Short-Interval Detailed Production Scheduling in 300mm Semiconductor Manufacturing using Mixed Integer and Constraint Programming

Robert Bixby
CSO, ILOG Inc.
Mountain View, CA, USA
bixby@ilog.com

Rich Burda
300mm Scheduling &
Dispatch, IBM
East Fishkill, NY, USA
burda@us.ibm.com

David Miller
Senior Tech Staff Member, IBM
Essex Junction, VT, USA
davemill@us.ibm.com

Abstract

Fully automated 300mm manufacturing requires the adoption of a real-time lot dispatching paradigm. Automated dispatching has provided significant improvements over manual dispatching by removing variability from the thousands of dispatching decisions made every day in a fab. Real-time resolution of tool queues, with consideration of changing equipment states, process restrictions, physical and logical location of WIP, supply chain objectives and a myriad of other parameters, is required to ensure successful dispatching in the dynamic fab environment.

However, the real-time dispatching decision in semiconductor manufacturing generally remains a reactive, heuristic response in existing applications, limited to the current queue of each tool. The shortcomings of this method of assigning WIP to tools, aptly named “opportunistic scavenging”[1], have become more apparent in lean manufacturing environments where lower WIP levels present fewer obvious opportunities for beneficial lot sequencing or batching.

Recent advancements in Mixed Integer Programming (MIP) and Constraint Programming (CP) have raised the possibility of integrating optimization software, commonly used outside of the fab environment to compute optimal solutions for scheduling scenarios ranging from order fulfillment systems to crew-shift-equipment assignments, with a real-time dispatcher to create a short-interval scheduler. The goal of such a scheduler is to optimize WIP flow through various sectors of the fab by expanding the analysis beyond the current WIP queue to consider upstream and downstream flow across the entire tool group or sector.

This article will describe the production implementation of a short-interval local area scheduler in IBM’s leading-edge 300mm fab located in East Fishkill, New York, including motivation, approach, and initial results.

Keywords: Short-Interval Detailed Production Scheduling, Mixed Integer Programming (MIP), Constraint Programming (CP), cycle time

MOTIVATION / PROBLEM DEFINITION

IBM’s B323 300mm fab in East Fishkill, NY is one of the most automated semiconductor fabs in the world [2]. The fab operates in a truly “touchless” mode with fully automated lot and reticle dispatching. Both lot and reticle dispatching decisions are made by a real-time dispatching system common to the industry, integrated through the fab manufacturing execution system (MES). The fab produces a wide mix of leading-edge products for both IBM and external customers, and also supports development activity for future semiconductor technology generations. As such, it presents a very challenging scheduling environment to satisfy numerous low volume, high mix product sets.

Sophisticated dispatching rules have been created and refined to enable fully automated fabrication in IBM’s 300mm fab. However, there are several aspects of fab dispatching that remain problematic for a rule based dispatching system. Some examples:

1. Super-Hot Lots – Business requirements drive the need for a small number of lots to be run through the fab as fast as possible. Fab-wide dispatching rules effectively dispatch super-hot lots immediately as they arrive in each tool queue. However some tools delay the processing of dispatched lots due to jobs currently processing or waiting to be processed on the tool. Various manual and automated schemes have been tried to keep super-hot lots from “queuing at the tool”. These schemes involve idling tools ahead of the arrival of super-hot lots and trading equipment utilization for super-hot lot cycle time
2. Coordinated Batching – Many process flows have wet cleaning (WET) operations followed by diffusion furnace (FRN) operations. WET and FRN are both batch processes with different maximum batch sizes and different internal material logistic constraints. Effective coordination of WET and FRN batching offers the opportunity for improved tool throughput while lowering cycle-time, but this opportunity has not been fully realized with the real-time dispatching system.

3. QTime Restrictions – There are several places in a process flow where a group of consecutive operations must be performed within a certain time window to avoid yield issues. Lots that do not move through the QTime window in time must be reworked or scrapped. Most of the QTime restrictions either begin or end at WET or FRN. The dispatching logic for automatic QTime management is complex [3], and requires adjustment for changing WIP mix and tool configurations. Some of the most stringent QTime requirements have been aided by manual releases through artificial stopping points in the route prior to the start of the window.

Other challenges with existing dispatch capabilities include scheduling of non-product wafers, management of product mix, balancing WIP across tool sets, and other similar issues that require a more global solution. Govind and Iyer also provide a summary of these challenges for highly automated semiconductor fabs [4].

To address these problems certain fab logistics data must be available in a form that can be quickly and easily utilized by the dispatching rules. Specifically, the ability to have an approximation of WIP arrival at a given process step and knowledge of the downstream flow is a prerequisite to effectively dispatch through WET and FRN. Also required is a fully resolved view across the toolsets showing tool/chamber states, current jobs dispatched or running on the tools, and tool/WIP compatibility including run time restrictions or inhibits that are placed on multiple objects within the fab logistics model. While the raw data needed to derive these views is available to the dispatching system, there is significant pre-processing of the data required in order to make the information usable by a real-time dispatching rule.

Even if the fab logistic data is available in a usable form to the dispatching system, the heuristic nature of real-time dispatching may not be appropriate for solving the more complex of the dispatching problems. Dispatching rules for WET and FRN are often adjusted using various rules for minimum batch size, maximum wait time for batches, and batch sequencing. These rules require extensive maintenance as WIP mix, tool configuration, and tool availability changed.

Perhaps the situation is best described by Dabbas and Fowler's [5] summary that "in effect the dispatcher has tunnel vision, dutifully rank ordering lots in the queue but oblivious to what is happening around it". The fact is that there are a number of situations in semiconductor manufacturing where the application of fixed dispatch rules to rank order queues of lots may not produce optimal results. An ideal dispatch scheduling solution would model all aspects of the dispatching problem and deliver an optimal solution based on the current conditions in the fab. This would be a move away from dispatching by "rule of thumb" and toward a more analytical approach in which coordinated dispatching can meet the requirements listed above and identify opportunities for efficiency across the fab.

DETAILED SCHEDULING TECHNOLOGY FOR OTHER INDUSTRIES

Business applications have long used "integrality-based" optimization techniques such as Mixed Integer Programming (MIP) and Constraint Programming (CP) to solve complex scheduling problems.

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.

For an overview of MIP see *Integer Programming* by Wolsey [6] and *Model Building in Mathematical Programming* by Williams [7]. An exposition of the fundamentals of CP is given in "Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming" by Lustig and Puget [8].

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, that 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 [9]. However, as demonstrated in [10], [11], and [12] that situation has changed dramatically in the last several years. The successful application of optimization techniques to the mission-critical operations of commercial airlines, hospitals and financial institutions inspired ILOG, in 2001, to investigate the use of MIP and CP to address detailed production scheduling for a large semiconductor fab.

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. Figure 1 illustrates this system.

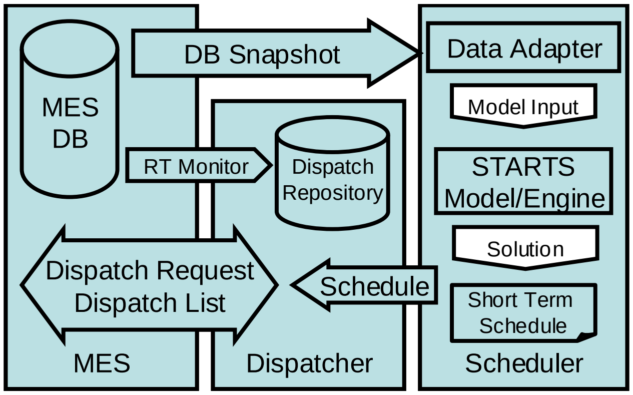

Figure 1. Conceptual Fab Production Scheduling System

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 lot-step 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 as shown in Figure 2. In this display, the left column contains a directory of tool resources while the right pane shows the recommended lot schedule for each tool resource. The beginning and end of a colored block indicates the recommended start time and expected finish time for a lot.

For any scheduling algorithm, the overall complexity of the model and the number of objects to be scheduled obviously influences the amount of time required by the scheduler to find an optimal solution. The STARTS algorithm uses separate, dedicated schedulers for each fab process area to reduce schedule computation time both by reducing the sizes of the individual problems that must be solved and by strongly reducing the number of “precedence constraints” that must be accounted for. The full, fab-wide scheduling problem can then be distributed across multiple CPUs. With this partitioning, the software provides an updated area schedule every five minutes. Each process area scheduler is designed to optimize multiple objectives, including minimizing the violations in soft constraints, while strictly obeying the specified hard constraints.

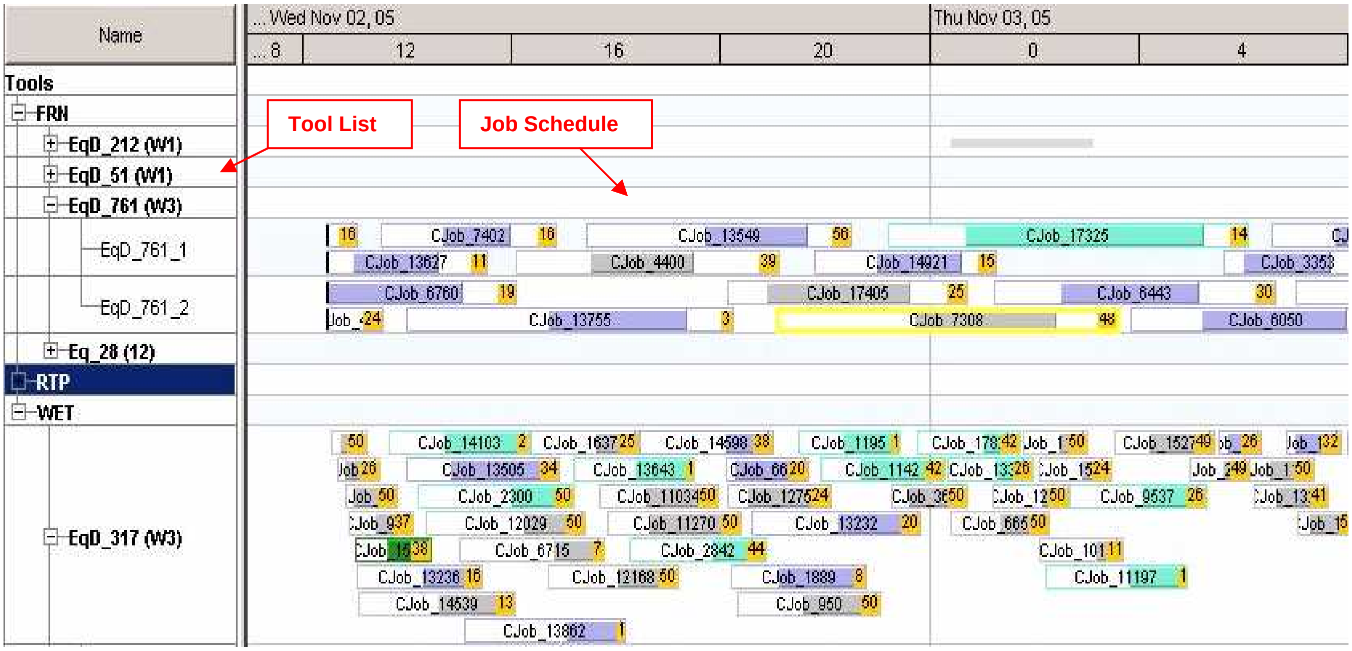

Figure 2. Detailed Production Scheduler Gantt Display

Multiple objectives can be considered simultaneously within a single objective function and ranked in importance as shown in the following example

Maximize {

(P1*(*urgent_lot_assignment*}) +

(P2*(*throughput*}) -

(P3*(*tool idle_time*})

},

where the idle_time term appears with a minus sign since it is to be minimized, and P1, P2, and P3 are priorities, or weights, that determine the ranking. By for example 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.

FAB-WIDE OBJECTIVES AND DATA REQUIREMENTS

Although each process area has unique aspects in its operational model, all process area schedulers share some common objectives and data requirements.

Common Scheduler Objectives

1) *Maximize {urgent lot assignment}:* Prioritizes the scheduling of hot lots and urgent monitor lots, or any other lot type specified as urgent. This objective helps the fab implement product development and critical process control goals or trouble-shoot yield problems.

2) *Maximize {throughput}:* Schedules the maximum number of wafers that can run with the set of tools for the specified time horizon. This objective can consider various lot attributes including priority and due dates in determining the optimal assignments and sequences. It is particularly effective in reducing average production cycle time.

3) *Minimize {time fence assignment changes}:* Avoids changing lot assignments made in a previous schedule within a specified time fence (i.e. a time window with respect to the current time). This objective effectively limits schedule thrashing and tries to avoid aborting physical transfers, tool loading or setups that may already be in progress.

4) *Minimize {bay moves}:* Assigns lots to tools that are physically close to the lot's start time location. This objective effectively reduces lot transfer time. Each tool and stocker location must be identified with a bay location to support this objective.

5) *Minimize {lot wait time}:* Prioritizes assignments for lots that have been waiting longer than other lots.

These basic objectives, along with their priority constants, guide each process area scheduler in managing trade-offs between conflicting operational goals.

Common Scheduler Data Requirements

1) Each tool within a scheduler's domain must be configured with the list of process recipes that it can accept.

2) A *raw process time* must be assigned for each process recipe. This information enables the scheduler to compute expected finish times, to synchronize batching (if applicable) and to sequence several lots or batches for each tool.

3) The *maximum lot capacity* for each tool must be set.

4) Each route with all process steps and process recipes required for each step and the list of tools used at each process step must be loaded into the scheduler model. This enables the scheduler to recommend future tool assignments for each lot for several steps downstream, including the lot's current location and immediate next step.

5) Tool status must equal "up" and "available" in order for the tool to be assigned lot-steps.

DIFFUSION SCHEDULING OBJECTIVES AND DATA REQUIREMENTS

Figure 3 illustrates the basic operating model for a diffusion process.

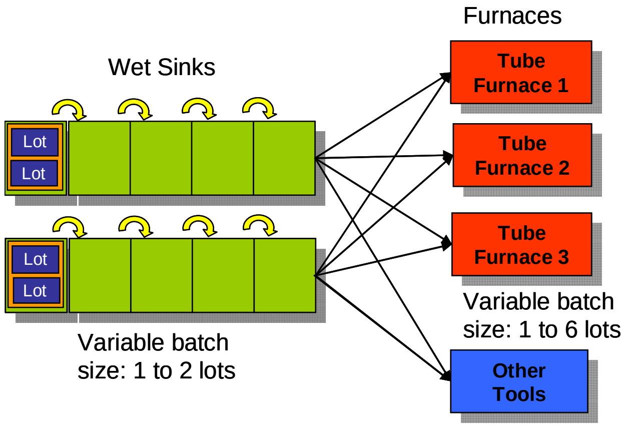

Figure 3: Diffusion Operational Model

Wet sinks typically process wafers in batches of up to two lots; they feed furnaces with a maximum lot batch size equal to six. In addition to the fab-wide common objectives and data requirements, a diffusion area scheduler must be modeled with specific objectives and data requirements.

Diffusion Scheduler Objectives

1) *Maximize {large batch}:* Constitutes batches in order to maximize the overall average batch size, subject to limitations on the maximum batch size and recipe compatibility conditions within batches. The scheduler may slightly delay a batch start time in order to allow more lots arriving at different times to join a batch. However, the *Minimize {lot wait time}* objective mitigates excessive delays and allows the scheduler to start smaller batches where excessive delays would result.

2) *Minimize {Qtime violations}* attempts to schedule lots for a thermal step before their Qtime expires. Qtime is the maximum time allowed between the completion of a wet clean process step and the start of a thermal process step. This objective usually has very high priority, but is nevertheless treated as an objective rather than a (hard) constraint, since current WIP and other fab conditions outside the control of the scheduler may make it impossible to avoid all Qtime violations.

3) *Maximize {monitor lot run with production batch}:* Rewards the scheduler for batching a monitor lot with a production lot when both use the same recipe and a monitor lot run is due.

Diffusion Scheduler Data Requirements

1) *Maximum batch size:* Enforces the limit on the maximum number of lots that can be run together in a batch.

2) *Maximum number of runs between monitor lots:* Defines the number of production lot-step assignments for a diffusion tool before a monitor lot must be run.

3) *ML/MR FOUP status:* Enables the scheduler to handle multiple lots (ML) and multiple recipes (MR) within the same FOUP.

4) *Internal tool buffer size:* Enables the scheduler to manage additional lots to be loaded and queued inside the tool while other lots are still processing. This helps reduce tool idle time.

5) *Tank status:* Enables the scheduler to consider routine maintenance and cleaning cycles for wet sink resources when determining lot processing schedules.

6) *Furnace boat movement sequence:* Enables the scheduler to minimize internal furnace resource idle time by synchronizing batch start times with boat loading and unloading.

Implementation Overview

The described scheduler solution was developed for implementation into IBM's B323 environment. The solution was seamlessly integrated into the fab IT environment, including interface with IBM's SiView MES solution via MES replicate DB2 tables containing necessary fab data,. The resultant schedule recommendations were fed to the existing dispatching application for actual dispatch. The system was thoroughly tested in a Proof of Concept (POC) evaluation.

Based on the results of the POC evaluation, the scheduler was implemented into the production fab environment to schedule all wafers through a population of 15 dual-tube furnaces and 18 wet tools with 6 tanks per tool. The schedule covered all product and non-product wafers processing through the tool set, including automated scheduling of QTime lots, hot lots, tool qualification runs, monitor builds, development and engineering lots, and all other processing, with the following considerations:

- Approximately 500 lot-step assignments were included in each schedule update
- The schedule horizon was 12 hours (e.g. approximately one work shift)
- Approximately 3 hours of WIP in other process areas upstream from diffusion was included in the diffusion schedule.
- The scheduler software ran on a single 2 GHz processor and completed each schedule run within about five minutes.

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. Thirty days of fab performance data from the period immediately before the scheduler release was used as baseline for the comparison analysis, and the first 30 days of summary performance data after release are included in the summary table.

The results versus baseline metrics are summarized in the table below. The scheduler provided benefits in throughput, cycle times, and hot lot performance, while automating managing of QTime lot scheduling. Furthermore, the benefits extended beyond the immediate Diffusion tool set to overall fab improvements in throughput and cycle time performance metrics.

The results in Table 1 demonstrate the potential of the scheduling solution implemented in IBM to produce improved results versus those achieved via sophisticated dispatch rules created and refined over several years.

Table 1. Throughput and cycle-time improvements

Results vs. Baseline	FRN	WET
TP vs Baseline	8.6%	6.9%
CT vs Baseline	-25.3%	-8.2%
Hot Lot CT Reduction	-15.4%	-17.9%

The scheduling solution also provided fab operations with improved visibility to fab status, dispatch decisions, and upcoming scheduled dispatches, reducing the dependence on manual activity to monitor and manage such situations as QTime and Hot Lot dispatching.

CONCLUSION

Detailed production scheduling software using the STARTS algorithm has demonstrated the potential to reduce cycle time, increase throughput and accelerate hot-lot processing in a fully-automated, leading-edge semiconductor fab. The following tasks were completed in order to deploy this software.

- Detailed operational models were configured for each process area scheduler.
- Relevant MES items (e.g. process routes, steps, recipes, raw process times, tool IDs, etc.) were downloaded to the scheduler before it was used in production.
- During production, real-time data interfaces provided current lot and tool status from the MES to the scheduling software.
- The scheduling software provided the recommended schedules in formats usable by the existing dispatching system.

Although this work was limited to the diffusion process areas, schedulers for other fab process areas are expected to provide improvements in their respective areas. Furthermore since fab areas are linked throughout the process flow, multiple schedulers are expected to provide additive benefits beyond the independent benefits of each scheduler.

ACKNOWLEDGMENTS

We sincerely thank the IBM and ILOG team members who worked on the development, integration and deployment of the scheduling solution, with special thanks to Dave Travagline, Vincent Gosselin and Michel Jaczynski.

REFERENCES

[1] Sullivan, G. 1987, "Logistics Management System (LMS): Lessons in Manufacturing Dispatch", in Flexible Manufacturing Systems, pp. 349-354, edited by K. Stecke and R. Suri, Elsevier, New York.

[2] "2005 Top Fab: IBM", December 2005, Semiconductor International *online magazine* http://www.reed-electronics.com/semiconductor/article/CA6285986l

[3] Scholl, Wolfgang and Domaschke, Joerg, "Implementation of Modeling and Simulation in Semiconductor Wafer Fabrication with Time Constraints Between Wet Etch and Furnace Operations",Aug 2000, IEEE Transactions on Semiconductor Manufacturing, Vol. 13, No. 3 (pp.273-277)

[4] Govind, Nirmal and Iyer, Bala, "The Case for Near Real-Time Production Scheduling in a Highly Automated Semiconductor Environement", 2005 IEEE/SEMI Advanced Semiconductor Manufacturing Conference

[5] Dabbas, Russ M. and Fowler, John W. "A New Scheduling Approach Using Combined Dispatching Criteria in Wafer Fabs", Aug 2003, IEEE Transactions on Semiconductor Manufacturing, Vol. 16, No. 3 (pp. 501-510)

[6] Wolsey, Laurence A. *Integer Programming,* 1998, John Wiley and Sons, ISBN 0-471-28366-5

[7] Williams, H. P. *Model Building in Mathematical Programming, 4th Edition*, 1999, John Wiley and Sons, ISBN 0-471-99799-9

[8] Lustig, Irvin J. and Puget, Jean-Francois, Dec 2001 "Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming", INTERFACES 31: 6 November-December 2001 (pp. 29-53)

[9] Durbin, M. and Hoffman, K "The Dance of the 30-ton Trucks", to appear in INFORMS Journal on Computing

[10] Bixby, Robert E, "Solving Real-World Linear Programs: A Decade and More of Progress", 2001, Operations Research, 50, pages 3-15

[11] Bixby, Robert E, Fenelon, M., Gu, Z, Rothberg, E. "Mixed-Integer Programming: A Progress Report", *The Sharpest Cut*, (pp. 309-326) 2004, Grotschel, M. (editor), Society for Industrial and Applied Mathematic ISBN 0898715520

[12] Bixby, Robert E. and Rothberg, Edward. 2003. "Solving Linear and Integer Programs", MPI Informatik ADFOCS 2003, http://www.mpi-inf.mpg.de/conferences/adfocs-03/Slides/Bixby_1.pdf

BIOGRAPHY

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.

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

David J. Miller earned a Bachelor of Science in Mathematics from Georgetown University, a Master of Science in Computer Science from the University of Vermont, and a Master of Science in Manufacturing Systems Engineering from Lehigh University. Miller is currently a Senior Technical Staff Member in IBM. His responsibilities include semiconductor factory automation and fabricator logistics solutions.