

**TECHNOLOGY THAT UPSETS THE SOCIAL ORDER – A PARADIGM SHIFT IN ASSIGNING LOTS TO TOOLS IN A WAFER FABRICATOR – THE TRANSITION FROM RULES TO OPTIMIZATION**

Ken Fordyce  
 IBM  
 Essex Junction, VT USA

Robert Bixby  
 ILOG  
 Mountain View, CA USA

Richard Burda  
 IBM  
 East Fishkill, NY USA

**ABSTRACT**

Historically the dominant decision technology to make dispatch decisions was “rules” which involves the following basic computational mechanisms: merge, select, sort, and if/then/else in a decision tree. Although rules do a reasonable job they fundamentally lack a robust ability to: (a) look across time, (b) look across tools at a tool set, (c) create an anticipated sequence of events at a tool set over some time horizon, (d) establish a formal metric and (f) search alternatives. However, standard wisdom was the rapid pace of change and short time interval between dispatch decisions precluded the use of optimization to build dispatch applications. Although this barrier was legitimate in the 1980s and most of the 1990s based on limitations in hardware and software (algorithms); the real barrier today is cultural; not technical. From 2004-2007, IBM and ILOG jointly worked to deploy the ILOG optimization product FPO to key tools sets in IBM’s 300mm fab resulting in substantial improvements in performance and significantly reduced overhead to adapt to changing circumstances. This paper will cover the fundamentals of the paradigm shift.

**1 INTRODUCTION**

The demand supply network for the production of semiconductor based packaged goods consists of a hierarchical sequence of decisions (Sullivan 1992; Figure 1) focused on assigning assets to meet demand ranging from a high level decision to allocate WIP and capacity to meet exit demand to the low level decision of assigning a specific manufacturing lot to a specific tool to receive a specific value add process in wafer fabrication. This last decision is called a dispatch or assignment decision and thousands of these decisions are made each day.

The assignment decision process must coordinate the actions and decisions of several logically isolated participants in a serially dependent system of activities. Therefore it balances the requirements of several goals (cycle time, output, serviceability, and inventory management) which compete for the same resource,

exploits emerging opportunities on the manufacturing floor, and reduces the distortion from unplanned events

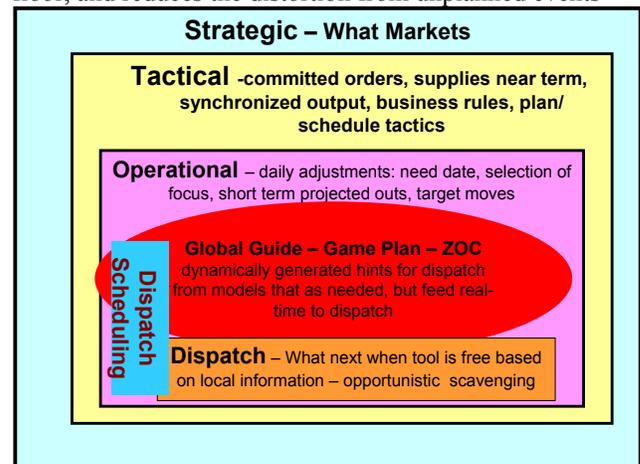


Figure 1: Sullivan decision tiers

Historically, dating back to the work in the early 1980s by IBM’s Gary Sullivan (Sullivan et al. 1992), the dominant decision technology found in applications to make dispatch decisions was “rules” which involve the follow basic computational mechanisms: merge, select, sort, and if/then/else in a decision tree. Although these “rules structures” (which we will call heuristic dispatch) can and do capture some reasonably complex logic and provide reasonable decisions in many cases, they fundamentally lack a robust ability to: (a) look across time, (b) look across tools at a tool set, (c) create an anticipated sequence of events at a tool set over some time horizon, (d) establish a formal metric for the quality of the decision, and (f) search across alternatives. However, standard wisdom was the rapid pace of change and short time interval between dispatch decisions precluded the use of more sophisticated decision technology (such as optimization -- which had successfully displaced rules in other situations) to build “dispatch” applications. Therefore the core approach changed little, if any, over twenty years; in spite of advances in algorithms and computer performance.

In 2003-2004 time frame IBM recognized that “rules” and their opportunistic scavenging approach (Fox

and Kempf 1985, Lin 1992, Sullivan 1987) was resulting in lost thput, increased cycle time, a loss of flexibility, and the allocation of manufacturing energy on the wrong lots in its fully automated 300mm wafer fabrication facility located in East Fishkill, NY. Additionally, IBM believed that twenty years after it had developed and deployed one of the first real time rule based systems to control dispatch decisions, computational advances in algorithms and hardware made optimization possible. At the same time ILOG had arrived at a similar conclusion and was developing its FPO rapid scheduling application for key tool sets in FABS relying on recent advances in Integer and Constraint Programming..

From 2004-2007, IBM and ILOG jointly worked to deploy the ILOG optimization product FPO to key tools sets in IBM's fully automated 300mm wafer fabrication facility. FPO now controls the dispatch decision at each major tool set (over 1/2 of the tools) in the 300mm fab. It has resulted in substantial improvements in performance; significantly reduced overhead to adapt to changing circumstances; and made the complex easy to understand and control. The Fab moved from "heuristic dispatch" to "schedule dispatch".

This paper will provide an overview of the dispatch environment; the key differences between "rules" and "optimization", the development and deployment of optimization to control the assignment of lots to tools, specific examples from key tool sets, the key advances for the business and the tool set managers, and the basics of the core advances in the mathematical optimization over the past 20 years that enables optimization to be a practical alternative in such a dynamic and rapid fire environment.

## 2 FUNDAMENTALS OF THE WAFER FABRICATION, ASSIGNMENT DECISION AND THE ARCHITECTURE OF RULE BASED DISPATCH SYSTEMS

The work by Sullivan, Kempf, Fowler, Graves, Wein, Glassey, Bitran and many others in the 1980s and early 1990s in planning, scheduling, and dispatching wafer fabricators has been extensively written about the basics of wafer fabrication flow (reentrant); the core considerations in assigning a lot to a tool; and the basic architecture of a rule based dispatch system is now well known. Specific the applications for rule based dispatch there are (and have been for a while) a number of commercial applications as well as "home grown" applications that are part of day to day life. This paper will assume the reader is familiar with the basics and focus on reviewing some key aspects of the assignment decision and its solution.

The core architecture of a dispatch system and the method of obtaining and organizing knowledge has remained remarkably similar to the original work done by Sullivan. Figure 2 (Sullivan 1991) and Figure 3 provide a basic overview.

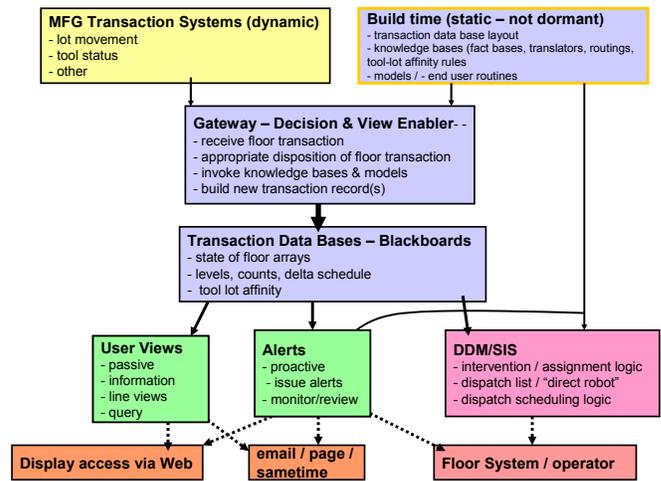


Figure 2: Core flow for rule based dispatch

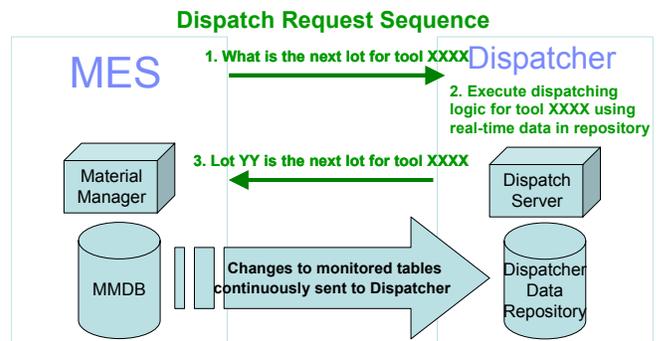


Figure 3: Typical dispatch request sequence

The problem has fundamentally remained true to John Fowler's 1992 (NSF Workshop October 1992 New Hampshire) description:

"Dynamic Production Planning and Scheduling is needed in semiconductor manufacturing because of the complexity of the manufacturing processes including factors such as unreliable equipment, batching, reentrant flows, rework, yield loss, hot lots, combination of production, engineering and R&D lots, and varying product mix and start rates. In addition to these factors, future wafer fabs will probably be more automated than current factories. This will require computers to control the flow of material through a fab instead of humans. While all of the factors mentioned above make the real-time control of a factory necessary, they also make it very complicated to determine what the best decisions are. There are, however, a few things that we have going for us.

1. Except for rework, most of the flow in a fab is deterministic (instead of probabilistic as in a job shop).
  2. The processing time per wafer or per lot or per batch is very nearly deterministic, so that once processing begins; we can get a very good prediction of when the processing will end.
  3. The shop floor control systems in place in current wafer fabs provide much of the information we need in order to make good decisions
- 1.6. status of lot – on hold, not on hold, send ahead required
  - 1.7. oper-tool relationships
  - 1.8. manufacturing engineering requirements
    - 1.8.1. counts (avoid too many wafers on certain tools)
    - 1.8.2. time limits (tool requires re-qualification)
    - 1.8.3. send ahead required
    - 1.8.4. result of metrology

**2.1 Fundamentals of the Assignment Decision**

Historically, the easiest way to conceive of dispatch is when a tool becomes free (moves from a busy state to a non busy state or comes back on line) lots compete to be the “next on the tool”. In practice, other triggers, such as the change in status of a lot or elapsed time, can trigger the logic to review and assign lot(s) to a tool. Figure 4 illustrates the basics of lots waiting for a tool at a tool set.

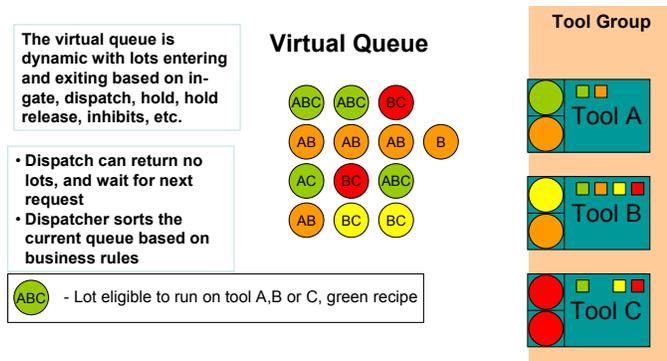


Figure 4: Basics of lots waiting to be dispatched

Fundamentally this decision process must first determine which lots are eligible to run on which tools and then narrow the selection process based on business guidelines (Due dates, Business Rules, dedication strategies, phase in tools); Process requirements (production paths, reticle available, tool specific inhibit); and Intrinsic properties of tools which significantly impact throughput (for example batch size and trains); and occasionally synchronization of events across the Fab. We can break the key elements down as follows:

1. tool – lot affinity
  - 1.1. what lots can run on this tool, what tools can handle this lot
  - 1.2. what are preferred tools, what are preferred lots
  - 1.3. establishes initial feasible state space with preference value based on tool-lot affinity
  - 1.4. includes
  - 1.5. what manufacturing activity (oper) next for a lot

2. global importance of the lot to the supply chain or business
  - 2.1. priority, customer, development vs production, lots
  - 2.2. one might have all lots the same, that is still global importance
3. pacing lot movement
  - 3.1. fluctuation smoothing, range management, delta schedule, critical ratio
4. local tool characteristics and performance
5. batching, trains
  - 5.1. time sensitivity for yield and rework
  - 5.2. local zone of control
  - 5.3. manufacturing engineering, tool-oper affinity
6. upstream and down stream requirements
  - 6.1. WIP aversion
  - 6.2. extended zones of control
7. more complex global controls beside pacing

**2.2 Guidance and Judgment**

The core dispatch decision making activities can be divided into two primary components: guidance and judgment.

**Guidance** or advocate logic is the set of computational activities (which may be a program or manual) to create information posted some location (often a table structure) that the assignment logic accesses or to trigger an assignment module to execute. The most common example is the calculation of some type of “delta schedule” value for each. Another example is the updating of a fact base that may contain operation – tool preference. A third example is the establishment of the tool deployment tables (operation – tool affinity) which is done separate from the assignment logic and posts this information to data set where the assignment logic can access this information.

**Judgment** or assignment is the set of computational activities that when completed, result in a change of state or action on the manufacturing floor. The judgment algorithm must balance competing requirements such as meeting on time delivery, but improving throughput with batches and trains.

Conceptually, the guidance logic and the assignment logic can run asynchronous – that is the guidance logic runs when it “wants” and without concern

for when the assignment logic runs and the reverse is true for the assignment logic. Sometimes during an implementation, we choose to run the two together; but conceptually the two could be split. Guidance logic can further be classified by the frequency with which it runs or its function.

### 2.3 Historical Limitations of Judgment

Dating back to the 1980s the computation methods used within judgment could be described as “decision tree one pass logic”:

- join or link
- sort and select
- scoring
- if then else
- one pass logic no looping (except accumulation), algorithm

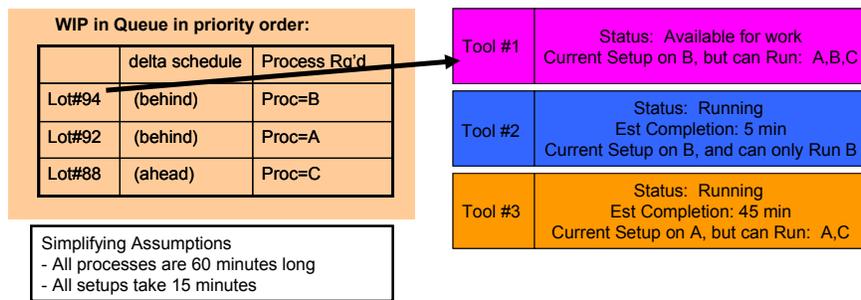
Although this method produced adequate decisions, the decisions were far from optimal and often rigid – that is requirement changes in the programming as business conditions changed. It stood in start contrast to the successful deployment of far more intelligent decision

making approaches in a wide range of applications from computer chess to supply chain planning to the placement of radioactive seeds to treat prostate cancer (\*\* give references). Despite the clear advances in other areas, Fabs have clung to their “only simple can work” approach and the corresponding culture of reaction and “heroic action”.

### 3 AN EXAMPLE OF THE LIMITATION OF RULES

The following example clearly illustrates the limitations of rules for “dispatch heuristics” and the value in optimization methods to deploy dispatch scheduling. These limitations include: the inability to (a) look across time, (b) look across tools at a tool set, (c) create an anticipated sequence of events at a tool set over some time horizon, (d) establish a formal metric and (f) search alternatives.

In top half of Figure 5 we have typical dispatch situation. There are three lots waiting to be processed. Two of the lots (94 and 92) are behind schedule. One of the lots is ahead of schedule (88). There are three tools in the tool group and their status is given in Figure 5. When tool#1 becomes available what lot is it assigned?



**Tool #1 has just become available -> What should it run next?**

Typically, heuristic dispatch will consider only the tool that is available. It will not look at the potential impact of actions in the future or across the tool set

**At time 0: Tool#1 free, chooses lot #94 to tool #1 because the lot is behind schedule and the setup matches (sound thinking)**

Figure 5: Example dispatch situation

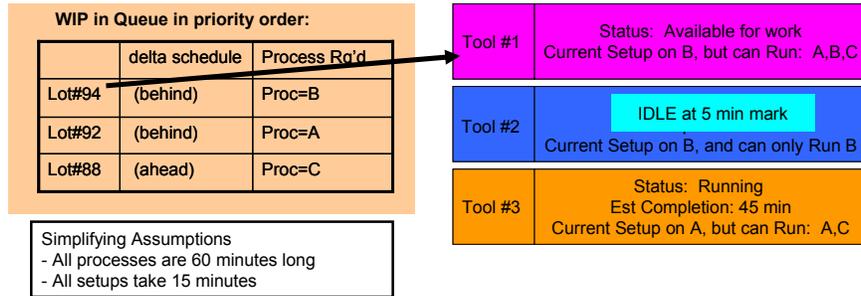
In bottom half of Figure 5 we have typical “intelligent” assignment decision by a rules engine. The dispatch engine chooses lot #94 to tool #1 because the lot is behind schedule and the setup matches. This meets two key criteria: working on lots that behind and improving throughput by avoiding setups --- very sound thinking -- however?

As we play events forward in time, we quickly (Figure 6) find previously “undetected collateral damage” from our decision -- at 5 minutes: tool # 2 idle -> no wip available since tool #2 can only run process B and there are no lots in queue that require process

Let’s return to the original decision and see how optimization can avoid the undetected collateral damage.

In Figure 7 we see three possible solutions that are projected forward through the complete of all lots currently waiting to be serviced by this tool group. Observe for each potential solution a tentative detailed schedule (in our example, we round to 10 minute intervals simply to avoid overburdening the reader with detail, but FPO keeps details to the minute or less if needed). Since we are projected forward in time across lots and across tools we can directly measure the impact of the decision in terms of such items as completion time for lots (Figure 7) and tool utilization or idle time (Figure 8). For both

measurements, options 2 and 3 substantially out perform option 1.



**Tool #1 has just become available -> What should it run next?**

Typically, heuristic dispatch will consider only the tool that is available. It will not look at the potential impact of actions in the future or across the tool set

**At time 0: Tool#1 free, chooses lot #94 to tool #1 because the lot is behind schedule and the setup matches**

**Undetected collateral impact of this decision**

**At 5 minutes: tool # 2 idle -> no wip available since tool #2 can only run process B and there are no lots in queue that require process B**

At aggregate level shows idle with WIP (albeit wrong WIP)

Figure 6: Collateral damage and limits of dispatch

### Three Options

scaled to 10 min intervals, start/stop are approximate

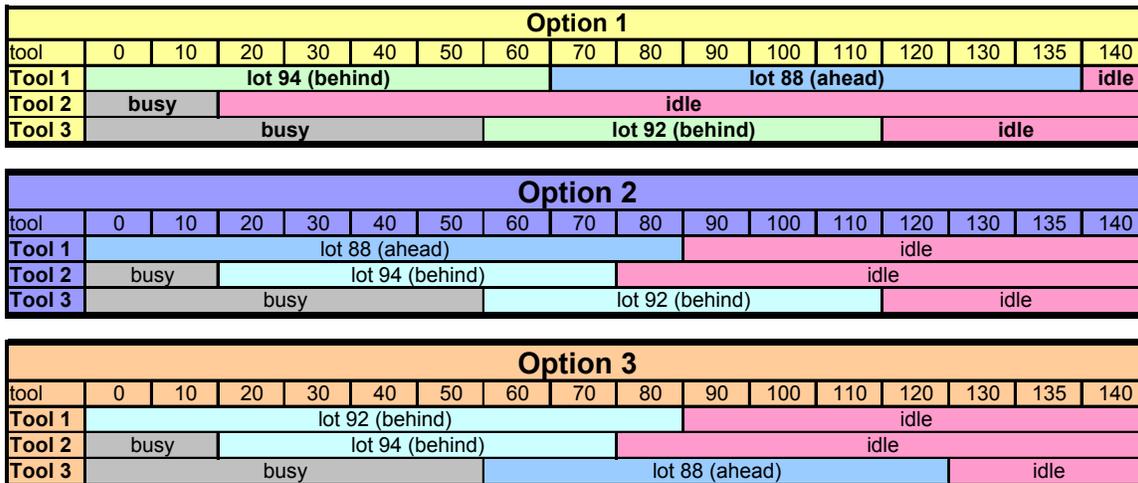


Figure 7: Three sequence schedule options

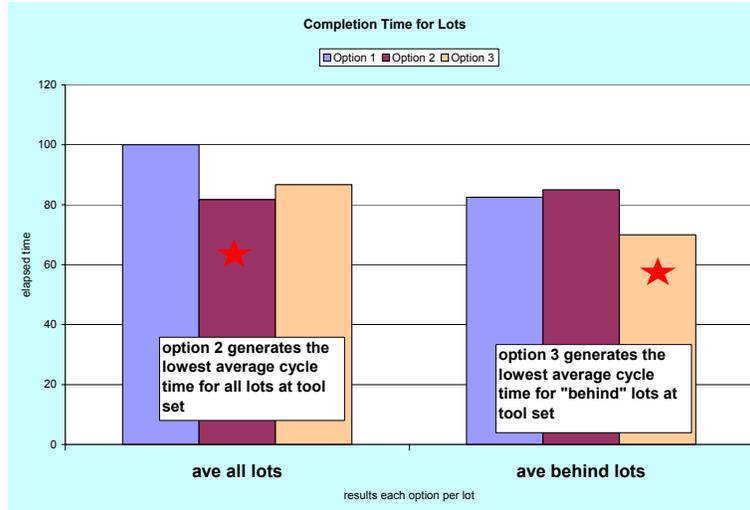


Figure 8: Comparison of three options for lot complete time

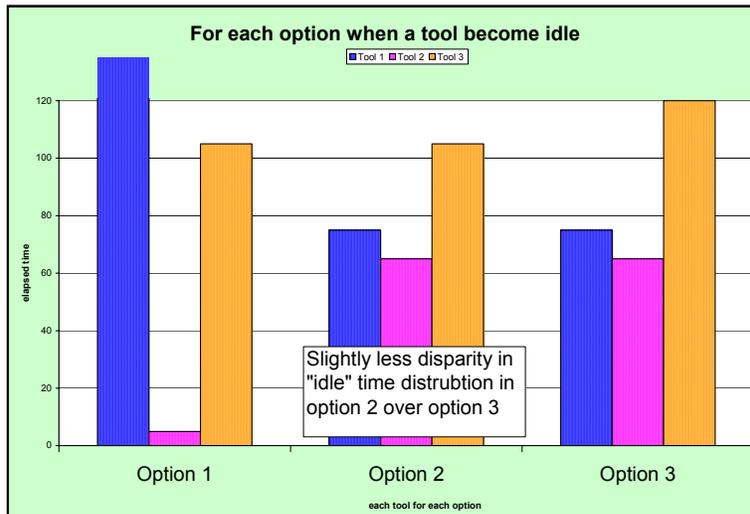


Figure 9: Comparison of three options for tool utilization

The application of optimization methods improved the assignment decision by generating optional decisions across time and tools, evaluating these decisions with a firm metric, searching for other options, and having a selection or stopping criteria. Observe: Option 2: lower average cycle time for lots and less variation in exit time for lots and tool use. Option 3: lot#92 (which is behind) completes sooner good utilization of tools. Both options are superior to option 1. The best option depends on (a) details (how far behind and ahead, lot priority, demand class, (b) near term WIP conditions, and (c) business priorities. However the value of optimization goes well beyond a better assignment decision, but provides a visibility impossible to obtain with simple point of sale rules. It enables everyone to clearly see an anticipated projected schedule for lots and tools and anticipated tool utilization and WIP levels. This is critical, since the assignment decision is strongly influenced by many other

decisions – two key ones are deployment decisions (the processes a tool is qualified and/or allowed to handle) and the movement, location, and cleaning of masks or reticles.

Additionally optimization enables the organization to move beyond ranking and transition from specifying what from how. Let's look at two examples.

A typical goal for a fab is to eliminate unnecessary transport of lots. The goal is "minimize" lot movement and this part of a goal of minimizing cycle time. The optimization engine with create solutions with this goal in mind.

A rule engine requires the rule writer to specify "how to limit" travel. A typical rule would be: <If lot is of type A, B, or C do not let it travel to another sector to be processed by a tool until it has waited at least 30 minutes in its current sector>. What the rule really wants is to keep a lot from traveling to another sector from its current sector if there is a tool in this sector that can handle the lot

reasonably soon. There is no way to specify this general policy. So the rule specifies the “specifics” of how and hopes many times a tool will become free in the lot’s current sector within 30 minutes that is a good match for this lot. Obviously, if a fab is heavily loaded this is a higher probability event. A fab that is only moderately loaded could have a problem and the rule would need to be manually changed.

A second common limitation involves the use of rankings. Assume lots A, B, and C are behind schedule by 1.1, 1.2, and 4.8 days respectively. The lot C has the rank of 1 (most behind) since it is 3.6 days worse off the lot B. Alternatively if lot C was behind schedule by 1.3 days it would still have the rank of 1, but it just barely worse off (0.1 days) then lot B. Typically, heuristic dispatch will work with the ranking and not be able make use of the richer information about the magnitude of the difference. Scheduling dispatch (mathematical optimization) will.

**4 WHAT MATHEMATICAL METHODS MAKES THIS POSSIBLE?**

The challenge, besides finding a quality solution, is finding a quality solution fast. The requirement is the generation of a complete solution in 60 seconds or less, which enables an updated solution to be generated every 5 to 10 minutes. Obviously, simple brute force of generating all or most alternatives and evaluating is not going to the job done. The key is the coordinated use of mixed integer programming (MIP) from Operations Research and constraint programming (CP) from computer science to narrow and direct search.

MIP is well suited to resource allocation applications and is used extensively to compute optimal order fulfillment locations, crew-shift equipment assignments, vehicle routes in transportation and production plans for manufacturing. CP has been successful in solving large combinatorial problems in the areas of planning, scheduling, natural language processing and DNA sequencing. CP techniques are a particularly effective companion for MIP techniques in detailed scheduling applications. Two fab examples are:

1. Two lots (A and B) with identical characteristics (manufacturing activity, lot size), etc
  - 1.1. Lot A is behind schedule and Lot B is ahead
    - 1.1.1. Lot B will never go before Lot A
2. Mask W has “50 wafers life” left before maintenance.
  - 2.1. Lots A, B, C, D, E (each with 25 wafers) require mask W
    - 2.1.1. Near term schedule will always be limited to 2 out of 5 of these lots

For an overview of MIP see Wolsey (1998) and Williams (1999). For an exposition of the fundamentals of CP see Lustig and Puget (2001).

In spite of the widespread applications of integrality-based techniques, up until as recently as the late

1990s, it was generally acknowledged, even by experts in mixed-integer programming, that while these techniques were a powerful tool in the solution of schedule models, they were simply not fast and robust enough to offer the turn-around times that were necessary in real-time, or even near real-time applications (Durbin and Hoffman 2001). However, as demonstrated in Bixby (2001), Bixby (2003), and Bixby and Rothberg (2003), that situation has changed dramatically in the last several years. In 2001 ILOG began to seriously investigate the use of MIP and CP to address detailed production scheduling for a large semiconductor fab.

**5 BASICS OF APPLYING MIP AND CP TO FAB PRODUCTION SCHEDULING**

“Our approach uses MIP and CP as components in a special-purpose decomposition algorithm that iterates alternately over the space and time dimensions. We call the algorithm STARTS for Space-Time Allocation for Real-Time Scheduling. To solve fab scheduling problems using the STARTS algorithm, operational models must first be defined in terms of variables, objectives and constraints. During production operations, lot and equipment status data from a fab manufacturing execution system (MES) are continually sent to the STARTS scheduling software and evaluated according to the defined model.

Conceptually, the scheduler begins by generating variable values that satisfy all of the constraints (i.e. by finding a feasible solution). When a feasible solution is found, the scheduler evaluates the objective function using these same values. If more than one feasible solution exists, the STARTS algorithm continues to modify the variables to improve the objective function. Feasible solutions are compared until the optimal solution is determined. It is central to the success of these methods that the theories of MIP and CP allow the determination of these ever-improving solutions by explicitly examining only a very small fraction of the total number of feasible solutions.

The optimal solution for a fab process area schedule contains a list of lotstep assignments to specific tools for a certain time horizon (usually 8 to 12 hours) starting from the current time, with recommended start times and expected finish times. This schedule can be packaged as messages, database tables or files to be used by a lot dispatcher and viewed in Gantt format.

The user controls the behavior or personality of the solver by inputting preferences or priorities and weights. For example, the user can set “preference” for urgent lots (P1), throughput (P2), and total item time (P3). For example, by setting P1 >> P2 >> P3, the urgent\_lot\_assignment would be maximized first, then the throughput, and, finally, the total\_idle\_time would be minimized. In effect, this choice would rank the urgent-lot

objective higher than throughput, and throughput higher than tool idle time. The priority values illustrated by P1, P2, and P3 (above) can be tuned to respond to changing operational goals. For example, during a new fab ramp, the urgent lot assignment objective may be ranked higher than the throughput objective in order to support critical process or product development. On the other hand, a fully-ramped production fab may rank the throughput objective higher than other objectives in order to meet critical production targets during high seasonal demand periods.” (Bixby, Burda, and Miller 2006)

## 6 RESULTS AND DISCUSSION

A set of critical fab metrics including throughput, cycle time, hot lot cycle time and QTime lot conformance, with secondary metrics including such measurements as batch size and balance across tool sets, were analyzed to determine the effectiveness of the solution. The scheduler provided benefits in throughput, cycle times, and hot lot performance, while automating managing of QTime lot scheduling. Additionally, the scheduling solution also provided fab operations with improved visibility. FPO is currently deployed in about ½ of the tools and most of the critical tool sets.

The new optimization methods enable fabs to take the next leap in dispatch scheduling. The biggest obstacle is no longer technology, but social order.

“There is no more delicate matter to take in hand, nor more dangerous to conduct, nor more doubtful of success, than to step up as a leader in the introduction of changes. For he who innovates will have for his enemies all those who are well off under the existing order of things, and only lukewarm support in those who might be better off under the new.” (Niccolo Machiavelli)

## REFERENCES

- Bitran, G., and D. Tirupati. 1989. Tradeoff curves, targeting and balancing in manufacturing networks. *Operations Research* 37:547-555.
- Bixby, R., R. Burda, and D. Miller. 2006. Short-interval detailed production scheduling in 300mm semiconductor manufacturing using mixed integer and constraint programming. In *Semiconductorfabtech – 32nd edition*, <www.fabtech.org>, 34-40.
- Bixby, R. 2001. Solving real-world linear programs: a decade and more of progress. *Operations Research* 50: 3-15.
- Bixby, R., M. Fenelon, Z. Gu, and E. Rothberg. 2004. Mixed-Integer programming: a progress report, The sharpest cut, Grotscchel, M. (editor), Society for Industrial and Applied Mathematic ISBN 0898715520.
- Bixby, R., and E. Rothberg. 2003. Solving linear and integer program. MPI Informatik ADFOCS 2003 </www.mpiinf.mpg.de/conferences/adfocs-03/Slides/Bixby\_1.pdf>
- Bruno, G., A. Elia, and P. Laface. A rule based system to schedule production. *IEEE Computer* 19(7):32-41.
- Chen, H., M. Harrison, A. Mandelbaum, A. Ackere, and L. Wein. 1988. Empirical evaluation of a queueing network model for semiconductor wafer fabrication. *Operations Research* 36(2):202-215.
- Consilium, 1988. Rule Based Dispatch User's Guide, Mountain View, CA.
- Dabbas, R., and J. Fowler. 2003. A new scheduling approach using combined dispatching criteria in wafer Fabs. *IEEE Transactions on Semiconductor Manufacturing* 16(3):501-510.
- Fargher, H., and R. Smith, 1994. Planning in a flexible semiconductor manufacturing environment. Chapter 19 in *Intelligent Scheduling*, edited by Mark Fox and Monte Zweben, Morgan Kaufman Publishers
- Fox, M. 1987. *Constraint-directed search: a case study of job-shop scheduling*. Morgan Kaufman Publishers, Los Altos, Ca.
- Fox, B., and K. Kempf. 1985. Complexity, uncertainty, and opportunistic scheduling. In *Proceedings of the IEEE Second Conference on Artificial Intelligence Applications: The Engineering Knowledge Based Systems*, Miami, Florida, 487-492.
- Glassey, C., J. G. Shanthikumar, and S. Seshadri. 1996. Linear control rules for production control semiconductor Fabs. *IEEE Transactions on Semiconductor Manufacturing* 9(4):536-549.
- Glassey, C., and W. Weng. 1991. Dynamic batching heuristic for simultaneous processing. *IEEE Transactions on Semiconductor Manufacturing*, 4(2):77-82.
- Graves, S., H. Meal, D. Stefek, and A. Zeghmi, 1983. Scheduling of re-entrant flow shops. *Journal of Operations Management* 3: 197-203.
- Gurnani, H., R. Anupindi, and R. Akella. 1992. Control of batch processing systems in semiconductor wafer fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing*, 5:4. 319-327.
- IBM Burlington Today*. 1990. Decision making, scheduling, and running the line. In *Communications Operations Department*, General Technology Division, IBM, Essex Junction, Vermont 05452, 26:136, 1.
- Johri, P. 1989. Dispatching in an integrated circuit wafer fabrication line. In *Proceeding of the 1989 Winter Simulation Conference*, 918-921.
- Kempf, K., D. Le Pape, S. Smith, and B. Fox 1991. Issues in the design of AI based schedulers. *AI Magazine* 11(5):37-45.
- Kempf, K. 1989. Manufacturing planning and scheduling: where we are and where we need to be. In

- Proceedings of the Fifth IEEE Conference on AI Applications*, IEEE Computer Society Press, Los Alamitos, CA. 13-19.
- Kempf, K. 1989. Manufacturing scheduling: intelligently combining existing methods. In *Working Notes of AAAI AI in Manufacturing Symposium*, M. Fox editor, AAAI, 445 Burgess Drive Menlo Park, CA 94025-3496.
- Kempf, K. 1994. Intelligent scheduling semiconductor wafer fabrication. *Intelligent Scheduling* 18: 473-516.
- Lee C., R. Uzsoy, L. Martin-Vega, and P. Leonard. 1991. Production scheduling algorithms for a semiconductor test facility. *IEEE Transactions on Semiconductor Manufacturing* 4:271-280.
- Lee C., R. Uzsoy, and L. Martin-Vega. 1992. Efficient algorithms for scheduling semiconductor burn-in operations. *Operations Research* 40(4):764-775.
- Lin, G. 1992. *An opportunistic price-based multiple resource and part scheduling*. Ph.D. dissertation, School of Industrial Engineering, Purdue University.
- Lustig, I., and J. Puget. 2001. Program does not equal program: constraint programming and its relationship to mathematical programming. *Interfaces*, 29-53.
- Martin-Vega, L., M. Pippin, E. Gerdon, R. Burcham. 1989. Apply just in time at a wafer Fab case study. *IEEE Transactions on Semiconductor* 2(1):16-22.
- Savell, D, R. Perez, and S. Koh. 1989. Scheduling semiconductor wafer production: an expert system implementation. *IEEE Expert* 4(3):9-15.
- Sullivan, G. 1987. Logistics Management System (LMS): lessons in manufacturing dispatch. In *Flexible Manufacturing Systems*, edited by K. Stecke and R. Suri, Elsevier, New York 33-44.
- Sullivan, G. 1994. Logistics Management System (LMS): integrating decision technologies for dispatch scheduling in semiconductor manufacturing. Chapter 17 in *Intelligent Scheduling*, edited M. Fox and M. Zweben, Morgan Kaufman Publishers, 473- 516.
- Sullivan, G., R. Dunki-Jacobs, B. Gerard, and R. Sell. 1992. Logistics Management System (LMS): an advanced decision support system for the fourth decision tier dispatch or short interval scheduling. *Production and Operations Management* 1(1):70-86.
- Sullivan, G., M. Morreale, and J. McGrew, J. 1992. Knowledge based systems: APL techniques. In *Encyclopedia of Microcomputers* 9:345-378.
- Sullivan, G. 1990. Logistics Management System (LMS): implementing the technology of logistics with knowledge based expert systems. *Innovative Expert System Applications*, 183-202, AAAI and MIT Press, Menlo Park, CA.
- Wein, L. 1988. Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing* 1(3):115-130.
- Wolsey, L. 1998. *Integer programming*. John Wiley and Sons.
- Williams, H. 1999. *Model building in mathematical programming*. 4<sup>th</sup> Edition, John Wiley and Sons..

#### AUTHOR BIOGRAPHIES

**KENNETH FORDYCE** joined IBM in 1977 and is involved in the integrated application of decision and information technology to improve organizational performance especially for planning, scheduling, and dispatch. He has received awards from IBM, INFORMS, POMS, and AAAI; been a key contributor to IBM's advanced supply chain planning solution used inside and outside of IBM; and published numerous papers on planning, scheduling, and dispatch. He is an adjunct at Columbia University; serves as a research team member at Einstein College of Medicine; and has a PhD from Union.

**DR. ROBERT BIXBY** earned a Bachelor of Science degree from the University of California-Berkeley and a PhD from Cornell University. Bixby holds positions at Rice University as research professor and Noah Harding Professor Emeritus of Computational and Applied Mathematics, and as research professor of management in the university's Jesse H. Jones Graduate School of Management. Bixby was formerly chairman of the Mathematical Programming Society, and editor-in-chief of the journal *Mathematical Programming*. In addition, he has authored over 50 scholarly publications. He is a member of the National Academy of Engineering, and has received the Mathematical Programming Society Beale-Orchard-Hayes Prize for Computational Mathematical Programming as well as the INFORMS Impact Prize.

**RICHARD BURDA** is the lead architect for 300mm scheduling and dispatch at IBM and received a number of patents. Burda has held several positions relating to industrial engineering, operations management and product marketing in the semiconductor and other industries. He holds a Bachelor of Science in Mechanical Engineering from Villanova University and an MS in Management from the Pennsylvania State University