The Report Committee for Chuwen Zhang
Certifies that this is the approved version of the following report:

Controlling Work in Process During Semiconductor Assembly and Test Operations

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Jonathan F. Bard

G. Rodolfo Chacon

Controlling Work in Process During Semiconductor Assembly and Test Operations

by

Chuwen Zhang, B.E.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2017

Acknowledgements

To my advisor, friend and mentor, Prof. Jonathan F. Bard: I want to thank you for providing me the research opportunity in the second year of my graduate studies; it has been a fantastic journey to work, discuss, and learn with you. Your diligence, work ethic, perseverance will always inspire me to work hard.

I'm grateful to my parents, who have been supporting me since I applied for the graduate degree at the University of Texas. I understood it was not an easy financial decision to fund a graduate student to pursue his dreams in the United States. I also greatly appreciated the fact that you had never required my company even though I am your only child.

A special gratitude goes to the love of my life, Mingyu, who has been providing me with the strongest supports, motivation and love. I also appreciate your forgiveness to keep the long-distance relationship going. The post-application season was not easy for me, and I wouldn't be so resilient to the disappointment and frustration I experienced without your constant consolation and your positive attitude to life.

I'm also grateful to the friends I have in Austin. Victor and Yao have been helping me in daily life and listening to my updates and news. Tianheng and Huidong are the ones with whom I can initiate an academic discussion. I want to thank these kind individuals for their assistance and help throughout my short stay in Austin. Thank you all for your support and motivation.

Abstract

Controlling Work in Process During Semiconductor Assembly and Test Operations

Chuwen Zhang, M.S.E

The University of Texas at Austin, 2017

Supervisor: Jonathan F. Bard

In the semiconductor industry, products go through a series of steps over a three- to four-month period that begins with the fabrication of chips and ends with assembly and test (AT) and shipment. This paper introduces a mid-term planning model for scheduling AT operations aimed at minimizing the difference between customer demand and product completions each day. A secondary objective is to maximize daily throughput. Typically, semiconductor companies have 1000s of products or devices in their catalog that can be organized into unique groups of up to 100 devices each. This simplifies the planning process because it is only necessary to consider the groups as a whole rather than the individual devices when constructing schedules.

In all, we developed and tested three related models. Each provides daily run rates at each processing step or logpoint for each device group for up to one month at a time. The models are distinguished by how cycle time is treated. The first takes a steady-state approach and uses Little's Law to formulate a WIP target constraint based on the average

cycle time at each processing step. The second and third include integer and fractional cycle times in the variable definitions. To find solutions, raw production data are analyzed in a preprocessing step and then converted to input files in a standard format. FlopC++ from the COIN-OR open source software project is used to write and solve the model. Testing was done using three datasets from the Taiwan AT facility of a global semiconductor firm. By comparing model output with historical data for 6 device groups and 33 logpoints, we were able to realize decreases in shortages of up to 40% per month.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Semiconductor devices are manufactured in a predefined sequence of operations that are spread across a global supply chain.[†] Compared to other types of manufacturing, wafer fabrication is perhaps the most technologically complex and capital intensive due to long cycle times and the need to carry out a precise sequence of processing steps in a particle-free clean-room (Leachman 2002; Uzsoy et al. 1992). After the wafers are fabricated their electrical circuits are tested using a set of microscopic contacts or probes. Next, they are sent to an assembly and test (AT) facility where they are cut into chips, packaged, and further tested in what are called *back-end* operations. During assembly, the chips are protected from environmental contamination by encasing them in plastic or ceramic material.

The major manufacturing steps are depicted in Figure 1. Each step can be viewed as a reentrant flow shop in which a variety of machines must be carefully set up with tooling to carry out the associated processes. Front-end operations, consisting of fabrication and probe, can take up to two months and require more than 60 process steps. Back-end operations are slightly less protracted, taking up to 20 days and requiring between 25 and 30 steps (Van Zant 2000). AT facilities, the focus of this paper, are particularly sensitive to market demand since finished products are shipped either directly to customers or placed in regional distribution centers to satisfy forecasted demand. The goals of scheduling for AT facilities, therefore, are to achieve high throughput, high utilization and stable inventory to ensure high levels of customer satisfaction.

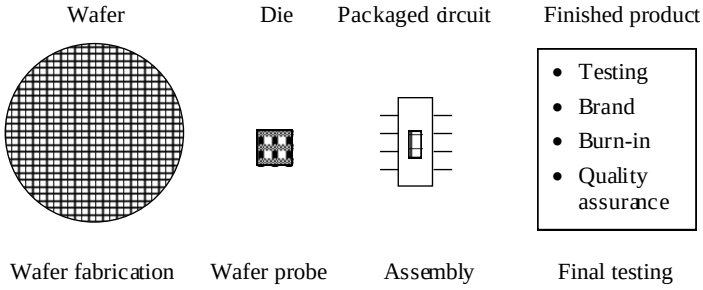

Figure 1: Basic steps in semiconductor manufacturing (Uszoy et al. 1992)

[†] Zhang, C., Bard, J. F., & Chacon, R. (2017). Controlling work in process during semiconductor assembly and test operations. *International Journal of Production Research*, 1-25. In this previous work, I was responsible for modeling, programming, data analysis, and testing.

In previous work, we investigated the machine setup-lot assignment problem over a planning horizon of a few days, and developed a mixed-integer programming (MIP) model with the primary objectives of minimizing the weighted sum of shortages and maximizing the weighted sum of lots processed (Deng et al. 2010). Solutions were obtained with a reactive greedy randomized adaptive search procedure (GRASP) that evaluates many combinations of machine-tooling and lot assignments as it iterates (Feo et al. 1991). In subsequent work, several limiting assumptions were relaxed. The expanded code now allows us to take into account initial machine setups, lot processing through multiple operations, and setups that require more than one tooling piece (Bard et al. 2013). To accommodate multiple passes (i.e., reentrant flow), a three-step heuristic was designed around the GRASP and was seen to provide higher machine utilization and improved schedules. More recently, Bard et al. (2015) used discrete event simulation to model AT facilities with AutoSched AP (ASAP). Inputs for the simulation were derived from the GRASP results which were used to set up and change over machines in near-real time. In a follow-on study, Jia et al. (2015) developed five new dispatch rules that were evaluated to determine their relative performance in meeting the hierarchical objectives. The findings are being used by Texas Instruments (TI), the company that sponsored this project, to provide guidance to shop floor supervisors.

Much of the aforementioned research has been aimed at short-term lot scheduling with machine setup considerations. The purpose of this paper is to show how mid-term production schedules can be developed, for up to a month at a time, to minimize expected shortages while fully utilizing the machine capacity at AT facilities. On any given day, these facilities may have thousands of lots in WIP (work in process) that consist of hundreds of different devices or products. At the TI Taiwan facility, a device typically goes through 27 operations, or what are referred to as *logpoints*, over 10 days before testing is complete and the final product is shipped to distribution centers. Daily demand is an input provided by the planning group and can vary widely over the month. The problem faced by shop floor supervisors is to determine which devices to run each day at each logpoint to best meet demand. In addition, since more than half of the orders are

forecast without firm customer commitments, the facilities operate in a built-to-stock environment. As a secondary objective then, it is desirable to process as many lots as possible rather than allowing machines to be idle. What further complicates the problem is capacity limits at each logpoint and the competition for machine time among all products.

In this paper, we provide a model that can be used to help planners determine production and WIP levels each day at each logpoint for each product. To control the size of real instances, end products are aggregated into device groups and scheduled in the aggregate. The model is less detailed than our models used for lot scheduling (Deng et al. 2010; Bard et al. 2013) because we are not interested in individual machine setups at the mid-term planning stage. We assume that once the plans have been transmitted to the floor, the GRASP and ASAP codes will be used to select the "optimal" machine-tooling combinations and lot assignments.

In addition to our planning model, the primary contributions of this paper are (1) a methodology for determining optimal levels of WIP and daily run rates by operation at AT facilities, and (2) a comprehensive analytic study based on data provided by TI. In the next section, we present some of the more recent literature related to short-term and mid-term production scheduling. In Section 3, the AT monthly scheduling problem is defined along with our input data files. In Section 4, we present the basic mathematical model that drives the computations, as well as two modified versions that offer a more accurate representation of process cycle times. The logic and data structures used in the implementation are described in Section 5. In Section 6, the models are tested and the results compared with actual production runs at TI's Taiwan facility (the input data has been modified slightly to conceal the faculty's production capacity and device characteristics). We also comment on the advantages and disadvantages of the different versions of the model. Additional observations and insights are provided in Section 7 along with several suggestions for future work.

Chapter 2: Literature Review

Over the past several decades there have been significant advances in the modeling and analysis of discrete parts manufacturing systems (Gershwin 2000; Monkman et al. 2008), with slightly less emphasis on the semiconductor industry (Montoya-Torres 2006). † Simulation has been widely used to study scheduling and dispatch rules during fabrication as well as during assembly and test (e.g., see Bard et al. 2015; Pfund et al. 2006; Sivakumar and Gupta 2002; Zhang et al. 2009). A general framework for production planning models in the semiconductor industry is provided by Hackman and Leachman (1989). An important component of their work is the derivation of procedures for dealing with fractional cycle times.

In semiconductor manufacturing, factors such as throughput, cycle time, utilization, and WIP are of primary importance. In the current market environment, manufacturers try to interact more closely with customers, and hence are more sensitive to due date performance. This is especially true for back-end operations, which are closer to the customer. The overwhelming factors faced by AT planners are the large number of machine-tooling combinations that must be considered when constructing schedules, along with the need to process the same lot multiple times. Allahverdi et al. (1999) undertook a comprehensive review of research aimed at solving static scheduling problems involving setup decisions. Lin and Lee (2011) updated the latter's findings by highlighting models, solution methods, and applications appearing in the literature through 2009.

An additional factor in managing semiconductor facilities is machine qualification, which affects capacity allocation and subsequently, daily production schedules. Fu et al. (2015) investigated this issue at Intel's back-end facilities. To deal with uncertain demand, they developed a stochastic optimization model whose solution balances the tradeoff between current machine qualification costs and future potential backorder costs due to insufficient capacity. Solutions for a range of instances were obtained with the L-shaped method.

† Zhang, C., Bard, J. F., & Chacon, R. (2017). Controlling work in process during semiconductor assembly and test operations. *International Journal of Production Research*, 1-25. In this previous work, I was responsible for modeling, programming, data analysis, and testing.

Short-term scheduling problems are mainly aimed at machine setups, tooling changeovers, and lot assignments. Uzsoy et al. (1992) provide a comprehensive review of semiconductor production planning and scheduling models. In our previous research, we developed a general model for setting up machines with tooling to run at specific temperatures, and for assigning lots to machines over a planning horizon of up to five days. Four objective functions hierarchically guided the analysis: (1) minimize demand shortages, (2) maximize weighted throughput, (3) minimize the number of machines used, and (4) minimize the makespan. Solutions were obtained with a reactive GRASP designed to exhaustively explore the feasible region. The algorithm is now running at TI's Clark AT facility in the Philippines.

In a follow-on study, Gao et al. (2015) developed a three-phase methodology based on optimization techniques that was competitive with the GRASP. In the first phase, an extended assignment model is solved to simultaneously assign tooling and lots to the machines. In the second phase, lots are optimally sequenced on their assigned machines using the four hierarchical objectives. Due to the precedent relations induced by the multiple pass requirements, some lots have to be delayed or removed from the assignment model solution to ensure that no machine runs beyond the planning horizon. In the third phase, machines are reset to allow additional lots to be processed when tooling is available. Comparative testing with the GRASP showed cost reduction across all objectives averaging 62% in the aggregate.

At a higher level of planning, Zhang et al. (2007) provide a hierarchical framework for allocating capacity for back-end operations. They focused on the reconfiguration of kit components during mid-term planning at AT facilities and proposed a MIP with the objective of reducing resource usage. Their methodology was successfully applied at one of Intel's AT sites resulting in an annual $10 million saving in the purchase of kit components. With respect to simulation, Zhou (1998) presented a tutorial on Petri net approaches for modeling and analyzing semiconductor manufacturing systems. He introduced the various properties of Petri nets and discussed their implications for model validation and system evaluation.

For flow balancing and tighter WIP control, techniques such as Kanban and CONWIP (constant work-in-process) have been widely proposed (Hopp and Spearman 2007). In a tutorial for simulating such systems, Marek et al. (2001) offer an overview centered around the products software ARENA/SIMAN 3.5/4.0. They also describe a heuristic for adjusting card levels in Kanban systems. Wang and Prabhu (2006) presented a parallel algorithm for CONWIP systems to reduce the computation times that grow exponentially with multiple products. Their algorithm searches over neighborhoods with high WIP levels to improve balance, and was shown to accelerate runtimes by a factor of five when implemented with ten parallel processors.

For complex systems, heuristics are routinely used for production planning. For example, Disney et al. (2000) proposed a genetic algorithm for managing various inventory systems. They studied three classic control policies that make use of sales, inventory, and pipeline information to set the order rate to achieve a desired stock level. Our problem falls into this category but without CONWIP assumptions.

For multistage, reentrant systems like those found in the semiconductor industry, modeling strategies are needed for planning purposes. Hung and Leachman (1996), for example, propose a methodology that iterates between simulation and linear programming. They focus on front-end operations characterized by "epoch-based" flow cycle times, i.e., fractional cycle times rather than discrete cycle times. In a groundbreaking paper, Leachman et al. (1996) describe their optimization-based production planning system developed for Harris Corporation, which integrates front-end and back-end operations to provide a complete solution. At a high level, their system includes requirements for binning and substitutable products, for representing dynamic capacity consumption within a reentrant flow environment, and for accommodating market priorities over time. A critical output was realistic delivery quotes that markedly increased customer satisfaction and market share.

Chapter 3: Problem Statement

When wafers arrive at back-end facilities they are cut into chips (devices), packaged, tested, and shipped to either customers or regional distribution centers. Each device undergoes 25 to 30 steps over a two-week period before it leaves the facility. [†] Companies like Texas Instruments have thousands of individual part numbers and dozens of customers, all of which greatly complicates production planning. To ensure on-time performance and continued customer satisfaction the highest priority must be given to daily scheduling.

To make the problem more manageable, similar devices are grouped and scheduled together. The problem starts with the demand for each device group, which is transmitted to the shop floor through TI's hierarchical planning system (see Figure 2). Demand is specified by group as the daily output quantity required (DOQR), and accompanied by "starts." At the beginning of each day, supervisors are given instructions on how many devices to feed into the system. In Figure 2, Manufacturing Planning is responsible for day-to-day activities, scheduling starts over the month, setting targets by operation for each product being manufactured, and deciding when to move WIP between operations. Decisions at this level are made by shop floor managers and line supervisors. Dispatch and Execution is the recipient of daily target data and is responsible for ensuring that the scheduled work is carried out.

[†] Zhang, C., Bard, J. F., & Chacon, R. (2017). Controlling work in process during semiconductor assembly and test operations. *International Journal of Production Research*, 1-25. In this previous work, I was responsible for modeling, programming, data analysis, and testing.

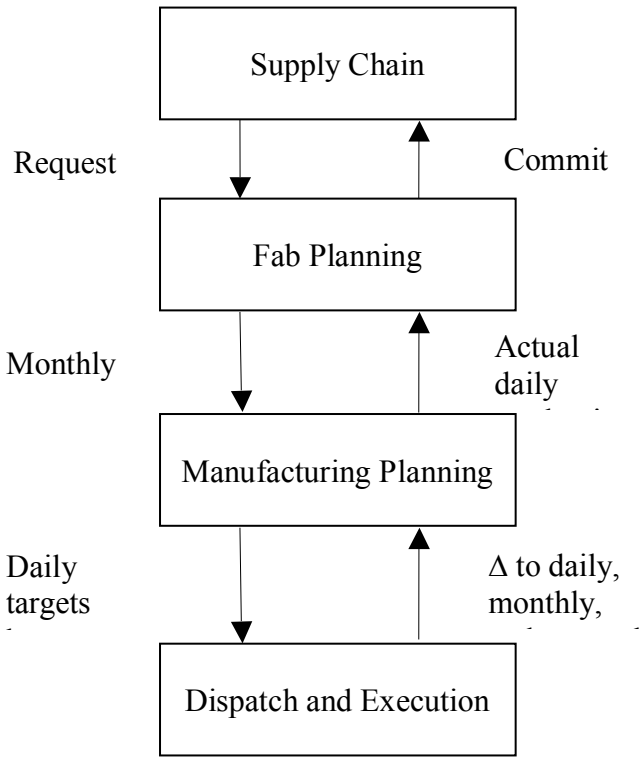

Figure 2: Hierarchical planning and scheduling at Texas Instruments

Device groups are defined by a combination of production line, type of tester on the line, number of pins or terminals attached to the package, and the mechanism by which devices are fed through the tester (this is referred to as the *flow*). Similarly, identical machines are grouped to handle a variety of operations at different logpoints. Devices from the same group follow the same logpoint sequence though the system. The length of the entire process is given by the production lead time (cycle time), and consists of the cumulative cycle times of the individual operations along a route. Generally, the full cycle time can have large variations for individual different operations and device groups. In monitoring performance, supervisors track the daily run rate (DRR) and WIP of each device group at each logpoint. WIP levels are observed at the end of the day, or equivalently, at the beginning of the next day.

The problem we wish to solve is to determine the daily run rates at each logpoint for each device group that minimizes total shortages and maximizes total throughput while staying within

the capacity limits of the facility. Capacity is specified either for each operation each day or for each operation by device group each day. An additional constraint is to stay within a certain percentage of WIP targets at each logpoint. These targets are established within the model and are used to hedge against uncertain demand. The two objective function terms are treated hierarchically as in weighted goal programming. Minimizing shortages is the more important of the two.

The planning horizon is typically four weeks, which is approximately twice the lead time of any device. When computing solutions, we assume that capacity is shared amongst device groups at each logpoint and that a day is divided into a fixed number of periods. In fact, many of the machines are flexible enough to perform several different operations as long as they have been previously qualified, so our results may be somewhat conservative. Note that in real-time scheduling, each machine must be set up with the appropriate tooling and then assigned lots. That problem requires much greater detail than the one we are considering here since we are not modeling individual machines, tooling and lots; see Bard et al. (2013) for a discussion of the short-term scheduling problem.

The input data for our problem are contained in four "csv" files that are created from a master file downloaded from the company's Oracle database system at the start of the planning horizon. Table 1 contains a portion of the "WIPBegin.csv" file. The first column lists the device, followed respectively by the four parameters "Prod line," "Pin," "Tester," and "Strip test" which define the device group. Next we have the logpoint, its description, its sequential order, its planned cycle time (in days), and the initial WIP level on the first day of the planning horizon. Note that there may be several devices within one group but they all follow the same sequence of operations given in the columns "LPT" and "LPT order." The order differs by device group. The cycle time of a route is the sum of the cycle times of the individual operations.

To illustrate, from Table 1 we see that "DFDRG4" is in device group "76-48-ZABC-Y." Its first operation is "5100" and its initial WIP is naturally 0 since there is never any carryover. After 0.09 days (130 min) it moves to logpoint "5105," and then to "5110" and so on. One

interesting observation is that the variation in initial WIP over all logpoints is extremely high. To some extent this is due to limited capacity at each operation as well as widely varying individual cycle times.

Table 2 gives a sample of the "WIPPlanStart.csv" file. The columns no previously identify list the dates of the planning horizon, the planned starts for each day ("Plan start"), and demand for each device ("Plan ship out"). If included, "Capacity" indicates the upper bound of production on that day; otherwise, it is defined in the user configuration file (see Section 5). Note that the date in Table 1 for begin WIP levels is August 11, 2016, which corresponds to the first day in Table 2.

Device	Prod line	Pin	Tester	Strip test	LPT	LPT Desc	LPT order	Plan CT	Begin WIP
DFDRG4	76	48	ZABC	Y	5100	LOT START	1	0.09	0
DFDRG4	76	48	ZABC	Y	5105	BACKGRIND	2	0.67	298193
DFDRG4	76	48	ZABC	Y	5110	SAW	3	0.73	0
DFDRG4	76	48	ZABC	Y	5200	MOUNT	4	1.77	94001
DFDRG4	76	48	ZABC	Y	5250	MOUNT CURE	5	0.23	35952
DFDRG4	76	48	ZABC	Y	5300	MOUNT PMI L/A	6	0.14	0
DFDRG4	76	48	ZABC	Y	5500	BOND	7	1.98	127764
DFDRG4	76	48	ZABC	Y	4800	PMI L/A	8	0.03	6636
DFDRG4	76	48	ZABC	Y	5700	MOLD	9	0.16	6468
DFDRG4	76	48	ZABC	Y	5720	MOLD PMI	10	0.01	0
DFDRG4	76	48	ZABC	Y	5750	MOLD CURE	11	0.27	13440
DFDRG4	76	48	ZABC	Y	6050	SYMBOL 2	12	0.13	53760
DFDRG4	76	48	ZABC	Y	6100	TRIM/FORM	13	0.19	13272
DFDRG4	76	48	ZABC	Y	6901	ASSY STAGING	14	0.09	0
DFDRG4	76	48	ZABC	Y	7100	FINAL TEST 1	15	1.53	483672
DFDRG4	76	48	ZABC	Y	7777	OUTLIER REQUIREMENTS	16	0.02	0
DFDRG4	76	48	ZABC	Y	6110	SYMBOL 3	17	0.39	294658

Table 1: Sample of WIPBegin.csv

Device	Prod line	Pin	Tester	Strip test	Date	Plan starts	Plan ship out	Capacity
DFDRG4	76	48	ZABC	Y	8/11/2016	35071	55552	645105
32PWR	76	23	ZABC	Y	8/11/2016	264980	138584	617419
47PHPR	21	48	ZABC	N	8/11/2016	359808	178664	595514
6PWRG4	76	27	ZADR	Y	8/11/2016	115576	14523	626145
6QDRCM	61	27	ZABC	Y	8/11/2016	57794	24969	624012
25PWPR	76	23	ZADR	Y	8/11/2016	141507	48926	757320
DFDRG4	76	48	ZABC	Y	8/12/2016	175355	55552	610369
32PWR	76	23	ZABC	Y	8/12/2016	198735	138584	728215
47PHPR	21	48	ZABC	N	8/12/2016	359808	178664	660960
6PWRG4	76	27	ZADR	Y	8/12/2016	115589	14523	592934
6QDRCM	61	27	ZABC	Y	8/12/2016	86691	24969	632810
25PWPR	76	23	ZADR	Y	8/12/2016	113205	48926	679675
DFDRG4	76	48	ZABC	Y	8/13/2016	113981	55552	645846
32PWR	76	23	ZABC	Y	8/13/2016	0	138584	641662
47PHPR	21	48	ZABC	N	8/13/2016	359808	178664	631590
6PWRG4	76	27	ZADR	Y	8/13/2016	101140	14523	718785
6QDRCM	61	27	ZABC	Y	8/13/2016	0	24969	676163
25PWPR	76	23	ZADR	Y	8/13/2016	56603	48926	614624

Table 2: Sample of WIPPlanStart.csv

The data in the "WIPBegin.csv" and "WIPPlanStart.csv" files are mostly sufficient to run our planning model, which provides guidance for daily run rates for the upcoming two to four weeks. To compare our solutions with actual production levels, an additional dataset contained in the "WIPActual.csv" file is needed. Table 3 identifies the elements in this file. The new data include actual DRRs for each device and operation, as well as WIP levels at the beginning of each day. Note that "Actual begin WIP" is exactly the end WIP of the previous day.

Device	Prod line	Pin	Tester	Strip test	Dates	LPT	LPT order	Actual begin WIP	Actual DRR
DFDRG4	76	48	ZABC	Y	8/11/2016	5100	1	0	223176
DFDRG4	76	48	ZABC	Y	8/11/2016	5105	2	298193	53896
DFDRG4	76	48	ZABC	Y	8/11/2016	5110	3	0	53896
DFDRG4	76	48	ZABC	Y	8/11/2016	5200	4	94001	93996
DFDRG4	76	48	ZABC	Y	8/11/2016	5250	5	35952	129948
DFDRG4	76	48	ZABC	Y	8/11/2016	5300	6	0	129948
DFDRG4	76	48	ZABC	Y	8/11/2016	5500	7	127764	72240
DFDRG4	76	48	ZABC	Y	8/11/2016	4800	8	6636	72240
DFDRG4	76	48	ZABC	Y	8/11/2016	5700	9	6468	71976
DFDRG4	76	48	ZABC	Y	8/11/2016	5720	10	0	71976
DFDRG4	76	48	ZABC	Y	8/11/2016	5750	11	13440	46776

Table 3: Sample of WIPActual.csv

To account for the cycle time of each operation at the beginning of the planning horizon, it is necessary to know the daily run rates on several days prior. These data are contained in the "DRRInitial.csv" file, as illustrated in Table 4. The number of days of data in this file is determined by the maximum cycle time over all logpoints. For example, if the maximum cycle time is 2.6 days, then three days of data are required.

Device	Prod line	Pin	Tester	Strip test	Dates	LPT	LPT order	Actual DRR
DFDRG4	76	48	ZABC	Y	8/8/2016	5100	1	167777
DFDRG4	76	48	ZABC	Y	8/8/2016	5105	2	107732
DFDRG4	76	48	ZABC	Y	8/8/2016	5110	3	0
DFDRG4	76	48	ZABC	Y	8/8/2016	5200	4	87360
DFDRG4	76	48	ZABC	Y	8/8/2016	5250	5	136920
DFDRG4	76	48	ZABC	Y	8/8/2016	5300	6	136920
DFDRG4	76	48	ZABC	Y	8/8/2016	5500	7	249228
DFDRG4	76	48	ZABC	Y	8/8/2016	5600	8	249228
DFDRG4	76	48	ZABC	Y	8/8/2016	5700	9	260820
DFDRG4	76	48	ZABC	Y	8/8/2016	5720	10	267372

Table 4: Sample of DRRInitial.csv

Chapter 4: Mathematical Models

Our first formulation, denoted by Model-I, is a simplification of the actual AT process because it assumes a 1-period delay between successive logpoints. † This means that if device i undergoes operation j in period p, then the device will be available for processing at operation $j + 1$ (logpoint $j + 1$) at the beginning of period $p + 1$. This assumption proved too optimistic because it ignores the individual cycle times. Our second and third formulations, denoted by Model-II and Model-III, allow for multiple period and fractional period delays, respectively, and proved much more accurate.

4.1 SINGLE PERIOD DELAY MODEL WITH AVERAGE CYCLE TIME: MODEL-I

Model-I is a series of equalities and constraints that track WIP at each logpoint from one day to the next and impose limits on their maximum levels. It also bounds daily throughput based on installed capacity. This is done in part by ensuring flow balance for each device group at each operation. Although the ultimate goal of the system is to meet daily output targets for each device group, this may not be possible if the targets are treated as hard constraints. To avoid infeasible instances, we introduce a set of shortage and surplus variables for the corresponding constraints and minimize the weighted sum of their values over the planning horizon.

In the development of the three models, we make use of the following notation.

Indices and sets

I set of device groups; $i \in I$

D set of days in planning horizon; $d \in D$

J set of operations (logpoints), $j \in J$

$J(i)$ set of operations for device group i; $j \in J(i)$

J_i^{End} index for last operation in $J(i)$; that is, last operation in route of device group i

P set of periods in a day, $p \in P$

† Zhang, C., Bard, J. F., & Chacon, R. (2017). Controlling work in process during semiconductor assembly and test operations. *International Journal of Production Research*, 1-25. In this previous work, I was responsible for modeling, programming, data analysis, and testing.

Data and parameters

$ACT\_WIP_{i,j,0}$ actual WIP levels of device group i at operation j at end of day 0 (beginning of day 1)

CT_TGT_j cycle time target for operation j (average value of cycle time over planning horizon of devices in a group at operation j)

Cap_Lim_{jd} capacity limit for operation j on day d

$STARTS_{id}$ number of starts for device group i on day d at operation 1

$DOQR_{id}$ daily output quantity required for device group i on day d

Δ_j^{WIP} upper bound on percentage by which daily WIP can exceed its target at operation j (e.g., $\Delta_j^{WIP} \leq 5\%$)

δ_{ij} 1 if j is the first logpoint in $J(i)$, 0 otherwise

α objective function penalty weight for shortages

β objective function penalty weight for surpluses

Initial conditions

$WIP_{i,j,0}^{Daily}$ WIP level for device group i at operation j at the start of the analysis, that is, for $d = 0$

Decision variables

RR_{ijd}^{Daily} daily run rate for device group i at operation j on day d (to meet demand and minimize daily WIP variation)

RR_{ijpd}^{Period} period run rate for device group i at operation j in period p on day d

WIP_{ijd}^{Daily} WIP level for device group i at operation j on day d (measured at end of day d or beginning of d+1)

WIP_{ijpd}^{Period} WIP level for device group i at operation j at end of period p on day d

WIP_TGT_{jd} WIP target for operation j on day d

Penalty variables

s_{id}^{TP+} shortage for device group i at its final operation j_i^{END} on day d with respect to the target value $DOQR_{id}$

s_{id}^{TP-} surplus of device group *i* at its final operation j_i^{END} on day *d* with respect to the target value $DOQR_{id}$

Model-I

Minimize $\sum_{i\in I}\sum_{d\in D}\left(\alpha\cdot s_{id}^{TP+}-\beta\cdot s_{id}^{TP-}\right)$ (1a)

Subject to

Period flow balance for each operation and device group ($p = 1$)

$\delta_{ij}\cdot STARTS_{id} + WIP_{i,j,|P|,d-1}^{Period} + RR_{i,j-1,|P|,d-1}^{Period} - RR_{i,j,1,d}^{Period} = WIP_{i,j,1,d}^{Period}, \quad \forall\, i\in I, j\in J(i), d\in D$ (1b)

Period flow balance for each operation and device group ($p \neq 1$)

$WIP_{i,j,p-1,d}^{Period} + RR_{i,j-1,p-1,d}^{Period} - RR_{ijpd}^{Period} = WIP_{ijpd}^{Period}, \quad \forall\, i\in I, j\in J(i), p\in P\backslash\{1\}, d\in D$ (1c)

Calculation of daily run rate

$RR_{i,J_i^{End},d}^{Daily} = \sum_{p\in P} RR_{i,J_i^{End},p,d}^{Period}, \ \forall\, i\in I, d\in D$ (1d)

Final operation in route for each device group

$RR_{i,J_i^{End},d}^{Daily} + s_{id}^{TP+} - s_{id}^{TP} = DOQR_{id}, \ \forall\, i\in I, d\in D$ (1e)

Capacity limit per day for each operation

$\sum_{i\in I}\sum_{p\in P} RR_{ijpd}^{Period} \le Cap\_Lim_{jd}, \ \forall\, j\in J, d\in D$ (1f)

Calculation of WIP at end of day

$WIP_{ijd}^{Daily} = WIP_{i,|P|,j,d}^{Period}, \quad \forall\, i\in I, j\in J(i), d\in D$ (1g)

WIP target per day for each operation

$WIP\_TGT_{jd} = CT\_TGT_j\cdot\sum_{i\in I} RR_{ijd}^{Daily}, \quad \forall\, d\in D, j\in J$ (1h)

WIP limits per day for each operation

$\sum_{i\in I} WIP_{ijd}^{Daily} \le \left(1+\Delta_j^{WIP}/100\right)\cdot WIP\_TGT_{jd}, \quad \forall\, d\in D, j\in J$ (1i)

Initial WIP conditions on day 0 and production rates on day d for operation "0"

$WIP_{i,j,|P|,0}^{Period} = ACT\_WIP_{i,j,0}, \quad \forall\, i\in I, j\in J(i), \quad RR_{i,0,p,d}^{Period} = 0, \ \forall\, i\in I, p\in P$ (1j)

Variable definitions

$WIP_{ijd}^{Daily}\ge 0,\ WIP_{ijpd}^{Period}\ge 0,\ WIP\_TGT_{jd}\ge 0,\ RR_{ijd}^{Daily}\ge 0,\ RR_{ijpd}^{Period}\ge 0,$

$s_{id}^{TP+}\ge 0,\ s_{id}^{TP-}\ge 0 \ \forall\, i\in I, d\in D, j\in J(i)\cup\{0\}, \quad p\in P\cup\{0\}$ (1k)

The objective function (1a) minimizes the weighted sum of the total shortage and surplus associated with daily WIP output targets. The variable s_{id}^{TP+} will be positive for the pair (i,d) in any solution when it is not possible to satisfy the demand for device group i on day d. The variable s_{id}^{TP-} will be positive for the pair (i,d) in any solution in which there is more output than required. Because shortages are more critical than surpluses, the penalty weights must be set such that $\alpha >> \beta$.

Constraints (1b) conserve flow of WIP for the first operation in the first period of the day. The first term on the left-hand side represents the number of starts for device group i on day d. Starts only occur in period 1 and at operation 1. The second term, $WIP_{i,1,|P|,d-1}^{Period}$, is the calculated WIP for device group i at operation 1 at the end of the last period of the previous day (that is, the end of day d–1). The third term, $RR_{i,j-1,|P|,d-1}^{Period}$, is the upstream run rate for operation j–1 of the previous day. The fourth term, $RR_{i,j,1,d}^{Period}$, is the quantity run for device group i on day d at operation j. This represents the throughput in period 1 for that device group on the current day. The right-hand side of the constraint is the WIP level; that is, the amount of WIP for device group i at operation j at the end of period 1.

Constraints (1c) represent general flow balance for the remaining periods and operations. The assumption is that all devices processed in the current period are transferred to their next operation and are available for processing in the following period. The daily run rate (throughput) is calculated in (1d) by summing the period run rate, $RR_{i,J_i^{End},p,d}^{Period}$, over all periods on day d at the last operation J_i^{End}.

Constraints (1e) embed the output requirements for each device group i on day d. The two penalty variables, which are minimized in (1a), are needed to ensure a feasible solution is obtainable. On some days, there may be insufficient WIP in the system to meet the output requirements. Constraints (1f) limit the throughput at operation j to the capacity of the facility on day d. On some days, machines may be down for maintenance so it is necessary to consider each day separately although capacity is likely to be constant over the planning horizon.

Constraints (1g) are for accounting purposes and set the WIP for operation j at the end of day d to the WIP at the end of period $|P|$ on that day. Constraints (1h) compute daily WIP targets for each operation. This equation is based on Little's law which says that the queue length is equal to the product of the processing rate and the cycle time. Constraints (1i) limit WIP levels at operation j on day d to a given percentage Δ_j^{WIP} above the WIP target for that operation. Although we would like to maintain constant WIP levels, fluctuations in demand may require increased levels at some operations if shortages are to be avoided.

Equations (1j) set the initial conditions for WIP on day" 0," or equivalently, the first day of the planning horizon, to the actual WIP in the system. We also set $RR_{i,0,p,d}^{Period} = 0$ for operation "0" and all device groups, periods and days. Variable definitions are given in (1k).

4.2 MODEL WITH INTEGER CYCLE TIME: MODEL-II

Model-I is based on the assumption that all devices processed in the current period are available for the next operation in the following period. In reality, the cycle time of device i at operation j, call it $C_{i,j}$, determines when processing is finished and the device can be transferred to the next operation. The way we considered this delay behavior in Model-I was to assume steady-state conditions, and rather than explicitly include transfer delays based on actual values, we simply used the average cycle time CT_TGT_j to compute WIP targets, as indicated by constraints (1e) and (1f). However, the cycle time may have large variance among device groups, which may lead constraints (1e) and (1f) to enforce inaccurate restrictions on production and WIP levels. To avoid this situation, Model-II includes the individual cycle times for each device group and embeds them in the flow balance constraints (1c).

In the formulation, cycle time $C_{i,j}$ is considered to be an integral number of periods (it can be fractional in terms of days). To achieve integrality, we replace $C_{i,j}$ with its ceiling, call it $U_{i,j}$. Thus, devices processed at the current logpoint are will be available after $U_{i,j}$ periods. For example, if the cycle time is 2.41 days and the period length is 0.1, i.e., 10 periods per day, we

have $C_{i,j} = 24.1$ and $U_{i,j} = \lceil C_{i,j} \rceil = 25$. With this adjustment, cycle time can now be accounted for in the flow balance equations so the stead-state constraints (1g) and (1h) can be removed.

Model-II

Minimize $\sum_{i \in I} \sum_{d \in D} \left(\alpha \cdot s_{id}^{TP+} - \beta \cdot s_{id}^{TP-} \right)$ (2a)

Subject to (1d), (1g) – (1k)

Period flow balance for each operation and device group ($p = 1$)

$$\delta_{ij} \cdot STARTS_{id} + WIP_{i,j,|P|,d-1}^{Period} + RR_{i,j-1,1-U_{i,j-1},d}^{Period} - RR_{i,j,1,d}^{Period} = WIP_{i,j,1,d}^{Period}, \quad \forall\, i \in I, j \in J(i), d \in D \tag{2b}$$

Period flow balance for each operation and device group ($p \neq 1$)

$$WIP_{i,j,p-1,d}^{Period} + RR_{i,j-1,p-U_{i,j-1},d}^{Period} - RR_{ijpd}^{Period} = WIP_{ijpd}^{Period}, \quad \forall\, i \in I, j \in J(i), p \in P \setminus \{1\}, d \in D \tag{2c}$$

Final operation in route for each device group

$$RR_{i,J_i^{End},d}^{Daily} + s_{id}^{TP+} - s_{id}^{TP} = DOQR_{id}, \forall\, i \in I, d \in D \tag{2e}$$

Constraints (2b) embed cycle time into the third term $RR_{i,j-1,1-U_{i,j-1},d}^{Period}$ which represents the upstream run rate from the previous logpoint for $p = 1$. Constraints (2c) are the general cases for $p \in P \setminus \{1\}$.

Notice that whenever $p - U_{i,j} \leq 0$, $RR_{i,j-1,p-U_{i,j-1},d}^{Period}$ must be replaced by the run rate from an earlier day; that is, we need to replace the index $p - U_{i,j}$ with $p + \eta |P| - U_{i,j}$ and the index d with $d - n$ in $RR_{i,j-1,p-U_{i,j-1},d}^{Period}$ to get $RR_{i,j-1,p+n|P|-U_{i,j-1},d-n}^{Period}$, where $n = \min_{\eta \in \mathbb{I}_+} \{\eta : p + \eta |P| - U_{i,j} > 0,\ \eta < d\}$. This value of n assures that devices processed at operation j on day $d - n$ are available in the current period. It could be possible that no such n exists, especially for the first several days in the planning interval. In that case, the upstream devices come from production that took place prior to the starting date ($d = 1$) listed in DRRInitial.csv (see Table 4). In constraints (2e), which duplicate (1e), consider the same case for day d such that $d \cdot |P| - U_{i,j_i^{End}} \leq 0$. Here, the output for device group i for day d is obtained from the last operation J_i^{End} in the column labeled "Actual DRR" in DRRInitial.csv. See Appendix A for more discussion of this issue.

4.3 MODEL WITH FRACTIONAL CYCLE TIME: MODEL-III

In practice, the cycle time for an operation can be any real number rather than an integer or a multiple of a fixed period. The datasets provided by Texas Instruments, for example, specify cycle time in days to 2 decimal places. If a day is divided into, say, 100 periods, then every $C_{i,j}$ will be integral, i.e., $U_{i,j} = C_{i,j}$. Devices processed at the most recent logpoint will arrive at the current logpoint after exactly $C_{i,j}$ periods. However, taking the ceiling of $C_{i,j}$ introduces an inappropriate delay in material transfer, especially when the number of periods is small. For example, if the day is divided into 10 periods and the cycle time for a particular logpoint is 0.62 (days), then $C_{i,j} = 6.2$ periods. Devices processed in the current period will be available after 6.2 periods. When Model-II is used, we have $U_{i,j} = \lceil C_{i,j} \rceil = 7$ so there will be an additional delay of 0.8 periods at this logpoint. In reality, this delay does not occur.

A natural way to model fractional cycle times is to treat them as continuous. If we assume that processing is uniformly distributed over each period, then any production from a sub-interval *l* of a period can be calculated by the length of *l* denoted by | *l* |. This idea is motivated by the work of Leachman et al. (1996).

Figure 3 depicts an example of material transfer from an upstream station to a downstream station for a cycle time of 2.4 periods. Considering Model-II, suppose we are in period 1 and $C_{i,j} = 2.4$ and $U_{i,j} = 3$ for operation *j*. The implication is that all devices processed in period 1 will be transferred uniformly to their next operation *j* + 1 in period 4 (upper portion of Figure 3). However, because the cycle time is really fractional, operation *j* will be finished 0.6 periods earlier than assumed in the model. What really happens under the uniform transfer assumption is that devices processed in the first 60% of period 1 are available in period 3 while the remaining 40% are transferred to period 4 as upstream inflows (bottom diagram in Figure 3).

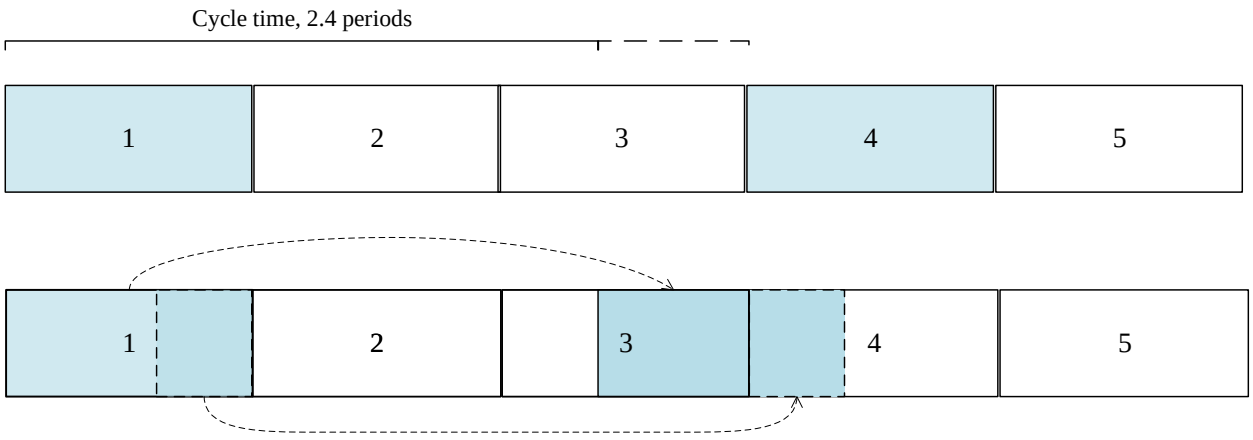

Figure 3: Integer cycle time transfer (above) vs. fractional cycle time transfer (below).

To account for fractional cycle times as suggested by this analysis it is necessary to modify constraints (2c). Run rates form upstream logpoints are now divided into two terms corresponding to the two fractional components of the cycle time. Letting $L_{i,j} = U_{i,j} - 1$, we have:

Period flow balance for each operation and device group ($p \neq 1$)

$$WIP^{Period}_{i,j,p-1,d} + (U_{i,j} - C_{i,j}) \cdot RR^{Period}_{i,j-1,p-L_{i,j-1},d} + (C_{i,j} - L_{i,j}) \cdot RR^{Period}_{i,j-1,p-U_{i,j-1},d} - RR^{Period}_{ijpd} = WIP^{Period}_{ijpd}$$

$$\forall\, i \in I, j \in J(i), p \in P \backslash \{1\}, d \in D \qquad (3c)$$

Model-III is equivalent to Model-II but with Eqs. (2c) replaced with (3c), which leads to the following.

Proposition 1. When the cycle time of every logpoint is an integer multiple of the period length, Model-II and Model-II are identical.

Proof. When the cycle time at every logpoint is an integer multiple of period length, we have $U_{i,j} - C_{i,j} = 0$. Thus, the left-hand side of Eq. (3c) reduces to

$$WIP^{Period}_{i,j,p-1,d} + RR^{Period}_{i,j-1,p-U_{i,j-1},d} - RR^{Period}_{ijpd} = WIP^{Period}_{ijpd}$$

which demonstrates the equivalence of the models. ■

Remark. When period length satisfies *Proposition 1*, Model-II and Model-III are guaranteed to provide identical objective function values; however, the corresponding decision variables (DRR, WIP, and so on) may differ due to the presence of multiple optimal solutions.

For the TI datasets in which the cycle time is given to 2 decimals, dividing the day into 100 periods (0.01 day per period) will satisfy *Proposition 1*. This observation is empirically demonstrated in Section 6.

Chapter 5: Implementation

The sequence of steps associated with the implementation of our model is depicted in Figure 4 and includes preprocessing, data input, model generation, solution, and output tabularization.[†] The first step is to preprocess the raw data as discussed in Section 3 to establish the device groups, logpoints associated with each, and the number of days in the planning horizon. The size of the model is determined by $|I|,|J|,|D|$, the number of device groups, operations, and days, respectively. Next we build a distinct logpoint sequence for each device group and save the data needed to construct the model, such as demand, initial WIP levels for each device group, logpoints, and dates. At this point, the solver is called and the results are formatted and written to several files.

Input files. Table 5 lists the four datasets created in the preprocessing step along with a fifth file called "input.txt" that specifies a set of parameter values required to build and run the model. These include the number of periods in the model, how capacity is to be treated (a single value for all operations or individual values), a capacity multiplier factor, and objective function weights α and β. Table 6 presents the input.txt file used in our analysis.

[†] Zhang, C., Bard, J. F., & Chacon, R. (2017). Controlling work in process during semiconductor assembly and test operations. *International Journal of Production Research*, 1-25. In this previous work, I was responsible for modeling, programming, data analysis, and testing.

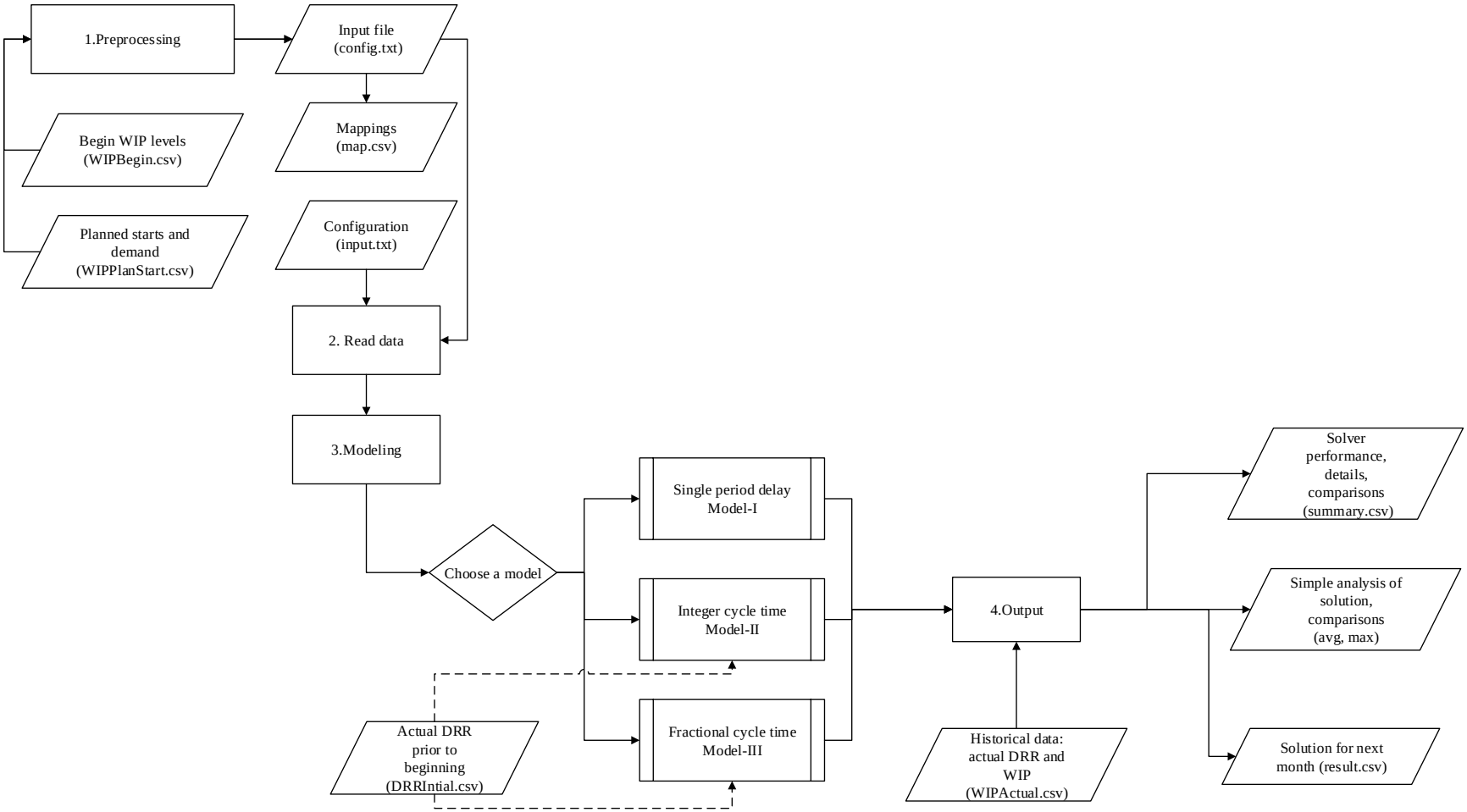

Figure 4: Design of implementation

File name	Description
WIPPlanStart.csv	Specifies device group, LPT, date, *STARTS*, and *DOQR*
WIPBegin.csv	Specifies device group, LPT, date, *CTTGT* and begin WIP
WIPActual.csv	Specifies device group, LPT, date, Actual DRR and Actual Begin WIP
DRRIniitial.csv	Specifies device group, LPT, date, Actual DRR for previous days.
input.txt	Configuration file that specifies model control parameter values

Table 5: Input data files

Value	Description
100	Number of periods per day
1	Use planned starts; 1-use, 0-otherwise
1	Use constant capacity; 1-use, 0-otherwise
690000	Capacity size when using constant capacity; must be positive
1.1	Capacity multiplier factor; model capacity = (factor)*(capacity)
10	Objective function weight parameter α for shortage; 0-no penalty
1	Objective function weight parameter β for surplus; 0-no penalty
1	Comparison parameter; 1-compare solution to historical data, 0-forecast only

Table 6: Sample of "input.txt" file

5.1 PREPROCESSING AND DATA INPUT

The first step is to read the first two data files in Table 5 and construct individual sets containing device groups, days, and operations. The cardinality of each of these sets is also determined. Because it is more convenient to model by the natural order of days and operations, *D* and *J* are sorted in ascending order. Then sets *I*, *D* and *J* are written to the file "map.csv," and saved along with the size of each in "config.txt." Pseudocode for the preprocessing procedure is given in Figure 5; pseudocode for data input is presented in Appendix B.

```
Procedure_Preprocessing
Description:      Preprocess data files and determines model size
Subprocedures:    DateSort, LPTSort
Input:            WIPPlanStart.csv: device group, LPT, date, STARTS, and DOQR
                  WIPBegin.csv: device group, LPT, date, CTTGT and begin WIP
Output:           config.txt: size of the model
                  map.csv: relationships between device group, date, LPT and their indices in sets I, D, J
Begin
Initialize three empty sets for device group, LPT, and date, respectively.
Define output file directory and name.
Process WIPPlanStart.csv
While (not EOF) {
    Get current row:
    row l = row_to_vector(current row)
    Add device group of this row to set I.
    Add date of this row d to set D.
  }
Process WIPBegin.csv
While (not EOF) {
    Get current row:
    row l = row_to_vector(current row)
    Add LPT of this row j to set J.
  }
Determine size of I, J, D and save to config.txt.
Close data files WIPPlanStart.csv and WIPBegin.csv.
Sort date set D in ascending order using a subroutine DateSort
(comment: translate date format "mm/dd/yyyy" to a numeric, then sort in ascending order)
Sort LPT in set J in ascending order using a subroutine LPTSort
(comment: sort in ascending order)
End
```

Figure 5: Preprocessing pseudocode

5.2 MODEL CONSTRUCTION AND SOLUTION PROCESS

The mathematical model is implemented in C++ using an open source library FlopC++ (Formulation of Linear Optimization Problems in C++) provided by the Coin-OR open-source project. The three models mentioned in Section 4 are written as three separate functions but the basic components are reused as much as possible. All datasets created during preprocessing are translated into the appropriate format within the modeling procedure.

5.2.1 Anomalies in the constraints

Nonincreasing logpoint sequence. It is not uncommon for the numerical designation of operations for a device group to be non-sequential. For example, one sequence we found was: 5100 –> 5105 –> 5110 –> 5200 –> 5400 –> 5250 –> 5300 –> 5500 –> 5501 –> 5600 –> ··· –> 9900, where 5100 represents "Starts" and 9900 represents "Ship out." Notice that after logpoint 5400 we have 5250, a decrease. This creates a problem because the elements of the logpoint set J are sorted in ascending order. In constraints (1b), inflow from a previous operation, denoted by $DRR_{i,d-1,j-1}$, would be incorrect for logpoint 5250 if the nature order were followed. To deal with this situation we developed a map denoted by *seq* such that *seq*: $I \times N \to J$, where $N = \{1, 2, 3, ..., |J|\}$ and $seq(i,k) = j$ if j is the k^{th} logpoint of device group i and 0 otherwise. This formulation allows us to manage the contstruction of the flow balance equations with a matrix of size $|I| \times |J|$.

As an example, in the sequence listed above, $seq(i,1) = 5100$, $seq(i,2) = 5105$, and so on. The flow balance constraints (2c) for device group i with the index k replacing j is as follows.

$$WIP^{Period}_{i,seq(i,k),p-1,d} + RR^{Period}_{i,seq(i,k-1),p-U_{i,j-1},d} - RR^{Period}_{i,seq(i,k),p,d} = WIP^{Period}_{i,seq(i,k),p,d}, \quad \forall\, i \in I, k \in N, p \in P \setminus \{1\}, d \in D.$$

Since the daily run rate DRR is summed over j, we can set the upper bounds to zero for those operations that are not in the logpoint sequence of i. This can be done by searching for j over $k \in N$. If there doesn't exist a k such that $seq(i,k) = j$, then the upper bound for $RR^{Period}_{i,seq(i,k),p,d}$ is set to be zero. See Appendix A for more discussion of this issue.

Initialization for flow constraints. Model-II and Model-III use cycle time in the indices of the flow constraints, so it is necessary to make an adjustment to $RR^{Period}_{i,j-1,p-U_{i,j-1},d}$ for first few days when $p - U_{i,j} < 0$. In such cases, these variables must be initialized with data provided in "DRRInitial.csv" corresponding to the actual DRRs coming from one or more days prior to the starting date. See Appendix A for more discussion.

5.3 OUTPUT

Results from the solver are saved as vectors or arrays in the three files referenced in Table 7 in a suitable form for analysis. Aggregate results are given in the "results.csv" file which contains the values of the variables for each device group at each logpoint for each day of the planning horizon. The "summary.csv" highlights the input parameter values and computational statistics, and presents the average and maximum values for DRR and WIP by device group. The most detail is given in the "periodreslts.csv" file which contains all the output for each period in the planning horizon.

File name	Description
results.csv	Daily solutions for each device group and logpoint
summary.csv	Information related to parameter values and computations, e.g., objective function, CPU time
periodresults.csv	Solutions for each device and logpoint for each period

Table 7: Output files

Table 8 is an example of the "summary.csv" file. The first few rows specify the run date and time, the number of device groups, the number of days in the planning horizon, CPU time, and so on. "SOLVER_STATUS" tells if the model is optimal, infeasible or unbounded. The rows that follow compare model results with actual values ("ACT" prefix) with respect to the objective function, total device output, shortages, and surpluses. Note that the objective function value is the weighted sum of shortage and surplus with weights α and β. "CAP_PERCENT" is a multiplier for "DAILY_CAPACITY" that reflects the parameter values listed in input.txt.

The lower part of the table presents the actual average DRR, the average DRR and Maximum DRR obtained from the model, the actual average WIP, and the average WIP obtained from the model over the planning horizon for each device group. These results are intended to provide guidelines for shop floor supervisors. The computed production rates and WIP should be viewed as targets and not as actual plans. Given the dynamics of the system and the inevitability of real-time disruptions, it may not be practical or even possible to implement the

period by period results contained in the "periodreslt.csv" file. In fact, these results represent aspirational levels, achievable if all goes well. However, unforeseen disruptions such as transfer delays, machine breakdowns, WIP shortages along a route, and changing priorities can undermine any finely tuned plan. When it is not possible to achieve production targets on some days for one device group, though, it may be possible to boost production up to the maximum value (MAX_DRR) of another group to compensate. Finally, by examining the two columns labeled "ACT_AVG_DRR" and "AVG_DRR," we can see the high level relationship between increased production and a decrease shortages.

DATE & TIME =	Thursday, September 1, 2016; 10:39:35		
DEVICE_GROUPS =	6	DAYS =	14
START_DATE =	8/11/2016	END_DATE =	8/24/2016
PERIODS =	100	TIME_UNIT (hr) =	0.24
SOLVER_STATUS =	OPTIMAL	CPU_TIME (sec) =	295
OBJECTIVE =	1.17E+07	ACT_OBJECTIVE =	2.11E+07
OUTPUT =	7.90E+06	ACT_OUTPUT =	7.35E+06
SHORTAGE =	1.46E+06	ACT_SHORTAGE =	1.84E+06
SURPLUS =	2.90E+06	ACT_SURPLUS =	2.74E+06
CAP_PERCENT =	1.1		
DAILY_CAPACITY =	690,000		

DEVICE_GROUP	ACT_AVG_DRR	AVG_DRR	MAX_DRR	ACT_AVG_WIP	AVG_WIP
21-48-ZABC-N	158632	254788	354672	94080.4	13605.2
61-27-ZABC-Y	33038.9	33750.7	50547.7	11401	7994.67
76-23-ZADR-Y	29948.7	53361	84837.5	12248.2	7182.8
76-23-ZABC-Y	158830	141091	162894	75749.5	27607.7
76-27-ZADR-Y	32645.8	32011	55763.1	11973.3	6804.04
76-48-ZABC-Y	80116.2	104509	156568	41666.8	2103.58

Table 8: An example of "summary.csv"

Table 9 (given two parts) is an example of the "results.csv" file. Statistics listed include DRR values, end-of-day WIP levels, actual DRR, actual WIP, demand for each device, operation, and date. Shortage and surplus values for each day are calculated at the last logpoint in the route for each device group and compared to the actual values (when known). Actual shortage and

surplus values are calculated as the difference between daily demand DOQR and DRR also at the last operation in a route.

DG index i	DEVICE GROUP	Day index d	DATE	LPT index j	LPT	STARTS(i,d,j)	CT(i,j)	DOQR(i,d)	DRR(i,d,j)
1	[21-48-ZABC-N]	1	8/11/2016	1	5100	359808	0.09		276194
1	[21-48-ZABC-N]	1	8/11/2016	2	5105	0	1.11		137844
1	[21-48-ZABC-N]	1	8/11/2016	3	5110	0	0.77		58321
1	[21-48-ZABC-N]	1	8/11/2016	4	5200	0	2.75		159116
1	[21-48-ZABC-N]	1	8/11/2016	5	5400	0	0.7		141433
1	[21-48-ZABC-N]	1	8/11/2016	6	5250	0	0.23		217033
1	[21-48-ZABC-N]	1	8/11/2016	7	5300	0	0.1		105754
1	[21-48-ZABC-N]	1	8/11/2016	8	5500	0	1.89		164652
1	[21-48-ZABC-N]	1	8/11/2016	9	5501	0	1.98		213503
1	[21-48-ZABC-N]	1	8/11/2016	10	5600	0	0.01		315
1	[21-48-ZABC-N]	1	8/11/2016	11	5700	0	0.33		25515
1	[21-48-ZABC-N]	1	8/11/2016	12	5720	0	0.03		25515
1	[21-48-ZABC-N]	1	8/11/2016	13	5750	0	0.25		84132
1	[21-48-ZABC-N]	1	8/11/2016	14	6000	0	0.17		84132
1	[21-48-ZABC-N]	1	8/11/2016	15	6010	0	0.15		0
1	[21-48-ZABC-N]	1	8/11/2016	16	6901	0	0.09		0
1	[21-48-ZABC-N]	1	8/11/2016	17	7100	0	3.74		176495
1	[21-48-ZABC-N]	1	8/11/2016	18	7777	0	0.02		196771
1	[21-48-ZABC-N]	1	8/11/2016	19	9050	0	0.01		196772
1	[21-48-ZABC-N]	1	8/11/2016	20	9060	0	0.54		99434
1	[21-48-ZABC-N]	1	8/11/2016	21	9070	0	2.12		246901
1	[21-48-ZABC-N]	1	8/11/2016	22	9080	0	0.03		263901
1	[21-48-ZABC-N]	1	8/11/2016	23	9085	0	0.04		284901
1	[21-48-ZABC-N]	1	8/11/2016	24	9900	0	0.01	178664	231309

WIPTARGET(d,j)	WIPLEVEL(d,j)	WIP(i,d,j)	SURPLUS(i,d)	SHORTAGE(i,d)	ACTUALDRR(i,d,j)	ACTUALBEGIN WIP(i,d,j)	ACTUALEND WIP(i,d,j)	ACTUALSURPLUS(i,d)	ACTUALSHORT AGE(i,d)
37593	557046	83613			199125	0	0		
450485	132486	0			183845	359953	375233		
256967	578710	0			108721	58321	133445		
954146	366555	0			210000	327781	226502		
99003	0	0			192568	141433	157665		
94249	100776	0			242968	75600	25200		
90849	0	0			242968	30154	30154		
968008	105754	105754			184168	109200	168000		
422735	187922	187922			185134	201934	200968		
3451	247514	209896			185134	315	315		
30901	0	0			168334	25200	42000		
1674	0	0			168312	0	0		
42062	0	0			193343	58617	33586		
14302	0	0			176544	0	16799		
0	0	0			176521	0	0		
7242	0	0			176521	0	0		
824360	197464	94532			156459	826723	844806		
10628	0	0			164736	8277	0		
6496	0	0			164736	0	0		
85393	0	0			149958	91157	105935		
599059	0	0			123316	17768	55208		
8987	0	0			124000	17000	5000		
12823	0	0			145000	21000	0		
2597	0	0	52645	0	145000	0	0	0	33664

Table 9: An example of result.csv

Chapter 6: Computational Results

Testing was done on three datasets provided by Texas Instrument from calendar year 2016. Dataset 1 spans 27 days from June 8th to July 4th, with three extra days prior to June 8th used to initialize DRR values. Dataset 2 spans 33 days from July 5th to Aug 6th, with three extra days prior to July 5th used for initialization purposes, and dataset 3 covers the 14 days from Aug 11th to Aug 24th again with three extra days from Aug 8th to Aug 10th. All datasets have 6 device groups with the maximum number of logpoints being 27 for any group. In all, there are 32 unique logpoints in each dataset.

6.1 COMPARISON BASED ON HISTORICAL DATA

6.1.1 Summary of results

For the day divided into 20 periods and 100 periods, Tables 10(a) and 10(b) respectively present a comparison of objective function values, total output, shortage, and surplus values between the solutions obtained with the three models and the actual results in dataset 1. Tables 10(c) and 10(d) provide the same comparisons for dataset 2 while Tables 10(e) and 10(f) present comparisons for dataset 3. All computations were handled in a Linux environment.

Performance	Model-I	Model-II	Model-III	Actual data
Objective	8850930	33080100	29475600	59034800
Output	18129600	14635500	15274500	12582500
Shortage	1367570	3671470	3341960	5556790
Surplus	4824720	3634580	3944010	3466870

Table 10(a): Results for dataset 1, 20 periods

Performance	Model-I	Model-II	Model-III	Actual data
Objective	5704470	27338420	27337800	59034800
Output	18578300	15570800	15570800	12582500
Shortage	1067820	3138460	3137360	5556790
Surplus	4973690	4046180	4035780	3466870

Table 10(b): Results for dataset 1, 100 periods

Performance	Model-I	Model-II	Model-III	Actual data
Objective	13825300	29900000	28637500	58860000
Output	17949600	14500000	14492800	13576000
Shortage	1851960	3250000	3113670	5490120
Surplus	4694300	2630000	2499190	3958850

Table 10(c): Results for dataset 2, 20 periods

Performance	Model-I	Model-II	Model-III	Actual data
Objective	12086100	28000000	28000000	58860000
Output	18554600	14600000	14600000	13576000
Shortage	1725940	3060000	3060000	5490120
Surplus	5173280	2530000	2530000	3958850

Table 10(d): Results for dataset 2, 100 periods

Performance	Model-I	Model-II	Model-III	Actual data
Objective	-2775300	15301300	12777600	21125700
Output	9660000	7255420	7693690	7354500
Shortage	47517	1788850	1557140	1838930
Surplus	3250470	2587210	2793780	2736380

Table 10(e): Results for dataset 3, 20 periods

Performance	Model-I	Model-II	Model-III	Actual data
Objective	-3202950	11702200	11702200	21125700
Output	9660000	7900190	7900190	7354500
Shortage	0	1460590	1460590	1838930
Surplus	3202950	2903720	2903730	2736380

Table 10(f): Results for dataset 2, 100 periods

In the case of 20 periods (0.05 days per period), Model-III provides more accurate results than Model-II, which is not surprising. This advantage decreases as the number of periods

increases to 100. Idle time included in Model-II is eliminated when cycle times are integer multiples of the period length (see Proposition 1). In all three cases, the results from the models dominate actual performance. Model-I yielded the best results with respect to shortages, eliminating them altogether for dataset 3. However, since Model-I has only a single-period delay which ignores the cycle times that are mostly longer than 1 period, it is fair to say that the model is too optimistic.

Model-II and Model-III provide relatively more realistic results in the comparisons with the historical data. Both models show an increase in total output and a reduction in shortages, but not necessarily an increase in the surplus. For both models and the case with 100 periods, for example, the total output increased for dataset 1 by 3 million (23.7%) compared to actual output; for dataset 2, the increase is nearly 1 million (7.5%); and for dataset 3, it is 545,690 (7.4%). Moreover, a slight increase in output may have a noticeable effect on shortage and surplus values. Shortages decreased by 20% for dataset 3, and 44% for datasets 1 and 2. Because dataset 3 spans two weeks only, 40% is a reasonable estimate of the month's decrease in shortages.

6.1.2 Interpretation of solution

Improvements over current performance are only achievable if production closely adheres to the period run rate each day. It is unrealistic, however, to expect shop supervisors to convert model outcomes to an hour by hour plan without experiencing occasional disruptions. It is far more likely that they will be able to adopt daily averages rather than period values as production targets. Accordingly, we provide average DRR values for the planning horizon by device group and operation (e.g., see Table 8). These are intended to serve as a guide. The corresponding average WIP levels are also calculated for each device group and operation.

Based on the results from Model-II with datasets 2, let ActualAvgDrr and ModelAvgDrr be the average DRR from the historical data and the model, respectively. The average WIP levels are defined as ActualAvgWIP and ModelAvgWIP. Figure 6 plots these four values in different subgraphs for each device group across all logpoints.

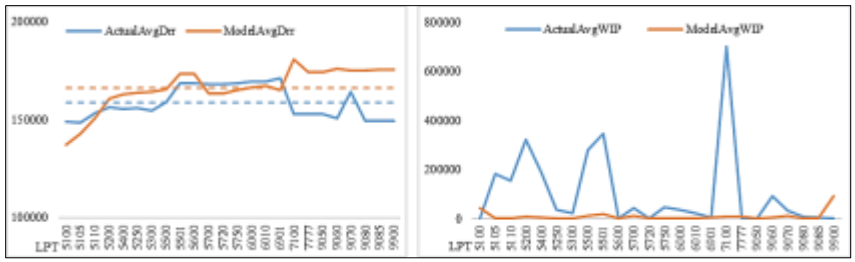

Figure 6(a): Average DRR and WIP for device group 1

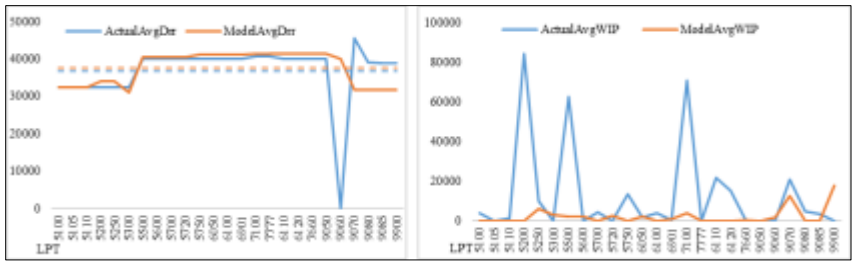

Figure 6(b): Average DRR and WIP for device group 2

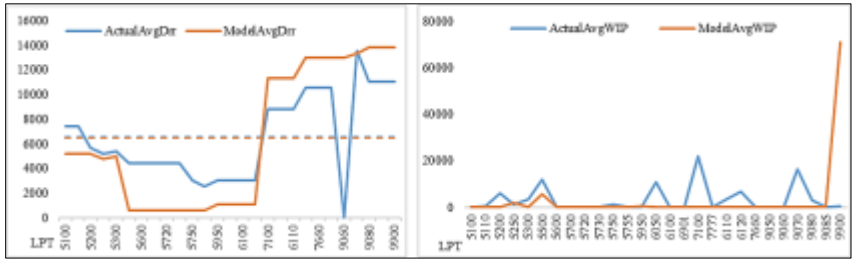

Figure 6(c): Average DRR and WIP for device group 3

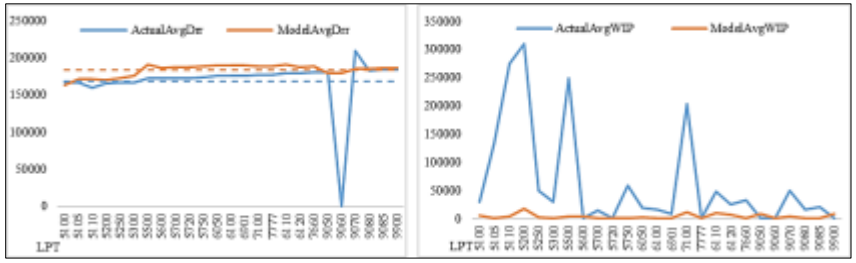

Figure 6(d): Average DRR and WIP for device group 4

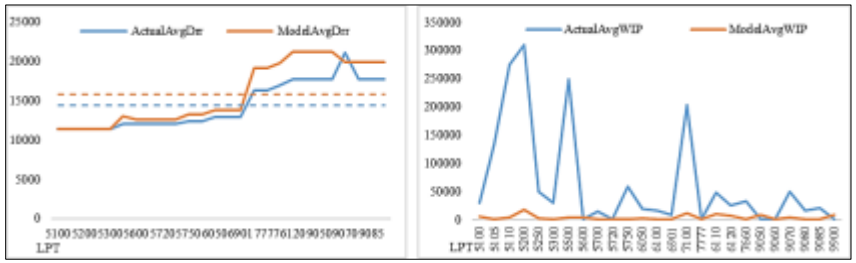

Figure 6(e): Average DRR and WIP for device group 5

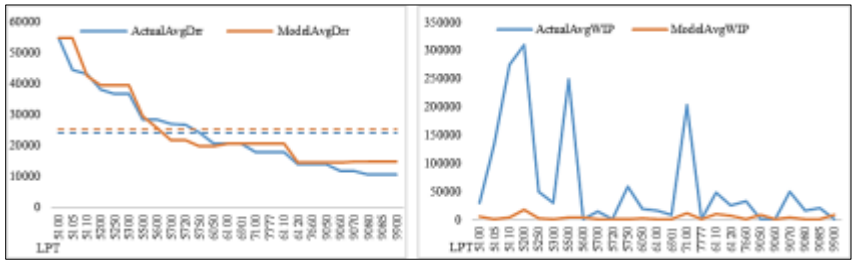

Figure 6(f): Average DRR and WIP for device group 6

For example, Figure 6(a) depicts the values for device group 1. The left subgraph shows the average DRR over the month at each logpoint for the actual and model results (lines are in different colors in the PDF), where the dashed lines represent the average over the logpoints. The right subgraph presents the average WIP at each logpoint over the month. For all plots in Figure 6, it can be seen that the model results slightly improve the actual DRRs while eliminating large variations in WIP levels. For planning purposes, then, the recommended strategy is for shop floor supervisors to aim for the grand average DRR values as represented by the dotted lines.

6.2 COMPARISON OF MODEL-II AND MODEL-III

6.2.1 Convergence

In this section, we present results for Model-II and Model-III for different numbers of periods to determine how granularity affects solution quality. As mentioned, Model-II is expected to be less accurate due to idle time induced in the flow balance constraints (2c). Taking into account, however, that the cycle time lengths $C_{i,j}$ are specified to two decimal places in the real datasets, as the number of periods in Model-II approaches 100, the results should become increasingly more accurate.

In the testing, we ran Model-II and Model-III with 10 to 100 periods using dataset 2. The results are presented in Table 11 and Figure 7 where it can be seen that shortages and surpluses converge to the same values. The same observation was made when we examined the bottom two lines in each portion of Table 11. This follows since Eq. (2c) corresponds to Eq. (3c) when the number of periods equals 100. Generally, Model-III is more accurate when a small number of periods is used, and convergence faster than Model-II.

Periods per day	Model-II shortage	Model-II surplus	Model-III shortage	Model-III surplus	Actual shortage	Actual surplus
10	3683210	2294130	3346770	2380760	5490120	3958850
20	3250000	2630000	3113670	2499190	5490120	3958850
30	3124270	2507420	3056380	2530320	5490120	3958850
50	3055280	2530280	3055770	2527780	5490120	3958850
70	3055280	2530280	3055280	2530280	5490120	3958850
100	3055280	2530280	3055280	2530280	5490120	3958850

Table 11: Comparison of Model-II and Model-III

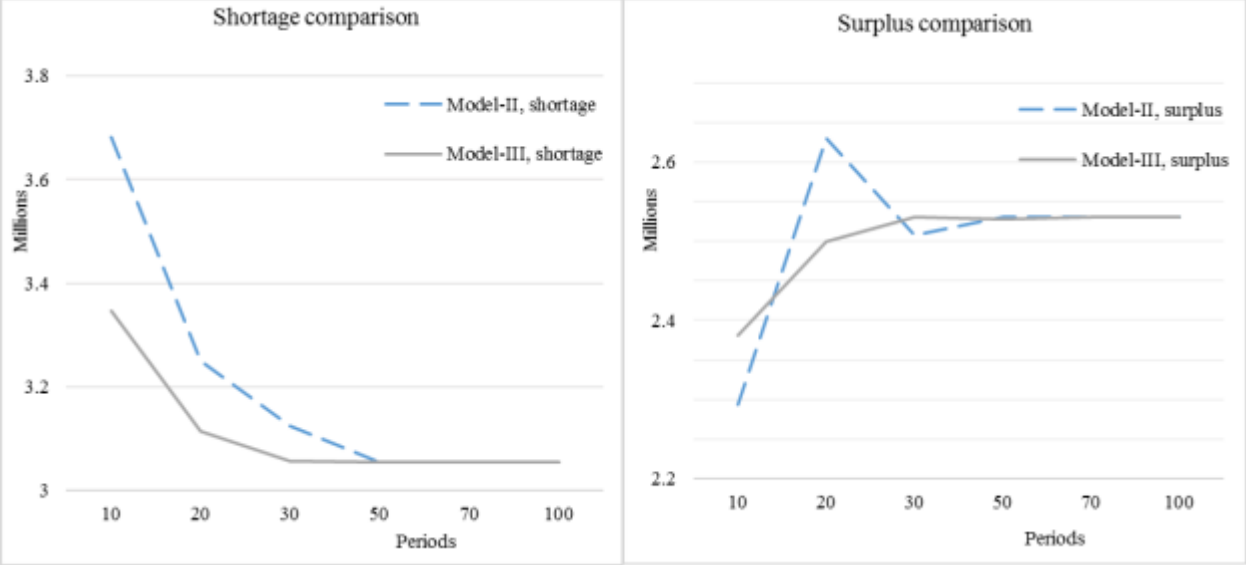

Figure 7: Comparison of Model-II and Model-III

6.2.2 Computation speed

One advantage that Model-II has over Model-III is that it usually runs much faster. Table 12 presents runtime comparisons on a 64-bit Linux PC, which confirms the above statement for any number of periods. The performance of Model-III was observed to be somewhat erratic, however, as indicated by the non-monotonic increase in CPU time. For 30 periods, for example, Model-III took 611 sec or 15.9% longer than for 50 periods, even though the latter instances have many more variables and constraints. We found this to be true when the number of periods was

less than 50 and did not divide evenly into 100. No theoretical justification for this observation is evident, though.

Periods	Model-II, CPU time (sec)	Model-III, CPU time (sec)
10	11	100
20	36	154
30	58	611
50	144	527
70	439	727
100	648	1530

Table 12: Runtime Comparison of Model-II and Model-III

Sparsity and integrality. It is generally known that network-type models like ours have sparse **A**-matrices. Looking at the equations in Sections 4.2 and 4.3, if we rearrange the terms by placing all the variables on left-hand side of their respective (in)equality signs, the constraints can be put in the form **Ax** (≤) = **b**. Here, the **x**-vector represents the decision variables *RR*, *WIP* and so on, and the **A**-matrix holds the corresponding parameter values.

Further examination of Model-II reveals that every entry of **A** is 0, –1, or 1, that is, $a_{ij} \in \{0,-1,1\}$. Also from Eq. (2c), **A** can be decomposed into and a banded block for RR_{ijpd}^{Period} and a bi-diagonal block for variables WIP_{ijpd}^{Period} (see Figure 8). Moreover, there are only 2 nonzero entries in one row of the banded block for RR_{ijpd}^{Period}. This translates into an extremely sparse structure for Model-II. In Model-III, the diagonal portion for $RR_{i,j-1,p-U_{i,j-1},d}^{Period}$ in Eq. (3c) is replaced with a bi-diagonal block for $RR_{i,j-1,p-L_{i,j-1},d}^{Period}$ and $RR_{i,j-1,p-U_{i,j-1},d}^{Period}$ that have fractional entries $U_{i,j} - C_{i,j}$ and $C_{i,j} - L_{i,j}$. The sparsity and absence of fractional entries in the **A**-matrix has much to do with the speed advantage of Model-II over Model-III.

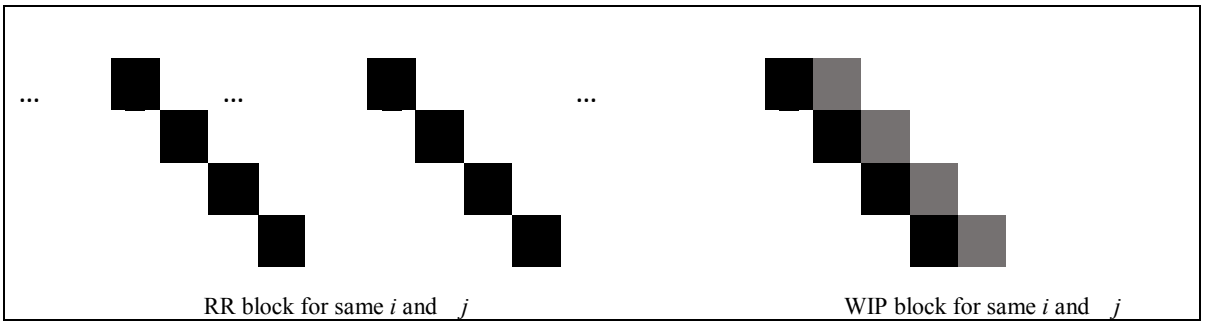

Figure 8: Matrix illustration for sparsity of Model II

For any choice of periods, Model-II rounds all cycle times to integers thus preserving the sparse structure. In contrast, Model-III is less robust due to fractional entries in the **A**-matrix which increase the computational effort, especially when the fractional portion of $C_{i,j}$ is small.

Chapter 7: Summary and Conclusion

In this paper, we first present a single period delay model (Model-I) that assumes any WIP being processed in the current period will be finished at the end of the period and immediately transferred to the next downstream operation in its route. Cycle times were accounted for by imposing steady-state conditions with the use of a constraint derived from Little's Law. The results proved to be inaccurate so two additional models were developed that incorporated cycle times directly into the WIP flow constraints. Model-II assumed that cycle times are integral multiples of the number of periods defined for a day, while Model-III allowed for fractional cycle times and was hence the most robust. Implementation consisted of five major steps including, raw data preprocessing, data input, model construction, model execution, and output of results.

Using real factory data from Texas Instrument, we conducted two sets of test. In the first set, we compared the results of our models to historical data and found measurable improvement in all instances. Model-I had the best improvement but was deemed too optimistic since it uses a single period delay, which is unrealistic. Model-II and Model-III were designed to more accurately represent the true system and provided reasonable increases in total output and reductions in shortages. The latter ranging from 7.5% to 40% monthly. In the second set of tests, the computations showed that a grid of 50 periods per day was sufficient for Model-II and Model-III to converge to the optimal solutions for all data sets. Model-III was slightly more accurate, while Model-II was seen to have a significant runtime advantage due to the sparsity of its constraint matrix.

With respect to the objective function of our models, one of the difficulties with linear penalty terms is that they can lead to unbalanced solutions. In our case, this means that they don't necessarily distribute the shortages and surpluses evenly over the device groups when insufficient capacity exists over the planning horizon. One remedy is to square the s_{id}^{TP+} and s_{id}^{TP-} variables so that shortages and surpluses are penalized at an increasing rate for each device group each day.

At the current time, Model-III is being tested by the planners at TI's Taiwan AT facility. Rather than trying to implement the period-by-period DRR results at each logpoint, they have adopted our suggestion and are using the average DRR results provided in the "summary.csv" file highlighted in Table 8.

Appendix A: Implementation of Initial Conditions

Table 13 enumerates a portion of the "WIPBegin.csv" file for device group "76-48-ZABC-Y." There are 25 logpoints in the route for this group of which the maximum cycle time is 2.12 days. Thus, it is necessary for DRRInitial.csv (see Table 14) to contain 3 days of data prior to the start of the planning horizon (July 5th in this example).

PROD_LINE	PIN	TESTER	STRIP_TEST	LPT	PLAN_CT	BEGIN_WIP
76	48	ZABC	Y	5100	0.09	0
76	48	ZABC	Y	5105	0.5	0
76	48	ZABC	Y	5110	0.72	0
76	48	ZABC	Y	5200	1.8	8977
76	48	ZABC	Y	5250	0.23	0
76	48	ZABC	Y	5300	0.17	0
76	48	ZABC	Y	5500	2.03	0
76	48	ZABC	Y	5600	0.04	0
76	48	ZABC	Y	5700	0.16	0
76	48	ZABC	Y	5720	0.01	0
76	48	ZABC	Y	5750	0.26	0
76	48	ZABC	Y	6050	0.13	0
76	48	ZABC	Y	6100	0.2	26760
76	48	ZABC	Y	6901	0.09	0
76	48	ZABC	Y	7100	1.56	0
76	48	ZABC	Y	7777	0.02	0
76	48	ZABC	Y	6110	0.35	0
76	48	ZABC	Y	6120	0.35	0
76	48	ZABC	Y	7660	0.12	0
76	48	ZABC	Y	9050	0.01	0
76	48	ZABC	Y	9060	0.54	0
76	48	ZABC	Y	9070	2.12	1759
76	48	ZABC	Y	9080	0.03	0
76	48	ZABC	Y	9085	0.05	0
76	48	ZABC	Y	9900	0.01	0

Table 13: Sample of WIPBegin.csv

PROD_LINE	PIN	TESTER	STRIP_TEST	DATES	LPT	LPT_DESC	ACTUAL_DRR
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
76	48	ZABC	Y	7/3/2016	7660	OUTLIER VERIFY	0
76	48	ZABC	Y	7/3/2016	9050	PACKING STAGE	0
76	48	ZABC	Y	7/3/2016	9060	DRY BAKE	0
76	48	ZABC	Y	7/3/2016	9070	INSPECTION	4997
76	48	ZABC	Y	7/3/2016	9080	COMBINE/LBL/PACK	10000
76	48	ZABC	Y	7/3/2016	9085	TAG VERIFY	10000
76	48	ZABC	Y	7/3/2016	9900	PACK/UTS	10000

Table 14: Sample of DRRInitial.csv

The historical data doesn't record the specific hour in which production took place, so we assume it happened at the beginning of the day, i.e., during the first period. If an operation ends after the start of the planning horizon, its actual DRR replaces $RR_{i,j,\eta,1}^{Period}$ in the period, call it η, in which it will finish. For example, in Table 14 the DRR for logpoint 9070 is 4997 and its cycle time is 2.12 days. This amount (4997 units) flows to the next operation 9080 after 2.12 days, which corresponds to the η^{th} period on July 5th, where $\eta = \lceil (0.12 \times |P|) \rceil$. For $|P|$ = 100, for example, production finishes in period η = 12 on July 5th and is transferred to the next operation. In this case, we use the DRR listed (4997) in place of $RR_{i,j,\eta,1}^{Period}$ in period η in Eq. (2c). Consequently, the solution will include the run rate in DRRInitial.csv. For the last operation, logpoint 9900, the cycle time is 0.01 days so the corresponding DRR listed in DRRInitial.csv will finish production on that date and will not affect shortage and surplus values during the planning horizon. Generally, if the cycle time for operation j_i^{End} is greater than 1 (for example, 1.2 days), then any production on July 3rd will be available on July 4th. In this case, the Actual DRR in Table 4 will replace $RR_{i,J_i^{End},d}^{Daily}$ in Eq. (2e).

Appendix B: Pseudocode for data input

Procedure_ Data_Input
Description: Determine model size and read data files
Input: WIPPlanStart.csv: device group, LPT, date, *STARTS*, and *DOQR*
WIPBegin.csv: device group, LPT, date, *CTTGT* and begin WIP
WIPActual.csv: actual DRR, WIP levels throughout the month (when these data are available)
config.txt: size of the model
input.txt: model parameter values, such as number of periods, weight parameter for shortages
Output: result.csv: daily result of the model, such as DRR of device group i, day d, LPT j
summary.csv: summary of the solution, such as CPU time, total amount of shortages, surpluses
analysis.csv: basic analysis of result.csv, such as average DRR of the month, maximum DRR, average WIP level and so on.
Begin
Process input.txt, save weight parameters α and β, capacity parameters and so on.
Allocate arrays for input data $ACT\_WIP_{i,j,0}$ (begin WIP), CT_TGT_j, Cap_Lim_{jd}, $STARTS_{id}$, $DOQR_{id}$,
Allocate arrays for model solutions $DRR_{i,d,j}$, $WIP_{i,d,j}$, s_{id}^{TP-}, s_{id}^{TP+}
Allocate arrays for actual solutions $ACT\_DRR_{i,d,j}$, $ACT\_WIP_{i,d,j}$
Read each row and convert to corresponding parameter
Process WIPBegin.csv
While (not EOF) {
    Get current row,
    row l = row_to_vector()
    find index of device group i, in set I
    find index of date d in set D
    find index of LPT j in set J
    (**comment:** Use C++ standard library std::find() method, compare elements until a match is found)
    Read $ACT\_WIP_{i,j,0}$, CT_TGT_j, Cap_Lim_{jd}
  }
Process WIPPlanStart.csv
While (not EOF) {
    Get current row,
    row l = row_to_vector()
    find index of device group i, in set I
    find index of date d in set D
    Read $STARTS_{id}$, $DOQR_{id}$
  }
Process WIPActual.csv, if any
While (not EOF) {
    Get current row,
    row l = row_to_vector()
    find index of device group i in set I
    find index of date d in set D
    find index of LPT j in set J
    Read $ACT\_DRR_{i,d,j}$, $ACT\_WIP_{i,d,j}$
  }
End

References

Allahverdi, A., Gupta, J. N. & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, 27(2), 219-239.

Bard, J. F., Gao, Z., Chacon, R. & Stuber, J. (2013). Daily scheduling of multi-pass lots at assembly and test facilities. *International Journal of Production Research*, 51(23-24), 7047-7070.

Bard, J. F., Jia, S., Chacon, R. & Stuber, J. (2015). Integrating optimization and simulation approaches for daily scheduling of assembly and test operations. *International Journal of Production Research*, 53(9), 2617-2632

Deng, Y., Bard, J. F., Chacon, G. R. & Stuber, J. (2010). Scheduling back-end operations in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 23(2), 210-220.

Disney, S. M., Naim, M. M. & Towill, D. R. (2000). Genetic algorithm optimisation of a class of inventory control systems. *International Journal of Production Economics*, 68(3), 259-278.

Feo, T. A., Venkatraman, K. & Bard, J. F. (1991). A GRASP for a difficult single machine scheduling problem. *Computers and Operations Research*, 18(9), 635-643.

Fu, M., Askin, R., Fowler, J., & Zhang, M. (2015). Stochastic optimization of product–machine qualification in a semiconductor back-end facility. *IIE Transactions on Design & manufacturing*, 47(7), 739-750.

Gao, Z., Bard, J. F., Chacon, R. & Stuber, J. (2015). An assignment-sequencing methodology for scheduling assembly and test operations with multi-pass requirements. *IIE Transactions*, 47(2), 153-172.

Gershwin, S.B. (2000). Design and operation of manufacturing systems: the control-point policy. *IIE Transactions*, 32(10), 891–906.

Hackman, S. T. & Leachman, R. C. (1989). A general framework for modeling production. *Management Science*, 35(4), 478-495.

Hopp, W.J. & Spearman, M.L. (2007). *Factory Physics*, Third Edition, McGraw-Hill/Irwin, New York.

Hung, Y. F. & Leachman, R. C. (1996). A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. *IEEE Transactions on Semiconductor Manufacturing*, 9(2), 257-269.

Leachman, R.C. (2002). Application of mathematical optimization to semiconductor production planning. In Resende, M. & Pardolos, P. (eds.), *Handbook of Applied Optimization*, 746-762, Oxford University Press, New York.

Leachman, R.C., Benson, R.F., Liu, C. & Raar, D.J., 1996. IMPReSS: An automated production-planning and delivery-quotation system at Harris Corporation—Semiconductor Sector. *Interfaces*, 26(1), 6-37.

Lin, D. & Lee, C.K.M. (2011). A review of the research methodology for the re-entrant scheduling problem. *International Journal of Production Research*, 49(8), 2221–2242.

Marek, R.P., Elkins, D.A. & Smith, D.R. (2001). Manufacturing controls: understanding the fundamentals of Kanban and CONWIP pull systems using simulation. In *Proceedings of the 33nd Conference on Winter Simulation*, 921-929. IEEE Computer Society.

Monkman, S.K., Morrice, D.J. & Bard, J.F. (2008). A production scheduling heuristic for an electronics manufacturer with sequence dependent set-up costs. *European Journal of Operational Research*, 187 (3), 1100–1114.

Montoya-Torres, J.R. (2006). A literature survey on the design approaches and operational issues of automated wafer-transport systems for wafer fabs. *Production Planning and Control*, 17(7), 648–663.

Pfund, M. E., Mason, S. J. & Fowler, J. W. (2006). Semiconductor manufacturing scheduling and dispatching. In *Handbook of Production Scheduling* (pp. 213-241). Springer US.

Gupta, A. K. & Sivakumar, A. L. (2002, December). Semiconductor manufacturing: simulation based multiobjective schedule optimization in semiconductor manufacturing. In Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes (eds.), .*Proceedings of the 34th Winter Simulation Conference: Exploring New Frontiers* (pp. 1862-1870). E. San Diego, CA.

Uzsoy, R., Lee, C.Y. & Martin-Vega, L.A. (1992). A review of production planning and scheduling models in the semiconductor industry part I: system characteristics, performance evaluation and production planning. *IIE transactions*, 24(4), 47-60.

Van Zant, P. (2000). *Microchip Fabrication: A Practical Guide to Semiconductor Processing*, 4th Edition, McGraw-Hill, New York.

Wang, L. & Prabhu, V. (2006). Parallel algorithm for setting WIP levels for multi-product CONWIP systems. *International journal of production research*, 44(21), 4681-4693.

Zhang, H., Jiang, Z. & Guo, C. (2009). Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology. *International Journal of Advanced Manufacturing Technology*, 41(1), 110-121.

Zhang, M. T., Niu, S., Deng, S., Zhang, Z., Li, Q. & Zheng, L. (2007). Hierarchical capacity planning with reconfigurable kits in global semiconductor assembly and test manufacturing. *IEEE Transactions on Automation Science and Engineering*, 4(4), 543-552.