfund, M. E., Fowler, J. W., & Chen, Y. Scheduling jobs on parallel machines with setup times and ready times, *Computers and Industrial Engineering*, to appear (2007)

Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, *5*, 287–326.

Hoogeveen, H. (2005). Multicriteria Scheduling, *European Journal of Operational Research*, *167*(3).

Holtsclaw, H. H., & Uzsoy, R. (1996). Machine criticality measures and subproblem solution procedures in shifting bottleneck methods: a computational study. *Journal of the Operational Research Society*, *47*(5), 666–677.

Iima, H., Hara, T., Ichimi, N., & Sannomiya, N. (1999). Autonomous decentralized scheduling algorithm for a job-shop scheduling problem with complicated constraints. In *Proceedings of the 1999 4th international symposium on autonomous decentralized systems* (pp. 366-369), Tokyo, Japan.

Itoh, K., Huang, D., & Enkawa, T. (1993). Twofold look-ahead search for multi-criterion job shop scheduling. *International Journal of Production Research*, *31*(9), 2215–2234.

Ivens, Ph., & Lambrecht, M. (1996). Extending the Shifting Bottleneck procedure to real-life applications. *European Journal of Operational Research*, *90*, 252–268.

Kacem, I., Hammadi, S., & Borne, P. (2002). Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. *Mathematics and Computers in Simulation*, *60*, 245–276.

Law, A. M., & Kelton, W. D. (2000). *Simulation modeling and analysis*. Boston: Mcgraw-Hill.

Lee, Y. H., & Pinedo, M. Scheduling jobs in parallel machines with sequence dependent setup times, *European Journal of Operational Research*, *100*, 464–474

Mason, S. J., Fowler, J. W., & Carlyle, W. M. (2002). A modified Shifting Bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling*, *5*(3), 247–262.

Mason, S. J., Fowler, J. W., Carlyle, W. M., & Montgomery, D. C. (2005). Heuristics for minimizing total weighted tardiness in complex job shops. *International Journal of Production Research*, *43*(10), 1943–1963.

Myers, R., & Montgomery, D. (1995). *Response surface methodology*. New York: Wiley.

Nagar, A., Haddock, J., & Heragu, S. (1995). Multiple and bicriteria scheduling: a literature survey. *European Journal of Operational Research*, *81*, 88–104.

Neacy, E., Brown, S., & McKiddie, R. *Measurement and improvement of manufacturing capacity (MIMAC) survey and interview results*. SEMATECH Technology Transfer #94052374A-XFR (1994)

Ovacik, I. M., & Uzsoy, R. (1992). A Shifting Bottleneck algorithm for scheduling semiconductor testing operations. *Journal of Electronic Manufacturing*, *2*, 119–134.

Pinedo, M. L., & Chao, X. (1999). *Operations scheduling with applications in manufacturing and services*. New York: Irwin/McGraw-Hill.

Pinedo, M. L., & Singer, M. (1999). A Shifting Bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics*, *46*, 1–17.

Rose, O., Mönch, L., & Sturm, R. (2002). Testing, comparison and implementation issues. In *Proceedings of the 12th international conference on flexible automation and intelligent manufacturing*, Dresden, Germany.

Schutten, M. (1998). Practical job shop scheduling. *Annals of Operational Research*, *83*, 161–177.

Semiconductor Industry Association (SIA) (2005). The international technology roadmap for semiconductors. http://public.itrs.net/.

Singer, M. (2001). Decomposition methods for large job shops. *Computers and Operation Research*, *28*, 193–207.

T'kindt, V. T., & Billaut, J.-C. (2006). *Multicriteria scheduling: theory, models and algorithms* (2nd ed.). Berlin: Springer.