Proceedings of the 2007 Winter Simulation Conference
S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.

ONLINE MULTIOBJECTIVE SINGLE MACHINE DYNAMIC SCHEDULING WITH SEQUENCE-DEPENDENT SETUPS USING SIMULATION-BASED GENETIC ALGORITHM WITH DESIRABILITY FUNCTION

Adeline T. H. Ang
Appa Iyer Sivakumar

School of Mechanical and Aerospace Engineering
50 Nanyang Avenue, Nanyang Technological University
Singapore 639798, SINGAPORE

ABSTRACT

This paper presents a Simulation-based Genetic Algorithm with Desirability function (SIMGAD) that could be used on-line for the dynamic scheduling of a single machine with sequence-dependent setups. The weights used to combine the criteria (dispatching rules) into a single rule using linear weighted aggregation is determined by genetic algorithm (GA). The GA evaluates the performance of each set of weights with discrete-event simulation that returns a fitness value after multiple performance measures (objectives) are each expressed as a desirability function and combined into a single objective function. An illustrative simulation example based on the scheduling of an ion implanter machine in wafer fabrication plant shows that SIMGAD works effectively in solving the multiobjective scheduling problem with capability of handling user preference in decision making to achieve the desired performances.

1 INTRODUCTION

Semiconductor manufacturing is probably one of the most complex systems in terms of equipment, manufacturing routes, and system dependency, and this poses great challenges for production planning and scheduling, as reported in Sivakumar and Gupta (2006). Review papers by Zhu and Wilhelm (2006) have also shown that few papers have considered the combined aspects of scheduling (i) of a stochastic nature, (ii) with sequence-dependent setup (SDS), and (iii) multiobjectives. SDS time implies that the setup time depends on both the part that has been processed and the next part to be processed.

Since stochastic scheduling problems involving SDS are strongly NP-hard (Baker 1974), heuristics in the form of dispatching rules and simulations are suitable solution methods. Dispatching is commonly used in practice due to its robustness (i.e. ability to react to uncertainties), low computational requirements, ease of implementation, and intuitive appeal (i.e. easy to comprehend). However, simple rules such as first-in, first-out (FIFO), shortest processing time (SPT), earliest due date (EDD), shortest setup (SSU), etc., on their own, are unable to optimize amongst the contradicting needs of cycle time (i.e. average cycle time, avgCT, and standard deviation of cycle time, sdCT) and delivery accuracy (i.e. average tardiness, avgTARD, and standard deviation of tardiness, sdTARD).

In studies that consider multiobjective optimization, many converted multiple objectives into a single objective optimization problem using a weighted combination of rules where each rule typically addresses an objective of interest (Deb 2001). As the fab managers and engineers are better at pinpointing the preferred operating point(s) on the characteristics curve than providing a set of weights to obtain the desired scheduling decisions, this paper focuses on optimizing the weights used in the weighted rule automatically, in an on-line and user friendly manner using SIMGAD.

The paper is organized as follows: Section 2 presents the related works; Section 3 put forth the proposed SIMGAD methodology; Section 4 uses an illustrative example to illustrate the potential capability of SIMGAD; Section 5 discusses the potential application; and Section 6 concludes the paper with a discussion of future work.

2 RELATED WORKS

As dispatching rules are unable to adjust to the dynamics of the facility, it is possible that a rule or a set of rules that provide unsurpassed performance at one point in time may not continue to do so as the facility evolves. Hence, various techniques have been developed for the selection of dispatching rules. The selection process may be periodically triggered, event-driven, or a hybrid of both. These techniques include enumerative periodic selection (Wu and Wysk 1989), artificial/competitive neural networks (Min and Yih 2003), genetic algorithms (Cardon et al. 2000), discrete-event simulation (Kim et al. 2003), etc., or various combinations of the above.

There is hardly any single dispatching rule that dominates when more than one conflicting objectives are involved. Hence, emphasis is not only placed on the selection of rules but also on the application of rules simultaneously in the form of a weighted combination of rules to optimize multiple performance measures of interest under a factory's changing condition (Dabbas et al. 2003, Lin et al. 2005, Chiang et al 2006, and Sivakumar and Gupta 2006). The results from all the multiobjective scheduling studies reviewed in this paper have generally shown significant improvement over the use of single dispatching rule.

Dabbas et al. (2003) and Lin et al. (2005) used a mixed design of experiments (DOE), response surface methodology (RSM), and desirability function to determine the weights for the weighted combination of multiple criteria for semiconductor scheduling. However, weights are determined off-line and in a manual manner that requires user to have the expertise to fit meta-models to each response for example.

Chiang et al. (2006) uses genetic algorithm to search for the optimal combination of release and machine selecting rules, and the weights for five linearly combined dispatching rules, and proposes colored timed Petri nets with the queueing systems for performance evaluation and scheduling for wafer fabrication. However, no sequence-dependent setups are considered.

3 SIMULATION-BASED GENETIC ALGORITHM WITH DESIRABILITY FUNCTION

The overview of the proposed Simulation-based Genetic Algorithm with Desirability function (SIMGAD) is summarized in Figure 1 and the detailed framework used in SIMGAD is shown in Figure 2. The desirability function, discrete event simulation, and genetic algorithm used in SIMGAD are described in Sections 3.1, 3.2, and 3.3 respectively.

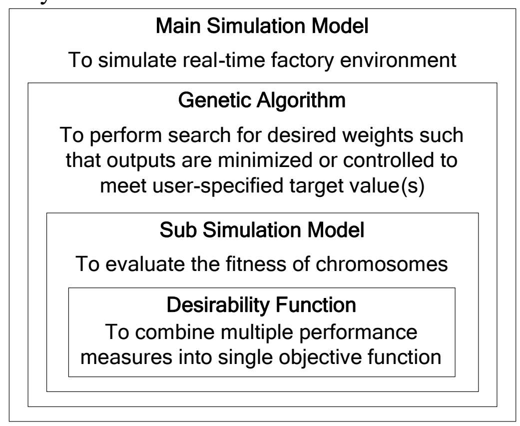

Figure 1: Overview of SIMGAD

3.1 Desirability Function

The desirability function approach is one of the most widely used methods in industry for dealing with the optimization of multiple-response problems (Pasandideh and Niaki 2006). The multiple responses (more commonly known as performance measures or objectives) are each expressed as a desirability function and combined into a single objective function. Depending on whether a particular response y_i is to be maximized, minimized, or assigned a target value, different desirability functions introduced by Derringer and Suich (1980) could be applied to transform each y_i into a desirability value $d_i(y_i)$, where $0 \le d_i(y_i) \le 1$.

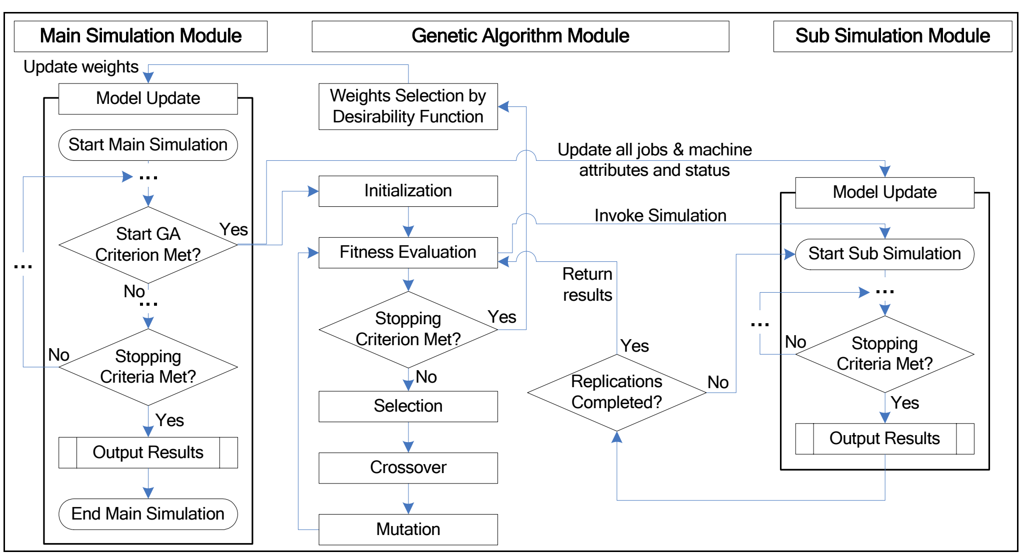

Figure 2: Detailed framework for SIMGAD

Desirability function for response with a target is given by

$$d_i(y_i) = \begin{cases} 0, & y_i \le l_i, \\ \left[\dfrac{y_i - l_i}{t_i - l_i}\right]^{\omega_1}, & l_i \le y_i \le t_i, \\ \left[\dfrac{y_i - u_i}{t_i - u_i}\right]^{\omega_2}, & t_i \le y_i \le u_i, \\ 0, & y_i \ge u_i, \end{cases} \tag{1}$$

where l_i, u_i, and t_i are the lower, upper, and target value of the response y_i respectively such that $l_i < y_i < u_i$, and the exponents ω_1 and ω_2 determine how strict the target value is desired. The overall (or total) desirability, D, is defined as the geometric mean of the individual desirability values:

$$D = \left(d_1(y_1) \times d_2(y_2) \times \ldots \times d_n(y_n)\right)^{1/n}, \tag{2}$$

where n denotes the number of responses.

3.2 Discrete-Event Simulation

Discrete-event simulation is used to simulate both the real-time factory (main simulation module) as well as for the evaluation of the weights used in the weighted rule (sub simulation module) since stochastic scheduling with multiobjectives and sequence-dependent setup (SDS) is analytically intractable. A model for the illustrative problem described in Section 4 is constructed and simulated using AutoSched™ Accelerated Processing (AutoSchedAP) v7.2 by Brooks Automation Inc. Some customizations of the dispatching rules used are necessary. The simulation models and rules are debugged and verified in a number of iterations using the 'trace' technique (Law 2007).

3.2.1 Dispatching Rules

As no dispatching rule has been shown to consistently produce better performance than all other rules under a variety of shop configurations and operating conditions (Blackstone et al. 1982), preliminary simulations have been carried out to identify the dispatching rule that is reasonably good for minimizing average cycle time (avgCT), standard deviation of cycle time (sdCT), average tardiness (avgTARD), and standard deviation of tardiness (sdTARD) respectively. For brevity, results are not presented. However, the dispatching rules are described below:

- Smallest SetUp Modified (SSU+) rule takes into account the relative cost of a particular setup by considering the potential runtime of available work in process that can make use of the setup on the machine. SSU+ is defined as $s_{i'i}/n_ip_i$ where $s_{i'i}$ is the setup time between lot i' that has been processed and lot i considered for processing next, n_i is the number of lots in the queue that could use the same setup required by lot i, and p_i is the expected average processing time of lot i. SSU+ is used to minimize avgCT.
- First-In, First-Out (FIFO) rule selects the lot with the earliest arrival time to the queue of the machine. FIFO is capable of minimizing sdCT.
- Critical Ratio (CR) rule selects the lot with the lowest critical ratio as defined by $(d_i - t_{now})/RPT_i$, where d_i is the due date of lot i, t_{now} is the simulated time at which the rule is applied, and RPT_i is the remaining mean processing time of lot i (for a single machine one operation system, $RPT_i = p_i$). CR can be used to minimize avgTARD and sdTARD in tandem.

The weighted rules made up of FIFO, CR, SSU+ is hereafter known as the MOFCS+ rule. Each of the criteria (presented in the form of dispatching rules) is normalized, weighted and combined into a single criterion. The individual rule index, $I_{R,i}$ is computed for each lot i in the queue at the time when machine is available to select the next lot, where R refers to the rules/criterion used. The minimum and maximum index for each rule is denoted as $I_{R,min}$ and $I_{R,max}$ respectively. The weighted sum of normalized index, I_i for each lot i is computed as follows:

$$I_i = \sum_{R=1}^{3}\left[w_R \frac{I_{R,\min} - I_{R,i}}{I_{R,\min} - I_{R,\max}}\right], \quad \sum_{R=1}^{3} w_R = 1 \tag{3}$$

where weight w_R reflect the relative importance of the criterion associated with the rule. The lot with the lowest weighted sum of normalized index is selected for processing next.

3.2.2 Sub Simulation Model

When genetic algorithm is invoked, for example, at main simulation model's simulated time t_{now}, a sub simulation is automatically generated and updated with the WIP present in the main simulation model at t_{now}. The sub simulation model uses the same lot arrival and processing time distribution as the main simulation model and will perform a five-replicates simulation that begins from time t_{now} and look-ahead in time to get the estimates of the steady state responses for the evaluation of each set of weights.

3.3 Genetic Algorithm

Genetic Algorithm (GA) is first developed by Holland (1975). The GA is a search algorithm based on the mechanism of natural selection and natural genetics. GAs have been extensively used as search and optimization tools in various problem domains due to their broad applicability, ease of use, and global perspective (Goldberg 1989). In GAs, each individual solution is represented in the form of a finite length string called a chromosome. A set of randomly generated chromosomes forms an initial population. Through the use of genetic operators, such as crossover and mutation, modifications to the chromosomes of randomly selected solutions are introduced in a systematic fashion to generate a new generation of solution alternatives, moving towards the optimization of certain criteria (Goldberg 1989). As generations increase, the optimal set of bit patterns (or schemata) will be found for the chromosome, from which the optimal weights are obtained.

In this study, we incorporate GAlib (Wall 1996), a highly customizable and well documented C++ library of GA objects, into AutoSchedAP as the library includes tools for facilitating the use of GA to perform optimization using any representation and genetic operators.

3.3.1 Encoding

For the problem under consideration, each chromosome (a candidate solution) comprises decision variables (refer to the weights used in MOFCS+ rule), x_i (i = 1,2,3) that are encoded into binary sub-strings of length L_i. For two-decimal-place precision, L_i is chosen to be the smallest integer satisfying $10^2\left(b_i - a_i\right) \le 2^{L_i} - 1$, where b_i and a_i refers to the upper and lower bound of variable x_i respectively. Each weight can take the value between zero and one, $b_i - a_i = 1$ for i = 1,2,3. As such, each variable x_i will be encoded as a binary sub-string of length seven and the total string length of each chromosome is therefore 21 bits. The first sub-string of seven bits encodes the weight to be used for the FIFO rule. The next and last sub-strings of seven bits encode the weights to be used for CR and SSU+ rules respectively.

3.3.2 Fitness Evaluation

The sub simulation model is used to evaluate the performance or fitness of each chromosome. The fitness measure is the overall (or total) desirability, D, given in (2) where the responses include avgCT, sdCT, avgTARD, and sdTARD. Due to the embedded random variations used in simulation, it is necessary to determine the length of simulation run and the number of simulation replications for the evaluation of one chromosome. In this study, five replications with simulation run length of 60 days are used.

3.3.3 Selection and Population Replacement

The selection process consists of determining from which chromosomes the next population would be generated. This study applies a tournament selector that uses the roulette wheel method to select two individuals and then picks the one with the higher fitness.

Since it is possible that the fittest chromosomes of the population may fail to produce offspring that is at least as fit as itself in the next generation, to avoid fitter chromosomes from being lost in the evolutionary process, an elitist strategy is adopted to ensure there are overlapping generations. The commonly known (μ+λ) selection strategy is applied in which the μ parents and λ offspring compete for survival and the μ best of offspring and old parents are selected as parents of the next generation. In this study, $\mu = \lambda$ is used.

Apart from the elitist strategy used from one generation to another, a global replacement strategy has also been introduced to bring the best individual from one GA run to another GA run. Apart from the best individual, the remaining individuals are randomly generated at the beginning of each GA run (except for the first run where equal weights are applied).

3.3.4 Genetic Operation

A genetic operation refers to the generation of an offspring (child) from the selected chromosomes using the crossover and mutation operators. In this study, two-point crossover is applied in which two positions are chosen uniformly at random and the segments between them exchanged. Crossover is not applied to all pairs of individuals selected for mating. A random choice is made according to a crossover probability (one of the GA parameters).

The mutation operator defines the procedure for mutating each chromosome. If the mutation probability is small (such that less than one bit requires a flip test), the flip test is performed on each bit based on the mutation probability. Otherwise, for streamlined execution as suggested in GAlib (Wall 1996), the known number of bits is mutated based on the mutation probability.

3.3.5 Parameter Optimization

As this is a preliminary study to evaluate SIMGAD, we circumvent the need to spend hours of computation time on fine parameter tuning by referencing existing rule of thumb and then choosing the GA parameters based on a few trial simulation runs. Population size, number of generation, crossover probability and mutation probability are fixed at 20, 10, 0.6, and 0.14 respectively for this study.

3.4 Reactive Optimization of Weights

The optimization of weights can be performed periodically (time-driven), upon the occurrence of events (event-driven), or under a hybrid of both depending on the dynamism of the factory and how responsive it must be to the state of the system. Apart from the change of dispatching rules, performance objectives, and user preferences that highly necessitates the re-optimization of weights, it is not known (not available in the literature to the best of our knowledge) what degree of disruptions and factory changes, or what kind of disruptions or unexpected events would invalidate the weights. In this study, re-optimization of weights is carried out after every 500 lots and 1000 lots.

4 AN ILLUSTRATIVE EXAMPLE

This study involves scheduling a dynamic and stochastic serial (i.e. single lot processing) ion implanter machine with SDS. The assumptions considered are:

- Operators, handlers, and hardware are not explicitly modeled.
- No yield losses are considered.
- No rework is modeled.
- Preemptions are not allowed.
- No machine breakdowns are modeled.
- All lots hold equal weights.

In this study, three lot types (namely T1, T2, and T3) are considered and released to the machine in proportional amounts (product mix of 1:1:1) to promote greater probability of machine setups. The processing times are lot-type dependent and the expected average processing times for T1, T2, and T3 jobs are 45, 50, and 55 minutes respectively. The lot processing times in both sets are perturbed using a uniformly distributed interval of U(-5,5) minutes. The perturbation is included to account for machine and/or operator efficiencies, yield losses, and loading and unloading variability.

Lots are assumed to arrive dynamically to the machine to maintain a Constant WIP of size six (denoted as CW6). The proportion of average setup time to average processing time is set at 20% which according to Kim and Bobrowski (1997), is comparable to the situation of a real job-shop. Under this setting, same sequence-dependent setup time of 10 minutes is applied when the type of the lot that has been processed is different from the type of the next lot to be processed. No setup is required to process consecutive lots of the same type.

The due date of lot i, d_i is determined by assigning a random flow allowance to each lot upon its arrival to generally represent the actual environment in which due dates are requested by the customer (Weng and Ren 2006) and is defined as $d_i = a_i + (1 + u\delta)\, p_i$, where a_i is the arrival time of lot i to the queue of the machine, $u \in U(0,1)$, δ is the due date tightness factor, and p_i is the expected average processing time of lot i. In this study, due date is set parallel to the loose due date setting used in Kim and Bobrowski (1997). The δ value is adjusted based on 10 simulation runs to result in approximately 20% tardy lots for the FIFO dispatching rule and found to take the value of 29.

Since we do not have any historical data based on our illustrative example described above, a randomly selected set of weights ($w_{FIFO} = 0$, $w_{CR} = 0.67$, $w_{SSU+} = 0.33$) used in (3) is simulated for 20 replications to produce a set of performance measures (with appropriate warmup) which we will use to represent user specified target values.

Table 1 shows the results of simulation with fixed weights (based on 20 replications), SIMGAD-500 and SIMGAD-1000 (based on one problem instance of 5000 lots each) where 500 and 1000 refers to the number of lots processed, based on which the re-optimization of weights is carried out. While the results of the SIMGAD-500 and SIMGAD-1000 is not exactly close to the user specified target values, it is not dominated by any of the tested sets of weights (fixed throughout the simulation) as shown in Table 2 (refer to Appendix A). It should be noted that these sets of weights are comprehensive enough for using the mixed design of experiment (DOE), response surface methodology (RSM) and desirability function to determine the optimal or desired weights in an off-line manner as described in Dabbas et al. (2003) and Lin et al. (2005). The preliminary results are reasonably good considering the GA parameters are not yet optimized.

Table 1: Results of simulation with fixed weights (user specified target is based on 20 replications), SIMGAD-500 and SIMGAD-1000 (based on one problem instance each)

Method	avgCT	sdCT	avgTARD	sdTARD
User Specified Target	327.079	291.424	0.699	8.596
SIMGAD-500	324.505	278.041	1.718	15.061
SIMGAD-1000	324.626	283.070	1.370	12.758

5 APPLICATION

In general, it takes approximately nine seconds for the evaluation of one chromosome (involving five simulation replications). Hence, one GA run with population size of 20 and 10 generations takes approximately 30 minutes (i.e. 9/60 minutes x 20 x 10). This implies that SIMGAD could potentially be incorporated into the scheduler and applied on-line to provide real-time optimization of the weights based on updated factory condition with weights re-optimization carried out after each lot is processed if the expected mean processing times for each lot is approximately 30 minutes or more. In fact, the time can be shortened easily through parallelization of the GA.

6 CONCLUSION

This paper presented the concepts, development and potential application of simulation-based genetic algorithm with desirability function (SIMGAD) for determining the weights to schedule an ion-implanter subjected to multiple conflicting objectives and sequence-dependent setups in semiconductor manufacturing. The initial results show considerable promise for the SIMGAD to be applied online, in an user-friendly manner to generate weights for the weighted rule automatically. With the use of SIMGAD, the user now has the capability to control the trade-off between the objectives. As with any new methods, continual improvement and refinement of this methodology is essential and is on-going. Further work includes GA parameter optimization, analysis of the effects of disruptions and disturbances on the rate of weights optimization, and carrying out computation experiments based on real factory data for various machine configurations.

ACKNOWLEDGMENTS

Adeline T. H. Ang is financially supported by A*STAR (Agency for Science, Technology, and Research) under the A*STAR Graduate Fellowship Scheme.

A APPENDIX A: SETS OF WEIGHTS USED

Table 2 lists the other 15 sets of weights tested over the problem configuration described in Section 4.

Table 2: Weights used for the MOFCS+ rule

Set ID of Weights	w_{FIFO}	w_{CR}	w_{SSU+}
1	0.000	0.333	0.667
2	0.000	0.500	0.500
3	0.000	0.667	0.333
4	0.167	0.417	0.417
5	0.167	0.667	0.167
6	0.167	0.167	0.667
7	0.333	0.333	0.333
8	0.333	0.667	0.000
9	0.333	0.000	0.667
10	0.417	0.167	0.417
11	0.417	0.417	0.167
12	0.500	0.000	0.500
13	0.500	0.500	0.000
14	0.667	0.167	0.167
15	0.667	0.000	0.333
16	0.667	0.333	0.000

REFERENCES

Baker, K. R. 1974. *Introduction to sequencing and scheduling*. New York: Wiley.

Blackstone, J. H., D. T. Phillips, and G. L. Hogg. 1982. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research* 20:27-45.

Cardon, A., T. Galinho, and J.-P. Vacher. 2000. Genetic algorithms using multi-objectives in a multi-Agent System. *Robotics and Autonomous Systems* 33:179-190.

Chiang, T.-C., A.-C. Huang, and L.-C. Fu. 2006. Modeling, scheduling, and performance evaluation for wafer fabrication: A queueing colored petri-net and GA-based approach. *IEEE Transactions on Automation Science and Engineering* 3(3):330-337.

Dabbas, R., J. W. Fowler, D. A. Rollier, and D. McCarvilles. 2003. Multiple response optimization using mixture-designed experiments and desirability functions in semiconductor scheduling. *International Journal of Production Research* 41(5):939-961.

Deb, K. 2001. *Multi-Objective Optimization using evolutionary algorithms*. New York: John Wiley & Sons.

Derringer, G., and R. Suich. 1980. Simultaneous optimization of several response variables. *Journal of Quality Technology* 12:214-219.

Goldberg, D. E. 1989. *Genetic algorithms in search, optimization & machine learning*. Reading. Mass.: Addison-Wesley Pub. Co.

Holland, J. 1975. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.

Kim, S. C., and P. M. Bobrowski. 1997. Scheduling jobs with uncertain setup times and sequence dependency. *Omega*, *Int. J. Mgmt Sci* 25(4): 437-447.

Kim, Y.-D., S.-O. Shim, B. Choi, and H. Hwang. 2003. Simplification methods for accelerating simulation-based real-time scheduling in a semiconductor wafer fabrication facility. *IEEE Transactions on Semiconductor Manufacturing* 16(2):290-298.

Law, A. M., 2007. *Simulation modeling and analysis*. 4th Ed. Boston: McGraw-Hill.

Lin, J. T., F. K. Wang, and P. C. Kuo. 2005. A parameterized-dispatching rule for a Logic IC sort in a wafer fabrication. *Production Planning & Control* 16(5): 426-436.

Min, H. S., and Y. Yih. 2003. Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system. *International Journal of Production Research* 41(16): 3921-3941.

Pasandideh, S. H. R., and S. T. A. Niaki. 2006. Multi-response simulation optimization using genetic algorithm within desirability function framework. *Applied Mathematics and Computation* 175:366-382.

Sivakumar, A. I., and A. K. Gupta. 2006. Online multiobjective pareto optimal dynamic scheduling of semiconductor back-end using conjunctive simulated scheduling. *IEEE Transactions on Electronics Packaging Manufacturing* 29(2):99-109.

Wall, M. 1996. GAlib: A C++ Library of Genetic Algorithm Components. Version 2.4. Revision B. Massachusetts Institute of Technology <http://lancet.mit.edu/ga/>.

Weng, M. X., and H. Ren. 2006. An efficient priority rule for scheduling job shops to minimize mean tardiness. *IIE Transactions* 38:789-795.

Wu, S.-Y. D., and R. A. Wysk. 1989. An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing. *International Journal of Production Research* 27:1603-1623.

Yang, T., Y. Kuo, and C. Cho. 2007. A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem. *European Journal of Operational Research* 176:1859-1873.

Zhu, X, and W. E. Wilhelm. 2006. Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE Transactions* 38:987-1007.

AUTHOR BIOGRAPHIES

ADELINE T. H. ANG is a Ph.D. research student in the School of Mechanical and Aerospace Engineering at Nanyang Technological University (NTU), Singapore. She obtained her bachelor degree in Mechanical Engineering (1st class Hons.) from National University of Singapore (NUS). Her research focuses on on-line, multiobjective dynamic scheduling of semiconductor manufacturing. Her email address is <adelath@pmail.ntu.edu.sg>.

APPA IYER SIVAKUMAR is an Associate Professor in the School of Mechanical and Aerospace Engineering (MAE) at the Nanyang Technological University, Singapore and a Faculty Fellow of Singapore - Massachusetts Institute of Technology (MIT) Alliance (SMA-MST programme). He was at Gintic Institute of Manufacturing Technology, Singapore prior to this appointment. His research interests are in the area of OR, advanced Manufacturing Systems engineering, discrete event Simulation, Scheduling, Logistics, Supply chain design, and Research Methodology. He has many years of industrial and research experience in the UK prior to post in Singapore in 1993. He held various senior positions including technical manager and project manager for a number of years at Lucas Systems and Engineering and Lucas Automotive, UK. During this period he was responsible for manufacturing re-engineering projects and involved in the introduction of IT and Scheduling systems. He received a Bachelors of Engineering in Manufacturing Systems Engineering and a PhD in Manufacturing Systems Engineering from University of Bradford, UK. He has made many contributions at international conferences and journals. He has trained and supervised a number of PhD students and many Master's level students. He has been the technical chair and co-edited 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>.