Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

THE IMPACT OF OPERATION-TO-TOOL DEDICATIONS ON FACTORY STABILITY

James P. Ignizio

University of Texas – Pan American
1201 West University Drive
Edinburg, Texas 78539-2999, USA

ABSTRACT

It is essential, or should be, that the pricey machines that support the process steps in the fabrication of semiconductor wafers be designed so as to achieve optimal, or near optimal, performance. Performance, in turn, is typically measured by average cycle time, capacity, yield, and cost. One metric of particular importance is, however, seldom considered. This is that of the *stability* of the machines, workstations, and production line as a whole. While too often ignored, it is vital that the components of the factory exhibit stable performance when exposed to everyday changes (e.g., minor fluctuations in product mix, slight changes in factory throughput). We examine, herein, the stability of the *reentrant* workstations that employ operation-to-machine dedications when those dedications are produced by either (a) optimization, (b) heuristics, or (c) genetic algorithms. The results of a multi-year effort reveal there is a significant difference in the stability of the resultant facility.

1 INTRODUCTION

The phone's ringing interrupts yet another nightmare. In this one Jack is standing alone on the beach, staring helplessly out to sea as a giant wave approaches.

Rubbing the sleep from his eyes, Jack picks up the phone. A quick glance at the bedside clock reveals that is not quite yet 3am.

"Hello, Pete," Jack says, stifling a yawn. "What's the problem this time?"

"Jack," Pete sputters, "get down here fast! It's bad, real bad! We expect a massive WIP bubble to hit with the next 12 hours!"

So, thinks Jack, what else is new? "I'll be there as soon as I can", he answers. "Just try to keep calm."

Hanging up the phone, Jack turns to see that Linda, his wife, is sitting up in bed – and giving him "that look."

"I'm sorry, honey," says Jack, "Pete, the night shift supervisor, claims we've got another WIP bubble heading toward our front-end lithography tool set. They need me, and I'm not sure just when I'll be getting back."

"Jack," says Linda, "is this new job worth it? They keep calling you in the middle of the night, and you never seem to get home until late evening. I'm beginning to believe that it was a mistake to accept the position at the Muddle Corporation – no matter what they promised you."

"I know, I know. But let's talk about this later," Jack answers, as he hurriedly ties his shoes.

"Let's do that. Jack, let's talk about it tonight. Tonight, Jack. Promise me that."

"I promise," says Jack unenthusiastically, "but now I've really got to go. I'll call you as soon as I get a chance." Jack, now fully clothed, reaches for the bedroom doorknob.

"Jack, just what in the world is a 'WIP bubble,' and why do they always seem to happen at night?"

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

*****

The characters in the vignette presented above are purely fictional, as is the thoroughly muddled, Muddle Corporation (for more about the trials and tribulations of the beleaguered employees of the Muddle Corporation, see Ignizio, 2009a). The story itself may, however, sound frightfully real to anyone who has ever worked at most any semiconductor manufacturing firm. WIP bubbles, in the form of a much higher than anticipated level of wafers approaching a given tool set (i.e., workstation), seem an all too common an experience. The same may be said for a host of other unwelcome and unexpected events that occur … uncommonly often.

There are a multitude of reasons for performance problems in semiconductor fabs but, based on the author's experience (in the roles of both an employee in, and consultant to the semiconductor industry), one primary cause is that of the operation-to-machine (a.k.a., process step-to-tool) allocation scheme employed. This is particularly problematic with respect to the operation-to-machine allocations (a.k.a., qualifications, or dedications) of the highly reentrant lithography and implant workstations (Ignizio, 2009a and 2009b).

More specifically, the determination of just which machines in a given workstation should be qualified for the support of a specific subset of operations serves to determine the effective capacity, per process step, of the workstation. The determination of these allocations (a.k.a., dedications, qualifications) is a delicate balancing act that, if not done properly, will result in long queues and the increased likelihood of WIP bubbles. The problem itself may be represented as a combinatorial optimization model of massive complexity (Ignizio and Cavalier, 1994).

For example, given *M* machines and *O* operations, the total number of all possible operation-to-machine allocation schemes is given as:

$$Number\, of\ Schemes = 2^{M*O} \tag{1}$$

Thus, in a factory consisting of, say, 16 photolithography machines within a given "litho" workstation, as intended for the support of 12 operations (e.g., "layers"), the number of operation-to-machine qualification schemes would be:

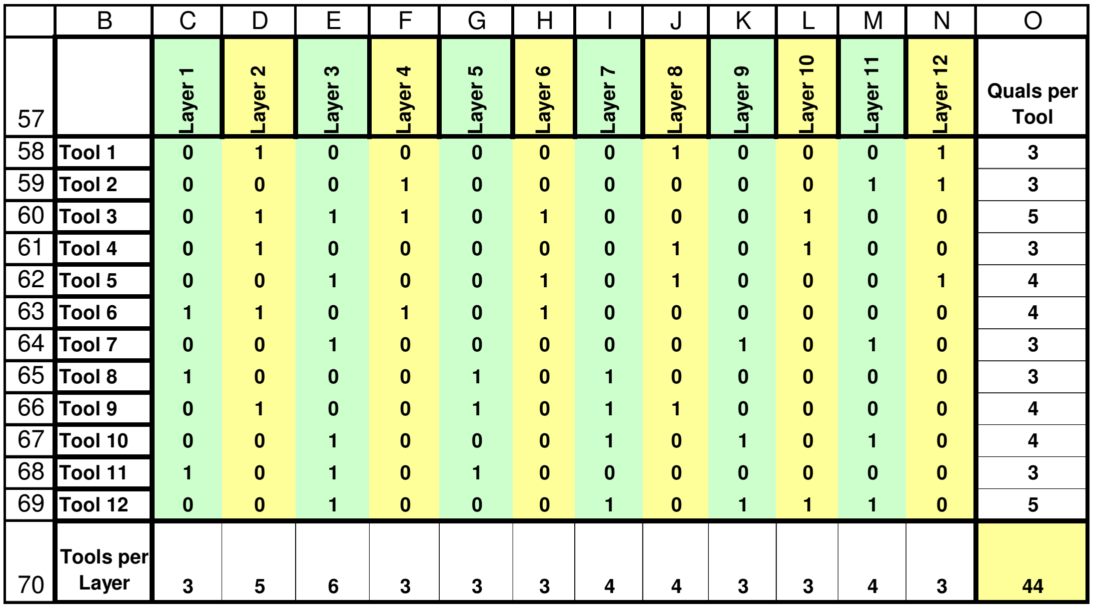
- Number of Boolean (0/1) Variables (i.e., $M*O$) = 12*16 = 192
- Number of Photolithography Layer Qualification Schemes = 2^{192}

or roughly 6.28e+57 different schemes!

Consequently, even if we employed the world's fastest computer (e.g., one running on the order of 35,600 gigaflops; i.e., 35,600 billion floating point operations per second), and even if that supercomputer could evaluate each allocation scheme via just a single floating point operation (in reality, of course, it would take far more operations), it would still take more than 5.6 x10^{36} years to evaluate all possible allocations. Clearly, the determination of machine-to-operation qualification schemes is hardly a trivial problem.

The number of constraints that must be employed to fully describe the problem further increases its complexity. For example, any allocation schemes that do not provide support for all operations must be immediately ruled out. Or there may be operation-to-operation constraints. For example, operation *X* and operation *Y* may have to always be performed on the same machine (e.g., 'lot to lens' dedication in photolithography) or perhaps operations *A* and *B* should *never* be performed on the same machine (e.g., due to incompatibilities in the photoresist materials used). Furthermore, due to the restrictions of the plumbing system in a lithography machine, there will be a practical limit to the total number of operations that any single machine may support. Finally, for purposes of redundancy it is common to make sure that each operation be supported by more than one machine.

Despite the combinatorial complexity of the problem it may be formulated and then solved by means of commercial optimization software (e.g., CPLEX) where the objective is to maximize workstation capacity. The specific form of the mathematical model is provided in the Appendix (Ignizio, 2009a and b). An example of a solution to the problem is depicted in Figure 1.



	B	C	D	E	F	G	H	I	J	K	L	M	N	O
57		Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7	Layer 8	Layer 9	Layer 10	Layer 11	Layer 12	Quals per Tool
58	Tool 1	0	1	0	0	0	0	0	1	0	0	0	1	3
59	Tool 2	0	0	0	1	0	0	0	0	0	0	1	1	3
60	Tool 3	0	1	1	1	0	1	0	0	0	1	0	0	5
61	Tool 4	0	1	0	0	0	0	0	1	0	1	0	0	3
62	Tool 5	0	0	1	0	0	1	0	1	0	0	0	1	4
63	Tool 6	1	1	0	1	0	1	0	0	0	0	0	0	4
64	Tool 7	0	0	1	0	0	0	0	0	1	0	1	0	3
65	Tool 8	1	0	0	0	1	0	1	0	0	0	0	0	3
66	Tool 9	0	1	0	0	1	0	1	1	0	0	0	0	4
67	Tool 10	0	0	1	0	0	0	1	0	1	0	1	0	4
68	Tool 11	1	0	1	0	1	0	0	0	0	0	0	0	3
69	Tool 12	0	0	1	0	0	0	1	0	1	1	1	0	5
70	Tools per Layer	3	5	6	3	3	3	4	4	3	3	4	3	44

Figure 1: An Illustration of Operation-to-Machine Dedication Solution

In figure 1 the operation-to-machine (i.e., layer-to-tool) dedications for a 12 machine workstation that must support 12 operations (i.e., "layers") is given. The intersection of a row labeled "tool" and a column labeled "layer" is either a zero or a one. A "zero" indicates the associated tool is not qualified for the associated layer whereas a "one" states that the layer may be processed on that tool. Also depicted, for purposes of information, are the number of tools that support each layer (e.g., five tools support layer 2) and the number of qualifications associated with each tool (e.g., tool 1 is qualified to support three different layers). The solution (i.e., the 0-1 assignments for this situation) was determined by means of solving the corresponding optimization model.

2 SOLUTION STABILITY

Based on this author's experience, as well as discussions with personnel from a number of semiconductor firms, the most common way that factory personnel derive a solution to the operation-to-machine allocation problem would appear to be manually assign dedications by means of various heuristics, as based on "experience and insight." (In several instances individuals responsible for determining the operation-to-machine dedications confided in the author that they employed "educated guesses.") Once a heuristic dedication scheme is developed its performance is typically evaluated via simulation.

A second, more formal and rigorous, approach to the derivation of an operation-to-machine solution is by means of the employment of optimization software; i.e., finding the optimal solution to a mathematical model, as shown in general form in the Appendix. Solutions derived in this manner are also most generally evaluated by means of a subsequent simulation exercise.

It is to be stressed, however, that the evaluations via simulation – when performed – do not consider the stability of the solution. Instead they typically focus on the resultant capacity of the workstation. In other words, minor fluctuations in the fab environment (i.e., the impact of slight changes in product mix, throughput rates, volume of priority lots, rework, etc., are not investigated). This is not surprising since the number of simulations that would be required to perform such analyses would be so large as to be considered impractical – and/or perhaps unwarranted.

A third approach to the problem, evidently seldom if ever employed in practice, is to solve for the operation-to-machine dedications by means of finding a solution to the optimization model (again, as shown in general form in the Appendix) *via the employment of evolutionary programming*; i.e., genetic algorithms. While a genetic algorithm typically develops a "very good" solution, it cannot guarantee optimality (Ignizio, 1991). It should also be noted that the employment of genetic algorithms requires certain decisions as to such matters as "cross-over rates," "mutation rate," and convergence properties. In other words, two individuals may apply genetic algorithms to the same, identical problem and mathematical model and come up with different dedication schemes.

An multi-year examination of each of the three approaches listed above, when applied to a detailed simulation model of a typical semiconductor fab lithography or implant workstation, provided the motivation for the development of the research conducted in support of this paper. Specifically, it was decided to compare the *stability* of the solutions achieved by each of the three methods. More specifically, each of the three methods (i.e., heuristics, optimization, and genetic algorithms) was first used to determine operation-to-machine dedications. Next the performance of the resultant designs, after seemingly minor changes were made, were evaluated for their stability by means of detailed simulation models.

We define, herein, *stability* as *the ability of a solution to maintain acceptable performance in the face of random fluctuations in the system* (Ignizio, 1998). In the semiconductor manufacturing problem we attempted to determine the impact on the performance of a given operation-to-machine dedication scheme in the face of random changes (as limited to small perturbations) in such common factory parameters as:

- changes in product mix
- ramping (i.e., the decision to increase or decrease the throughput rate of the products produced)
- random fluctuations in the variability of jobs arriving at the workstation
- random fluctuations in the times consumed by both scheduled and unscheduled maintenance events
- random fluctuations in the amount of rework imposed on the workstation
- random fluctuations in the volume of the priority lots supported by the workstation

The results, in general terms, as developed via literally hundreds of time consuming experiments may be summarized as follows:

- Optimization provided dedication schemes that were, as would be expected, invariably superior to its alternatives in terms of common metrics such as capacity and cycle time – but which also exhibited significant instability when small changes in the environment were experienced.
- The performance of the workstation, when the dedication scheme was accomplished heuristically, was considerably less impressive than that achieved via optimization and the solutions were found to be unstable when small changes to the factory's environment were experienced.
- The solutions achieved by means of the application of genetic algorithms to the mathematical model indicated in the Appendix were only slightly less effective than those achieved via true optimization. More importantly, however, was the fact that the solutions derived via genetic algorithms were – in virtually every instance – by far the most stable of any of the approaches. (Once again, however, it is to be stressed that the solutions derived by genetic algorithms are highly dependent on the choice of operational parameters – e.g., cross-over rates, mutation rates – and convergence determinations. In this effort an evaluation was first made of those parameters before deciding on the parameter settings to be employed by the genetic algorithm.)

Based on the experiments run, it was found that the single most effective indicator of instability was that of the degradation of workstation cycle time efficiency. Cycle time efficiency, designated as "CTE," is defined as the ratio of process time to cycle time – where, the higher the value of CTE, the better (Ignizio, 2009a). In this effort the raw process time (i.e., minimum possible process time) of the workstation was divided by its actual cycle time. Thus, whenever there is a degradation in cycle time efficiency, there is an increase in the average time taken by the workstation to process a wafer. This increase in cycle time, in turn, results in a less effective and efficient factory – one with increased waiting times (e.g., queues) and increased inventory levels.

Table 1 lists averages of the results, over 25 scenarios, achieved when examining the impact on cycle time efficiency of seemingly minor changes in the *product mix* that the workstation was to support. The fluctuations in this instance were on the order of one or two percent (for example, given two products a 50-50 product mix would be changed to a 49-51 and 48-52 mix). The dedications were first determined via the three methods (optimization, heuristics, and genetic algorithms) and then evaluated for the fluctuations in product mix via a detailed simulation model of the workstation.

Table 1: Comparison of Cycle Time Efficiencies subject to fluctuations in product mix

	CTE – *before* perturbations	**CTE – *after* perturbations**	**Degradation in percent**
Optimization	52 percent	29 percent	44 percent
Heuristic method	32 percent	22 percent	31 percent
Genetic Algorithms	49 percent	45 percent	8 percent

The table clearly indicates the robustness of the solutions obtained by means of genetic algorithms – and the fact that these solutions are quite close to those achieved via formal optimization procedures. It also shows the surprising level of degradation in the performance of solutions derived by either optimization or heuristic means.

3 OBSERVATIONS

Similar results as those noted in Table 1 were noted when minor changes were made to the other parameters (e.g., throughput rate, ramping, variability of arrivals, and time to conduct maintenance events). ***That is, it was found that the solutions achieved via genetic algorithms were close to those produced by optimization, and much more stable than solutions found by either optimization or heuristic means***.

While the finite number of experiments performed do not constitute a rigorous proof of the superiority, in terms of stability, of genetic algorithms they certainly indicate that this is a strong possibility. And they certainly indicate that there is a strong likelihood of significant differences in the inherent stability of dedications derived by various means.

One of the more interesting matters observed during the subject effort was the importance of the choice of parameters (e.g., cross-over rate, mutation rate, and convergence properties) employed by the genetic algorithm. More specifically, genetic algorithms that fit a certain profile produce, by far, the most stable and most close to optimal solutions.

4 SUMMARY AND CONCLUSIONS

The author is convinced that the role of stability in the performance of semiconductor fab workstations, and their machines, should be considered. The experiments conducted over several years, and reported in summary form herein, indicate that the solutions to problems of operation-to-machine dedications as derived by means of heuristics, optimization, and genetic algorithms, differ – often substantially – in terms of their inherent stability. It was shown, for example, that seemingly minor changes in such factors as product mix, throughput rates, variability of wafer arrival rates, maintenance times, rework, and the volume of priority lots may have an unexpectedly significant impact on performance. This impact is, however, much less when genetic algorithms are employed in the dedication procedure.

Further research in this area is underway. One of the matters under investigation is the determination of an approach for the determination of stability that would require less in the way of time and effort (e.g., a reduction in the massive number of simulations, optimizations, and solution derivations via genetic algorithms) than was employed herein. This would, we believe, involve the determination of a simpler, proxy metric that may be used to more efficiently determine system stability. The results of this effort will be reported in a future paper.

A APPENDIX

The most basic form of the Boolean Optimization model that serves to define the operation-to-machine qualification problem is presented below. It is assumed that the problem involves jobs (e.g., semiconductor wafers) that arrive at a photolithography workstation in the form of lots (e.g., 25 wafers per lot) and that the time period of interest is a week (168 hour work week). Note in particular that the primary objective is to balance the loadings across all machines in the workstation, a concept first employed by Sorenson at the Ford Model T factory (Ignizio, 2009a).

Just as Sorenson's team discovered that a balanced loading reduced cycle time, a balanced loading in the workstations of a semiconductor fab similarly serves to optimize overall factory performance. A more detailed discussion of the optimization model, and its extensions, may be found in the references (Ignizio, 2009a and b).

Definitions:

r(i, j, k) = a Boolean variable, where r(i, j, k) is 1 if operation k is performed on machine *i* of lot *j* during the week, and is 0 otherwise (alternately, this variable could be used to represent the number of operation type *k* performed on machine *i* of a *cascade* of lots)

y(i, k) = a Boolean variable; where y(i, k) is 1 if operation *k* is qualified on machine *i*; and 0 otherwise

x(i, r) = a Boolean variable; where x(i, r) is 1 if machine *i* uses photoresist *r*; and 0 otherwise

θ(r) = the set of operations (photolithography layers) that require photoresist of type *r*

$R_{max}(i)$ = the maximum number of photoresists that may be allocated to machine *i*

a(i, k) = time required (in hours) for performance of operation *k* per lot on machine *i*. Note that this includes the additional average time for rework, test, and setups. (Alternately, this could be the time, in hours, required for the performance of operation *k* on machine *i* for a *cascade* of lots.)

TB(i) = time available, per week, for performance of any and all operations assigned to machine *i*. (Note, this is simply 168 hours per week times the availability of the machine minus any time required for other supportive work).

T(k) = the time that must be made available, each week, for the conduct of operation *k*, including additional time for rework at the operation (this, in turn, is a function of the desired throughput of the factory).

gap = minimum gap across all of the machines (note that the gap is defined as the difference between the time available on the machine and the time consumed by the operations performed by the machine)

λ = a small multiplier (e.g., 0.0001 in our case) used to control the values of x(i,r) – i.e., used in the support of the transformation of a nonlinear function into a linear function.

M = a large multiplier (e.g., 1000 in our case) used in the support of the transformation of a nonlinear function into a linear function.

k = 1, … K **i =** 1, m **j =** 1, …, n **r =** 1,…,R

Formulation:

$$\textit{Maximize } \{gap - \lambda \bullet x(i,r)\} \quad (2)$$

subject to:

$$\sum_{k=1}^{K}\sum_{j=1}^{n}\{r(i,j,k)\ a(i,\ k)\} + gap < TB(i) \qquad \forall \mathrm{i} \tag{3}$$

$$y(i,k)\bullet M \geq \sum_{j=1}^{n}\{r(i,j,k)\} \qquad \forall i,k \tag{4}$$

$$\sum_{i=1}^{m} y(i,k) \geq 2 \qquad \textit{for all k} \tag{5}$$

$$\sum_{k=1}^{K} y(i,k) \leq max \quad \textit{number of operations on machine i} \qquad \forall i \tag{6}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} r\{i,j,k\}\bullet a(i,k) \geq T(k) \qquad \forall k \tag{7}$$

$$\sum_{i=1}^{m} r\{i,j,k\} = 1 \qquad \forall j,k \tag{8}$$

$$\sum_{k\in\theta(r)} y(i,k) \leq x(i,r)\bullet M \qquad \forall i,r \tag{9}$$

$$\sum_{r=1}^{R} x(i,r) \leq R_{\max}(i) \tag{10}$$

where x(i,r) and y(i,k) are 0-1 (i.e., Boolean) variables

Plus any necessary practical considerations such as:

$y(i,5) + y(i,7) \leq 1$ i.e., operation 5 and 7 cannot both be performed on machine i

$y(3,3) - y(3,9) = 0$ i.e., if operation 3 is performed on machine 3, then operation 9 must be performed on machine 3. All other functions in the model are described in Table 2.

Table 2: Model Components Definition

Function	Description
2	**Objective Function**: We seek to maximize the minimum gap across the photolithography tool set; i.e., balance the workload across the set of photolithography tools so as to minimize factory variability. Subtracted from this is the number of photoresists across the tool set multiplied by some small number (e.g., so as to set the number of photoresists per tool to zero unless absolutely required to support the constraint set).
3	**Constraint**: Limit the time devoted to all the operations on a given tool, for the week, to less than the total time available on that tool.
4	**Constraint**: Assures a qualification is allocated to a tool *only if necessary*.
5	**Constraint**: Assures that at least two tools are qualified for every layer so as to maintain redundancy.
6	**Constraint**: Limits the maximum number of qualifications on each tool.
7	**Constraint**: Requires that the time devoted to the layers on the lots equals or exceeds the minimum time required for that week.
8	**Constraint**: Assures that every layer of every lot is supported.
9	**Constraint:** Assures that the variable x(i,r) is set to a value of one if and only if this is necessary to satisfy other constraints.
10	**Constraint**: The total number of photoresists employed by each tool must be less than its maximum photoresist capacity.
Others	A wide variety of other constraints (e.g., a limitation on the minimum or maximum number of tools qualified per layer, or layers qualified per tool) may be appended to the formulation if deemed necessary.

REFERENCES

Ignizio, J. P. 2009a. <Optimizing Factory Performance>. New York, McGraw-Hill.

Ignizio, J. P. 2009b. Cycle time reduction via machine-to-operation qualification. *International Journal of Production Research*, Vol. 47, No. 24:6899-6906.

Ignizio, J.P. 1998. Integrating, Cost, Effectiveness, and Stability, *Acquisition Review Quarterly*, Vol. 5, No. 1: 51-60

Ignizio, J. P. and T. M. Cavalier. 1994. *Linear Programming*, Englewood Cliffs, New Jersey, Prentice-Hall.

Ignizio, J.P. 1991. *Introduction to Expert Systems*, New York: McGraw-Hill, 1991.

AUTHOR BIOGRAPHY

JAMES IGNIZIO is the Louis A. Beecherl, Jr. Professor of Engineering in the College of Engineering at the University of Texas – Pan American. Previously he held the positions of Staff Scientist at the Intel Corporation, Professor and Chair at the University of Virginia, Professor and Chair at the University of Houston, Professor at the Pennsylvania State University, and as a Program Manager on the Apollo manned moon-landing mission. He received the Ph.D. in Industrial and Systems Engineering from Virginia Tech. He is the author of 9 books and more than 350 papers, including more than 150 in various peer-reviewed professional journals. Dr. Ignizio is a Fellow of IIE; a Fellow of ORS, and a Fellow of WAPS. His research interests are in the application of Operations Research and Artificial Intelligence to real world problems, including those in the military and semiconductor manufacturing sectors. His email address is <igniziojp@utpa.edu>.