Dynamic Scheduling of Manufacturing Systems with Setups and Random Disruptions

by

Fernando Tubilla

B.S., Mechanical Engineering, Universidad Iberoamericana, Mexico City (2003)
S.M., Mechanical Engineering, Massachusetts Institute of Technology (2007)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author ....

Department of Mechanical Engineering
May 23, 2011

Certified by ....

Stanley B. Gershwin
Senior Research Scientist, Mechanical Engineering
Thesis Supervisor

Accepted by ....

David E. Hardt
Chairman, Department Committee on Graduate Students

Dynamic Scheduling of Manufacturing Systems with Setups and Random Disruptions

by

Fernando Tubilla

Submitted to the Department of Mechanical Engineering
on May 23, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Manufacturing systems are often composed of machines that can produce a variety of items but that most undergo time-consuming (and possibly costly) setups when switching between product types. Scheduling these setups efficiently can have important economic effects on the performance of the plant and involves a tradeoff between throughput, inventory, and operating costs. In addition, the schedule must be robust to random disruptions such as failures or raw material shortages, which are common in production environments.

In this thesis, we study policies that address the setup scheduling problem dynamically, in response to current conditions in the system. A new heuristic, called the *Hedging Zone Policy* (HZP), is introduced and developed. It is a dynamic-sequence policy that always produces the current part type at its maximum production rate until a fixed base stock level is reached. Then, before switching setups, the policy might produce the current part type at its demand rate for some additional time. When selecting changeovers, the HZP implements two types of decision rules. If the difference between base stock and surplus level is small for all part types, the item with the largest weighted difference is selected. Otherwise, the policy uses a fixed priority ranking to select between items that are far from their base stock value. In order to demonstrate the benefits of our policy, we also adapt and implement several other heuristics that have been proposed in the literature for related models.

The policies are first analyzed in a purely deterministic setting. The stability of the HZP is addressed and it is shown that a poor selection of its parameters leads to a condition in which some low-priority parts are ignored, resulting in an unstable system. Using Lyapunov's direct method, we obtain an easy-to-evaluate and not-too-conservative condition that ensures production of all part types with bounded surplus. We then compare, through a series of extensive numerical experiments with three-part-type systems, the deterministic performance of the policies in both make-to-order and make-to-stock settings. We show that the HZP outperforms other policies within its class in both cases, a fact that is mainly attributed to its priority-based decisions. When compared to the approximate optimal cost of the problem, our policy performs very well in the make-to-order case, while the simplicity of its base stock structure makes it less competitive in the deterministic make-to-stock problem.

The results are then leveraged for the study of a stochastic model, where we consider the effect of random disruptions in the form of machine failures. We prove that our model converges to a fluid limit under an appropriate scaling. This fact allows us to employ our deterministic stability conditions to verify the stochastic (rate) stability of the failure-prone system. We also extend our previous numerical experiments by characterizing the performance of the policies in the stochastic setting. The results show that the HZP still outperforms other policies in the same class. Furthermore, we find that except for cases where failures occur much less or much more frequently than changeovers, the HZP outperforms a fixed-sequence policy that is designed to track a pre-determined, near-optimal deterministic schedule.

Thesis Supervisor: Stanley B. Gershwin
Title: Senior Research Scientist, Mechanical Engineering

Acknowledgments

I am deeply thankful to my advisor, Dr. Stanley Gershwin, for his support and guidance over the course of this project, and for granting me the freedom to pursue my own research ideas. Having Dr. Gershwin as an advisor was an incredibly enlightening experience, and his mentorship extended well beyond manufacturing systems engineering. I cherish his invaluable lessons about research, science, and life in general, and I look forward to our future collaborations. Thank you Stan and Fran for your friendship!

My research was significantly enriched by the insightful observations and suggestions of my thesis committee. I was very fortunate to have Professors David Gamarnik, Stephen Graves, and Sanjay Sarma in my committee, and I thank them for their time and interest in my project, as well as for their enthusiasm during our meetings. Throughout my studies and teaching assistant duties, I also had the chance to interact with many fellow researchers and students that greatly contributed to my learning. I thank my friends Chuan Shi and Prof. Marcello Colledani for our interesting conversations, and for their feedback in different aspects of my work. I also thank the students of 2.853/4 during the fall 2008 and 2009 semesters; their motivation and interest in manufacturing systems pushed me to become a better TA.

One of the great things about MIT is its interdisciplinary spirit, which allows you to reinvent yourself several times. Before devoting my energies to the exciting fields of systems science and operations research, I had the chance to participate in interesting projects in the areas of smart structures and compliant mechanical systems. I thank my master's thesis advisor, Prof. Steven Hall, as well as Prof. Yet-Ming Chiang, Prof. Martin Culpepper, and Dr. Julio Guerrero, for their guidance while working on these projects. I also thank Professors Alexander Slocum and Kamal Youcef-Toumi for offering me teaching assistant positions in their classes.

My life as a student at MIT was made more enjoyable thanks to the help and support from all my friends, labmates, and staff in the Laboratory for Manufacturing and Productivity (LMP) and the Mechanical Engineering Department. I would also like to extend my appreciation to the MIT Portugal Program, for their financial support during the spring 2008 semester, as well as to the Legatum Center, for granting me the 2010-2011 Legatum Fellowship and for guiding my entrepreneurial aspirations.

Finally, I would not have reached this point in my life without the support from my family and friends, to whom I dedicate this thesis. *Mamá* and *Papá*, I cannot thank you enough for your love and for always believing in me. *Alby*, thanks for your friendship and advice. *Mon*, thanks for your wisdom and uncountable life lessons. To all my relatives and friends in Mexico and Boston, thank you for being there for me!

Contents

List of Figures

Nomenclature

C_I Recovery or excess I costs for an initial condition $(\boldsymbol{x}_0, \sigma_0)$, page 120

C_J Recovery or excess J costs for an initial condition $(\boldsymbol{y}_0, \sigma_0)$, page 120

$D_i(t)$ Cumulative demand of type i at time t, page 38

I Long-term average inventory and backlog cost, page 118

J Long-term average surplus-deviation cost, page 115

K_{ij} Setup cost due to a changeover from i to j, page 38

N Number of part types or items, page 38

$P_i(t)$ Cumulative production of type i at time t, page 39

$R_i(t)$ Accumulated repair time with the machine set to produce type i, page 169

S Sum of setup times for sequence-independent systems (i.e., $S = S_1 + S_2 + \cdots + S_N$), page 154

S_{ij} Setup time due to a changeover from i to j, page 38

$T_i(t)$ Production allocation time for item i up to time t, page 169

$V(\boldsymbol{x})$ Lyapunov function given by $\boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{Z}^{\mathrm{U}} - \boldsymbol{x})$, page 80

$\boldsymbol{\Delta Z}$ $N \times 1$ vector of hedging points differences, page 51

MTTF Mean time to fail of the machine, measured over non-setup time, page 169

MTTR Mean time to repair, page 169

$\boldsymbol{Z}^{\mathrm{L}}$ $N \times 1$ vector of lower hedging points, page 51

Z_i^{L} Lower hedging point of type i, page 51

$\mathcal{Z}_N$ The hedging zone, given by the set of states $\boldsymbol{x} \in \partial\mathcal{X}$ such that $x_i \in [Z_i^{\mathrm{L}}, Z_i^{\mathrm{U}}]$ for $i = 1, 2, \ldots, N$, page 52

$\boldsymbol{Z}^{\mathrm{U}}$ $N \times 1$ vector of upper hedging points, page 46

Z_i^{U} Base stock level of type i for a CC policy (in the HZP, it is also called the upper hedging point), page 46

$\boldsymbol{K}$ Matrix(vector) of sequence-dependent(independent) setup costs, page 42

$\boldsymbol{S}$ Matrix(vector) of sequence-dependent(independent) setup times, page 42

$\boldsymbol{e}_j$ Unit vector in the j-th direction, page 90

$\boldsymbol{b}$ $N \times 1$ vector of unit backlog costs per time, page 118

$\boldsymbol{c}$ $N \times 1$ vector of unit surplus-deviation costs per time, page 115

$\boldsymbol{f}$ A periodic sequence of M positions $[f^1, f^2, \ldots, f^M]$ with $f^i \in \mathcal{Q}$, page 137

$\boldsymbol{\psi}$ Discrete-time map of state $(\boldsymbol{x}_n, \sigma_n)$ to $(\boldsymbol{x}_{n+1}, \sigma_{n+1})$, page 42

$\boldsymbol{h}$ $N \times 1$ vector of unit holding costs per time, page 118

$[\![\cdot]\!]$ Iverson bracket; evaluates to 1 if the condition inside is true, otherwise to 0, page 118

$\mathbf{B}$ Matrix $\mathbf{I} - \boldsymbol{\tau}\boldsymbol{d}^{\mathrm{T}}$, page 90

$\mathbf{I}$ Identity matrix, page 90

μ_i Maximum production rate of type i, page 39

$\partial\mathcal{X}$ Boundary of $\mathcal{X}$, where $x_i = Z_i^{\mathrm{U}}$ for at least one i, page 47

p_i^{c} Long-term fraction of cruising time with type i, page 122

$\boldsymbol{\phi}$ $N \times 1$ positive vector used in the definition of $V(\boldsymbol{x})$, page 80

p_i^{s} Long-term fraction of sprinting time with type i, page 122

$\mathcal{R}(\boldsymbol{x})$ In the HZP, the set of all items j with $x_j < Z_j^{\mathrm{L}}$, page 51

ρ System utilization (equal to $\sum_i \rho_i$), page 39

ρ_i Utilization of product i (equal to d_i/μ_i), page 39

$\mathcal{X}$ Set of surplus states such that $\boldsymbol{x} \leq \boldsymbol{Z}^{\mathrm{U}}$, page 47

$\sigma(t)$ Setup state variable at time t, page 40

σ_n Part type produced during the n-th run (equal to $\sigma(t_n^-)$), page 42

τ_i Minimum production time of a unit of material of type i (equal to $1/\mu_i$), page 39

θ_i Service level of item i (long-term fraction of time with $x_i > 0$), page 118

$\boldsymbol{u}(t)$ $N \times 1$ vector of production rates, page 41

$\boldsymbol{x}(t)$ $N \times 1$ surplus vector, page 40

$\boldsymbol{x}_n$ Surplus vector at the end of the n-th run (equal to $\boldsymbol{x}(t_n)$), page 42

$\boldsymbol{y}(t)$ $N \times 1$ vector of surplus deviations at time t, page 48

d_i Demand rate for type i, page 38

e Machine efficiency, given by the ratio $\mathrm{MTTF}/(\mathrm{MTTF} + \mathrm{MTTR})$, page 169

i, j, k Part type indices, page 38

n_i Long-term frequency of production runs of type i, page 125

n_i^* Ideal production frequencies, as computed by the lower bound, page 130

n_{ij} Long-term frequency of changeovers from type i to type j, page 125

t_n Instant at which the n-th run concludes, page 42

$u_i(t)$ Controllable production rate of type i at time t, page 39

$x_i(t)$ Surplus of type i at time t (equal to $P_i(t) - D_i(t)$), page 40

y_i^* Ideal surplus deviation, as computed by the lower bound, page 131

y_j^V Surplus deviation corresponding to a V-neutral run, page 98

$y_i(t)$ Surplus deviation of type i at time t, equal to $Z_i^{\mathrm{U}} - x_i(t)$, page 48

$p_i^{\mathrm{c}*}$ Ideal cruising time fractions, as computed by the lower bound, page 130

$\mathcal{Q}$ Set of all part types, page 38

Σ A deterministic system instance described in Definition 2.1.1, page 42

Σ_π A system Σ operated under some closed-loop policy π, page 79

$\hat{\Sigma}$ A stochastic system as described in Definition 5.1.1, page 171

$\hat{\Sigma}_\pi$ A system $\hat{\Sigma}$ operated under some closed-loop policy π, page 171

Chapter 1

Introduction

Production systems are often composed of machines that can produce a variety of items but that must undergo time-consuming and possibly expensive *setups* when switching between them. While these setups may sometimes be reduced through appropriate practices (see, e.g., Hopp and Spearman 2008, p. 162 and references therein), they cannot always be eliminated completely from the production process. Therefore, factory managers often face a complex decision problem that can greatly affect the efficiency of the plant. On the one hand, they would like to be able to change setups often in order to meet the demand for all items with small amounts of inventory and short lead times. On the other hand, frequent changeovers will reduce the plant's capacity and waste valuable production time. Moreover, the decision needs to be made under considerable uncertainty since random events such as machine failures or raw material shortages will likely affect any envisioned production plan.

Given the economic significance of the problem (Allahverdi et al. 2008), it is not surprising that production scheduling has attracted so much interest among researchers and practitioners alike. But, despite decades of effort, there is still a need for devising methods that perform well under conditions that are representative of actual production environments and that are also practical to implement in the shop floor. In this introductory chapter, we discuss how the present thesis aims to contribute towards such goal. After motivating the problem, we will overview the relevant literature and state of the art in scheduling research. We then describe the main objectives of the thesis and provide an outline of its contents.

1.1 The Setups Scheduling Problem

In a general sense, we can think of a schedule as a rule that relates a set of activities with the times at which those activities should take place. Such a rule must take into account constraints such as the availability of the resources that are needed for carrying out the activities, as well as the times at which those activities should begin or conclude. In addition, the schedule should have adequate (if not optimal) performance with respect to some specified metric. In a manufacturing environment—which is what this thesis focuses on—the activities to schedule typically consist of machine operations, and the resources that need to be coordinated include the machines, raw parts to feed into the process, personnel, process consumables, and storage space for the finished goods. Besides being feasible, the schedule should be such that customers are kept happy (e.g., by not making them wait too much for their products) and should also allow stakeholders to obtain the maximum value from their investments (e.g., by operating with low inventory levels and using resources efficiently).

In today's markets, customers increasingly demand more product variety and customization. This means that manufacturers must deal with a wide range of different *items* or *product types* in their production lines (e.g., one of the factories that the author worked with at the beginning of this research, whose process is described in Xie 2008, deals with more than 200 different product types!). Furthermore, even when the number of product types is small, it is not always economical to devote entire production lines or machines to each one of these items. Thus, it is often necessary to share the system's capacity among the different items, leading to setup scheduling considerations.

A *setup* or changeover is defined as any preparatory procedure that needs to be performed whenever a machine switches production between different items and before good parts of the new item are produced. Examples of setups abound in industry, including changing dies in metal stamping operations, cleaning containers in food-processing industries, adjusting raw-material fixtures, and recalibrating machines (see Allahverdi et al. 2008 for other examples). When these preparatory procedures take a significant amount of time to complete, the most appropriate way to account for them is by including setup *times* explicitly in the scheduling formulation, as opposed to simply using a surrogate cost penalty (Dobson 1987). However, there are settings where we can also associate a direct cost with each setup change. Such is the case, for example, when the first few parts of every new production run have poor quality and need to be discarded, or when expensive solutions are used for cleaning a machine before changing product types such as paint colors or food ingredients.

Any schedule formulation should strive for a balance between throughput, inventory, and setup costs: Too many setups lead to wasted production capacity and large operating costs, while infrequent setups imply long cycle times and forces a manufacturer to stock more inventory of each item to avoid costly backlogs. To make matters more complicated, this balance must be usually addressed in *real-time*, since the conditions in the factory floor are continuously changing. For example, suppose a machine fails in the middle of a production run. What should a manager do once his crew is able to repair it? Should he resume the production of the part type that was set up for production, or should he change and produce another item? Would it be more efficient to satisfy the customer's orders that accumulated during the breakdown by doing infrequent setups and thus not wasting capacity, or by doing short runs of each item so that the orders for each product type have similar lead times?

These and many other challenging questions call for a dynamic approach to the problem.

1.2 Background

The scheduling literature is vast and growing (Allahverdi et al. 2008). The purpose of this section is to provide a brief overview of the literature so that the objectives and contributions of our work can be appreciated within this context. We can classify the scheduling literature along many dimensions, including the way in which jobs or requirements are generated, the performance measure of interest, and the size or complexity of the processing plant (Graves 1981). However, for our present discussions, our main discerning criteria will be the distinction between *open-loop* and *closed-loop* formulations.

We say that a formulation is open-loop if it does not model explicitly any feedback in the schedule generation process. Thus, given some parameters and requirements describing the problem, the goal in these formulations simply consists of coming up with an efficient *plan* or *schedule*. An open-loop formulation tacitly assumes that the conditions in the factory will not change much over the planning period. However, random disruptions are a common occurrence in factories and thus managers often face what is known as the *rescheduling problem*, where they need to constantly revise the open-loop schedule due to changing conditions in the plant. This can lead to system nervousness and poor performance (Graves 1981).

Contrary to open-loop approaches, closed-loop formulations seek a *policy* that determines the schedule in real-time, based on the current state of the system. That is, reactive decisions are incorporated explicitly into the model and used to achieve better performance through the use of feedback. Thus, while an open-loop formulation outputs a plan that might state something like Produce part type i at time t, a closed-loop formulation will instead give a rule that depends on the state of the system, such as Produce the part type with the largest backlog.

In the following sections, we provide an overview of the research on these two types of formulations, with a special emphasis on the closed-loop literature.

1.2.1 Open-Loop Scheduling

Most of the early scheduling research has been open-loop, and an excellent survey was done by Graves (1981). The "traditional" problem in this area focuses on sequencing a fixed number of jobs with the objective of optimizing some metric such as the utilization level of the production resources, the time it takes to complete all jobs, or the average time jobs spend in the system. Jobs are assumed to be generated directly from customer's orders, and thus the system is assumed to be *make-to-order*. Problems in this category are of a combinatoric nature and deal with either single or multiple stations; in the latter case, jobs may follow either a fixed route (*flow shop*) or more general routes (*job shop*).

One important critique about this literature is the assumption that the number of jobs is fixed at the beginning of the scheduling exercise and that no more jobs will arrive during the period under consideration. This restriction is unrealistic in many cases (Conway and Maxwell 2003), and has contributed to the gap between theoretical research and scheduling practice (Dudek et al. 1992). However, some of the sequencing insights obtained from this research have inspired heuristic dispatching rules that have been found to be robust in dynamic settings, where jobs arrive randomly into the system (see, e.g., Vollmann et al. 2005, p. 446).

Another important stream of open-loop formulations consists of *make-to-stock* models, in which the production requirements are triggered by inventory replenishment decisions, and customers or downstream machines expect to satisfy their demand from a finished-goods buffer without delay (Hopp and Spearman 2008, p. 230). These problems involve the determination of lot sizes on each production run as well as the setup sequence, and can be cast as mixed integer linear programs (Graves 1981). Two important formulations in this area are the *Capacitated Lot Scheduling Problem* (CLSP) and the *Economic Lot Scheduling Problem* (ELSP), both of which deal with multiple items and a single stage or machine.

The CLSP is a finite-horizon, discrete time problem in which the goal consists of minimizing holding and setup costs while meeting the capacity constraints and avoiding stockouts. Some heuristics for this problem are reviewed in Maes and Wassenhove (1988), while Trigeiro et al. (1989) considers the more complex problem with setup times.

The ELSP, on the other hand, is an infinite-horizon, continuous time problem in which the goal consists of minimizing average inventory holding and setup costs (see Elmaghraby 1978 for a review of some of the early approaches; more recent works include Hsu 1983, Dobson 1987, and Moon and Silver 2002). Material in the ELSP is assumed to be continuous and the demand for each item increases at a constant rate; the solution consists of a sequence of changeovers and the lot sizes at each position in the sequence. Since the ELSP holds a tight connection with our model formulation, we will discuss it in more detail in Chapter 4.

1.2.2 Closed-Loop Scheduling

Although less extensive than for open-loop scheduling, there is a considerable literature addressing closed-loop scheduling formulations. Given that most open- and closed-loop problems are intractable, researchers generally focus on developing and studying heuristics that are sometimes motivated by optimality results and characterizations derived for simple systems. In the make-to-order setting, *polling models* have received wide attention. In their most basic form, a polling system consists of a single processor that serves different types of jobs, each arriving to a corresponding buffer, and with a setup time incurred whenever the server switches between queues. This model has a wide range of applications, including computer networks, robotics, traffic control, transportation, and manufacturing systems

(Levy and Sidi 1990).

The scheduling policy in the polling model formulation must make two decisions: which queue to visit next and for how long to serve it. The former is usually addressed through a periodic sequence, a special case of which is when each queue is visited exactly once in the sequence (this case is called a *cycle* or *rotation*). For the length of the service period determination, some of the most commonly-studied disciplines include *exhaustive* service, where the current queue is emptied before changing queues, *gated*, where only the jobs present at the beginning of the service period of the current queue are processed, or *limited*, where each queue is served until emptied or until a specified number of jobs are served (Takagi 1988). Some authors have also studied policies where the sequence of visits to the queues is generated dynamically. For example, Duenyas and van Oyen (1996) constructed a scheduling policy that makes decisions based on the rates of reward of different control actions. In a recent paper, Lan and Olsen (2006) studied an exhaustive policy for a polling model with both setup times and costs. This policy will be discussed in Chapter 2, where we adapt it to our model formulation and use it as a benchmark for comparisons (in both make-to-order and make-to-stock settings).

Closed-loop formulations have also been developed for scheduling make-to-stock systems. In this case, the model is usually referred to as the *Stochastic Economic Lot Scheduling Problem* (SELSP) (see Sox et al. 1999 for a review). The SELSP mirrors its deterministic counterpart except for the fact that the production process is discrete, and the arrival/processing/setup times are random. One of the first works in this area was by Graves (1980), who developed a dynamic-sequence policy for the case with setup costs and no setup times. Federgruen and Katalan (1996) developed cyclic base stock policies for systems with setup times and costs; these policies follow either the exhaustive or gated service disciplines, and allow for the insertion of possible idle times between service periods as a way to reduce the frequency of setups and obtain lower costs. Other cyclic policies are those of Markowitz et al. (2000), which are based on heavy-traffic approximations. An SELSP heuristic that will also form part of our benchmarking policies was proposed by Gallego (1990). This policy starts with a periodic solution to an extended version of the ELSP and implements a control strategy for adjusting the lot sizes dynamically, in order to recover from disruptions. The policy will be discussed in more detail in Chapters 2 and 4.

A great portion of the closed-loop scheduling literature has focused on systems where the most significant source of randomness comes from the job arrival and service processes. However, an important research stream from the manufacturing systems community has focused on closed-loop scheduling under disruptions due to machine failures. The seminal work in this area was by Kimemia and Gershwin (1983), in which the production management of a make-to-stock, failure-prone flexible manufacturing system was formulated as a flow control problem with negligible setup times. In this model, the machine is able to perform many operations before it fails and thus, at the time scale at which failures occur, the production process can be regarded as continuous. The authors stated the dynamic programming optimality equations and, based on their structure, characterized the form of the optimal policy for this system, which is called a *Hedging Point Policy* (HPP). According to this policy, the system should build up enough inventory until it reaches a point called the *hedging point*, which is optimal for hedging against future failures. The optimality of this strategy was latter verified rigorously for the single-machine, single-part-type case by Akella and Kumar (1986) and by Bielecki and Kumar (1988), for the discounted and average cost cases (respectively).

The single-part-type HPP has an intuitive structure. Defining the machine's *surplus* as the difference between the cumulative production and the cumulative demand (see Chapter 2), the policy starts by setting the hedging point or *base stock level* Z. Then, whenever the current surplus is below Z and the machine is up, the system produces at full capacity until the target is reached. Once the base stock level is reached, the machines keeps producing at the demand rate (assumed constant) so that the surplus stays at that level. Whenever a failure occurs, the built up inventory is used to satisfy the demand during the repair period and any excess demand is backlogged. Thus, Z serves for hedging against the risk that a disruption will prevent the system from meeting its demand for a long period, leading to large backlog costs. As Srivatsan and Dallery (1998) put it, "*The hedging point represents the desired surplus level based on tradeoffs between expected inventory and backlog costs*". Interestingly enough, Bielecki and Kumar (1988) proved that the optimal Z can sometimes be 0, showing that operating with safety stock is not always optimal.

Except for a few cases (see, e.g., Khmelnitsky et al. 2009) analytical optimality results for more complex flow-control systems have been out of reach. Srivatsan and Dallery (1998) conjectured that for systems with two or more part types, the structure of the HPP is still optimal. That is, the optimal policy partitions the surplus space into zones, each associated with a constant control, and the boundaries of these areas are such that the system's surplus is attracted towards the hedging point. The authors provided a partial characterization of the optimal hedging point policy for a system with two part types under a linear surplus cost structure. They showed that if the system is such that the (two-dimensional) hedging point is 0, the optimal policy follows the familiar $c^{-}\mu$ rule (this rule is discussed in Appendix B), with c^{-} corresponding to the unit backlog cost (in this thesis, we denote this unit cost by b). However, in the more general case where the hedging point is not zero, the boundaries separating the different control zones are conjectured to be nonlinear (at least over some portion of the surplus space) and hard to determine analytically.

The fact that, even for the two-part-type case, it not possible to obtain the optimal policy analytically shows that, in general, a heuristic approach must be adopted. Furthermore, even if we were able to obtain the optimal policy, it may still be too cumbersome to state and implement, making it impractical for applying it in factory floors. One heuristic approach consists of approximating the cost-to-go function of the optimal control problem using quadratic functions (Gershwin et al. 1985). Perkins and Srikant (1997) focused instead on obtaining the optimal policy within a specific class. Their policy, called the *Prioritized Hedging Point Policy* (PHP), generalizes some of the characteristics of the two-part-type HPP. In particular, items are first rank ordered using some fixed priority assignments. The policy then keeps the highest priority part types at their hedging points and allocates the remaining capacity to producing the next highest priority type whose current surplus level is below its hedging point. A similar generalization was also proposed by Gershwin (2000), in what he called the *Control Point Policy*. This policy controls the release of material at different points in a production line, and selects the machine's production rates based on the generalized hedging point concept as well as other local information such as upstream/downstream buffer space.

While Kimemia and Gershwin (1983) only considered in their flow-control model two events with different characteristic frequencies (namely, machine operations and failures), Gershwin (1989) later extended the concept into a *hierarchical control* framework. In this framework, the production scheduling of a factory is separated into different levels or subproblems, each one encompassing a particular time scale of interest. For each level in the hierarchy, events that occur at higher frequencies with respect to the level's time scale manifest themselves in terms of their averages rates, while variables that change at much lower frequencies can be treated as constant over the level's optimization period. The production rates obtained by scheduling the current level in the hierarchy are then passed on to the next level, where these rates become the target for a shorter-term planner to follow (Gershwin 2002). A survey of asymptotic optimality results for the hierarchical control approach can be found in (Sethi et al. 2002).

The hierarchical decomposition of events may be used for extending the flow control formulation to cases where setup times are non-negligible and changeovers occur at a frequency that differs largely from that of other events. In this case, a long-term plan would not need to schedule the detailed times at which setups changes occur; it would only need to consider their effect in an aggregate way, by setting target production rates that take into account the lost capacity due to changeover frequencies (Gershwin 2002). At a shorter time scale, another planner would then translate these target production rates into actual changeover schedules. With this hierarchical decomposition approach in mind, Perkins and Kumar (1989) studied a class of policies that track a set of target production rates in the presence of setups and under upstream inventory costs. The policies proposed in this influential paper follow a *clearing* or exhaustive discipline, always produce at full capacity the current part type, and generate the changeover sequence dynamically. The authors also derived a lower bound on the average cost of their model and proposed a policy based on their bound. This policy is described in Chapter 2.

Perkins and Kumar's lower bound was later refined by Chase and Ramadge (1992), who recognized that in some cases it may be optimal to produce at the demand rate for some time before switching setups. In a discrete-material model, this is equivalent to inserting idleness into the schedule; however, as discussed in more detail in Section 2.1.2, in a continuous-material model idleness manifests itself in the form of a reduced production rate. For this reason, Lan and Olsen (2006) preferred the term *cruising* for referring to the periods of time when the machine is producing at the demand rate. In fact, Lan and Olsen's policy (mentioned earlier in this section) is an extension of Perkins and Kumar's policy that allows cruising, which, as the authors point out, is essential in systems with setup costs and no setup times.

Another setup scheduling policy based on the hierarchical framework of Gershwin (1989) was proposed by Sharifnia et al. (1991). This policy, called the *Corridor Policy*, establishes hyperplanes in surplus space along target surplus trajectories that are determined from the long-term planner. Changeovers are triggered whenever the surplus trajectory hits one of these planes, and the authors derived conditions on the design of the corridor that guarantee that the trajectory converges into a limit cycle. (See Section 2.4.5 for more on this policy.)

1.3 Objectives of our Work

Based on the literature review of the previous section, we can identify two important streams of closed-loop scheduling formulations. One of these streams focuses on systems with discrete material and random arrivals/processing times, in both make-to-order and make-to-stock settings (i.e., polling models and SELSP formulations). The other stream treats material as continuous and focuses instead on the flow control problem in the presence of failures or other long disruptions. Of course, there is considerable overlap between the two formulations, and many of the insights and policies derived for one problem are applicable to the other problem.

While the flow control problem with no setups has received significant attention in the literature, for the case of systems with setups most of the policies and performance comparisons have been performed in the context of either polling models or the SELSP. We note that a failure-prone machine could in principle be modeled within the random processing times formulation, for example, by letting the processing time be equal to a constant with probability p (corresponding to the normal operation time) or to some random variable with probability $1 - p$ (corresponding to a random breakdown and repair). However, a random-processing-times formulation may not have the power to consider arbitrary reliability models without dropping the typical assumption of independent processing times. Thus, since most of the experiments reported in the literature assume independent, exponentially-distributed processing times (see, e.g., Markowitz et al. 2000 and Lan and Olsen 2006), the conclusions derived from these experiments might not hold or be transferable to the failure-prone model, where the system may spend a significant amount of time operating normally and then experiences a long disruption due to a random breakdown.

There is therefore a need for adapting or developing new policies for systems with setup times and random breakdowns (or other long disruptions), as well as for a systematic study of the policies' performance. This thesis aims to address these needs. In particular, we seek to leverage the knowledge derived from the flow control problem and the insights from polling-systems/SELSP formulations, in order to develop a new heuristic that is geared mainly towards unreliable systems. This policy should be intuitive and simple to implement in the shop floor so that it does not require overly complex computations, even when dealing with many part types. Furthermore, we seek to study the performance of the policy and compare it with other reasonable formulations from the literature adapted to our model. Ideally, the results should serve as a guideline so that a manufacturer can choose the most suitable method for scheduling changeovers in his or her plant.

1.4 Thesis Overview

The approach we adopt for synthesizing our scheduling policy consists of starting with a deterministic, closed-loop system. (As we will argue, studying a deterministic model as a stepping stone towards a more complex stochastic analysis allows us to build valuable insights and obtain useful results with less analytical difficulty.) We will describe in detail this deterministic model in Chapter 2, and formulate the scheduling policies that will be studied throughout the thesis. One of the main contributions of this thesis consists of the development and analysis of a new heuristic, the *Hedging Zone Policy* (HZP), which will be also introduced in Chapter 2. As mentioned, this policy is inspired by the results from both the flow-control and the closed-loop setups scheduling literature, and addresses systems with failures or other disruptions that occur much less frequently than machine operations.

One of the most basic concerns with any dynamic scheduling heuristic consists of verifying that it is stable. In Chapter 3, we will show that the Hedging Zone Policy introduces some analytical challenges when addressing its stability, and we thus develop a theory for dealing with this problem. Our stability theory relies on the use of Lyapunov functions and our end result will consist of an easy-to-evaluate condition that ensures stable behavior of the system and that is not too conservative.

We will then consider in Chapter 4 the issue of performance of the deterministic model. We will first provide the derivation of some cost bounds that have been developed in the literature and which have proved useful for designing reasonable heuristics. Through the derivation of these bounds, we will motivate some of the features of our policy (as well as the other policies considered in the thesis) and we will establish a method for selecting the parameters of the HZP. We then describe the design of a novel series of experiments on three part type systems that allow us to examine more thoroughly the behavior of the policies. These deterministic experiments set the stage for the stochastic simulations to follow.

Chapter 5 incorporates randomness into our model in the form of time-dependent failures of the machine. The developments in this chapter will rely heavily on the results and conclusions from the previous two. In particular, we prove rigorously that the deterministic stability conditions of Chapter 3 can be used to verify the stability of the stochastic model. We will then analyze the results of our stochastic experiments, which we believe constitute the first systematic performance study of the problem of scheduling failure-prone manufacturing systems with setup times under both make-to-order and make-to-stock formulations.

In Chapter 6 we discuss several extensions and lines of interesting research for future work. We discuss issues such as systems with sequence-dependent setups, systems with more than three part types, and systems with setup costs. We also describe a problem in which random disruptions are caused by delays in the supply of raw material. We propose a natural modification of our policy for dealing with this problem and discuss how this model fits into the problem of distributed scheduling of multi-stage manufacturing systems.

Finally, Chapter 7 concludes the thesis by summarizing our main results and contributions. Several of the supporting proofs have been deferred to the thesis appendices, namely, Appendix A through C. Some of these proofs are original, while others can be found in the references indicated and are provided for completeness. Additional details about the implementation of our simulation experiments and the datasets used are included in Appendix D.

Chapter 2

Model and Scheduling Policies

In this chapter, we describe real-time strategies aimed at resolving setup scheduling decisions in the shop floor. We begin by presenting in detail the notation and mathematical model of our dynamic system and motivating its assumptions. We then describe a popular class of scheduling controls and introduce the *Hedging Zone Policy*, a heuristic whose development and analysis constitutes the main contribution of this thesis. Finally, we state and compare other relevant policies that are also applicable to our problem and that will serve as a benchmark for comparisons later. Up to Chapter 4, our model and policies formulations will ignore any explicit stochastic disturbances. The idea is to first find good *closed-loop* solutions to the deterministic scheduling problem and then to study the behavior of these solutions when randomness is incorporated explicitly into the model.

2.1 Problem Statement

2.1.1 Model Description and Notation

We consider a single machine capable of producing N product types, labelled 1 through N. Let $\mathcal{Q} = \{1, 2, \ldots, N\}$ denote the set of all part types, with cardinality $|\mathcal{Q}| = N$. Changing over from product i to product j involves a fixed amount of (strictly positive) setup time and possibly some setup cost, which we denote by S_{ij} and K_{ij} (respectively) for the sequence-dependent case, and by S_j and K_j for the sequence-independent case. (Throughout, we will reserve the dummy variables i, j, and k for representing part type labels.)

The machine is perfectly reliable and it always has raw material to work on any item (these two assumptions will be relaxed when we deal with the stochastic models in Chapters 5 and 6). We ignore the possible discreteness of the parts and instead adopt a continuous-material model. Consistent with this view, the cumulative demand of part type i at time t is denoted by $D_i(t)$ and is modeled as a deterministic, constant-rate process. Thus, for all times $t \geq 0$,

$$D_i(t) = d_i t, \tag{2.1}$$

where d_i is the demand rate for product i.

The cumulative production process at time t is denoted by $P_i(t)$ and is given by

$$P_i(t) = \int_{s=0}^{t} u_i(s)\mathrm{d}s, \tag{2.2}$$

where $u_i(t)$ is the (controllable) production rate of product i at time t. Since the machine can only be producing one type of product at any given time, the production rate $u_i(t)$ is constrained to be 0 whenever the machine has a setup different than i, as well as when a changeover is taking place. On the other hand, if the machine is currently ready to produce type i, rate $u_i(t)$ can be adjusted to any value between 0 and μ_i, the maximum production rate for this product. The reciprocal of the maximum production rate μ_i will be denoted by τ_i, and we note that this quantity corresponds to the minimum time it takes to produce a unit of material of type i.

It is usual to define

$$\rho_i = \frac{d_i}{\mu_i} = d_i \tau_i,$$

which is variously called the *workload arrival rate* (Lan and Olsen 2006) or the *utilization of product i* (Markowitz and Wein 2001). We then say that a system has sufficient capacity if the *system utilization* ρ (i.e., the sum of all product's utilizations) is less than 1. This gives the Capacity Condition

$$\rho = \sum_{i=1}^{N} \rho_i < 1. \tag{2.3}$$

Since the utilization ρ constitutes the minimum fraction of time that the machine needs to be working in order to keep up with demand, (2.3) is a necessary condition for the stability of the system. However, as we will see in the next chapter, the condition is not sufficient.

Figure 2-1 illustrates the cumulative demand and production process for product i. As shown in the figure, the difference between the cumulative production and demand is denoted by $x_i(t)$ and is referred to as the product's *surplus*. That is,

$$x_i(t) = P_i(t) - D_i(t). \tag{2.4}$$

The surplus $x_i(t)$ can be positive, in which case we say that the system is carrying *inventory*, or negative, which corresponds to the case of a *backlog*. This variable serves as a measure of the system state and will be fed back into the controller for adjusting the schedule in real time. As Gershwin (2000) puts it, a major objective consists of keeping the surplus near zero

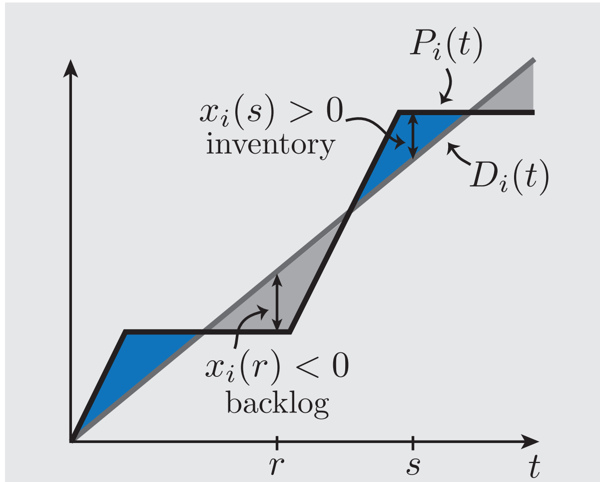


Figure 2-1: Cumulative production and demand process for type i. The surplus $x_i(t)$ is equal to the difference $P_i(t) - D_i(t)$ and it corresponds to an inventory of material when positive or to an order backlog when negative.

since, if this variable is negative, customers are not satisfied and, if it is large and positive, there will be a large inventory of finished goods.

We assume that there are no lost sales due to customer defection, so that the system can carry any amount of backlog. The column vector containing all surpluses, $\boldsymbol{x}(t) = (x_1(t), x_2(t), \ldots, x_N(t))^{\mathrm{T}}$, is referred to as the *surplus vector*.[1]

The *setup state* of the system at time t is given by the discrete quantity $\sigma(t)$. If the system is currently set up to produce type i, then $\sigma(t) = i$. If the system is currently performing a changeover from type i to type j, then $\sigma(t) = (i, j)$. To avoid ambiguities, we define $\sigma(t)$ to be a right-continuous process.

The dynamics of the system are determined by the control policy, which specifies at any time t the production rate vector $\boldsymbol{u}(t) = (u_1(t), u_2(t), \ldots, u_N(t))^{\mathrm{T}}$ and the instants at which a setup change occurs (changeover epochs) based on the current system state. We will restrict ourselves to policies where the time dependence in $\boldsymbol{u}(t)$ comes solely from $(\boldsymbol{x}(t), \sigma(t))$ (i.e.,

[1]Unless otherwise specified, all vectors are column vectors and are denoted with lowercase bold symbols.

we consider *stationary* policies). Thus, with a slight abuse of notation,

$$\boldsymbol{u}(t) = \boldsymbol{u}(\boldsymbol{x}(t), \sigma(t)).$$

Apart from this restriction, the form of an admissible policy can be quite arbitrary as long as the production rate constraint

$$u_i(t) \begin{cases} = 0 & \text{if } \sigma(t) \neq i, \\ \in [0, \mu_i] & \text{if } \sigma(t) = i, \end{cases} \tag{2.5}$$

is satisfied and a unique solution $x_i(t)$ to (2.4) exists.

While the system evolves in continuous time, we can study its behavior as a discrete-time sampled process, a viewpoint that will come very handy during our stability analysis. Under this perspective, the state of the system is sampled at the end of each *production run* where, as shown in Fig. 2-2, the n-th production run is the period of time between the start of the n-th changeover and the start of the $(n+1)$-st changeover. Letting sample time t_n correspond to the end of the n-th run, the surplus vector at this instant, $\boldsymbol{x}(t_n)$, will be denoted as $\boldsymbol{x}_n$.[2] Similarly, we denote by σ_n the part type that was produced during the n-th run (i.e., $\sigma_n = \sigma(t_n^-)$).

The sequence of surplus vector samples $\{\, \boldsymbol{x}_n \ ; \ n \geq 0 \,\}$ constitutes the trajectory of $\boldsymbol{x}_0$ in discrete time. This trajectory can be considered to be generated by a map

$$(\boldsymbol{x}_{n+1}, \sigma_{n+1}) = \boldsymbol{\psi}(\boldsymbol{x}_n, \sigma_n),$$

[2]To avoid any confusion, when referring to a specific component of the sampled surplus vector, we will use the longer notation. Thus, the surplus of part type i at the end of the n-th run will be always denoted as $x_i(t_n)$.

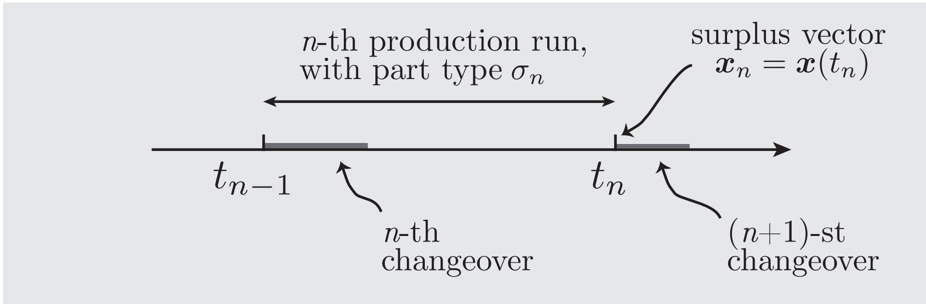


Figure 2-2: Definition of a production run and summary of the notation used to describe the discrete-time view of the process.

where $\boldsymbol{\psi}$ is a function whose specific form depends on the control policy implemented. (In most of the policies that we will consider, the setup-state variable can be inferred from $\boldsymbol{x}_n$ for $n \geq 1$, and so we will often omit σ_n in our expressions.)

The above comprise the main assumptions of our deterministic model, which are summarized in Fig. 2-3. The setup times and costs, together with the demand and maximum production rates for each item, constitute the set of system parameters that define a particular instance of the scheduling problem. These parameters, coupled with the system dynamics, form what we will refer to as a *system* Σ (defined below). Note that the behavior of a system is not completely determined until we specify a suitable scheduling policy.

Definition 2.1.1 (System) *A system* Σ *is a model instance with parameters* $(\boldsymbol{\mu}, \boldsymbol{d}, \boldsymbol{S}, \boldsymbol{K})$, *where* $\boldsymbol{\mu}$ *is an* $N \times 1$ *vector of maximum production rates,* $\boldsymbol{d}$ *is an* $N \times 1$ *vector of demand rates,* $\boldsymbol{S}$ *is an* $N \times N$ *matrix (or* $N \times 1$ *vector for the sequence-independent case) of setup times, and* $\boldsymbol{K}$ *is an* $N \times N$ *matrix (or* $N \times 1$ *vector for the sequence-independent case) of setup costs. Its dynamics satisfy* (2.1), (2.2), (2.4), *and* (2.5).

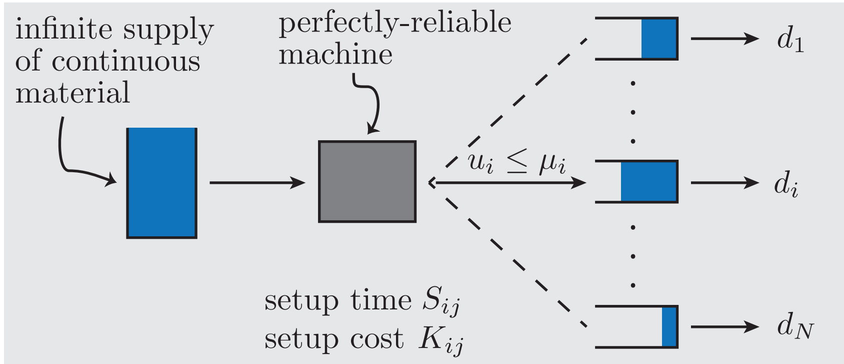


Figure 2-3: Schematic diagram of the deterministic model described in this chapter (backlog accumulation process is not depicted).

2.1.2 Model Justification

Deterministic Dynamics

Although we presented a fully deterministic model in the previous section, this thesis is concerned with scheduling setups in systems where random disruptions are the norm. The fact that we are obtaining *closed-loop* scheduling policies (as opposed to finding fixed, predetermined schedules) implies that the system will have robustness for dealing with these disruptions. This approach of relying on a deterministic model as a starting point for synthesizing a controller is common (Meyn 2008, p. 9). The idea is to first design a controller that gives the system good behavior in terms of deterministic performance measures, and then to refine it as random disturbances are incorporated into the model.

There are advantages of tackling a deterministic system as a first step towards studying stochastic models. In the first place, a deterministic model is simpler to analyze and still provides valuable intuition about system behavior. And, secondly, many results derived for a deterministic model can be shown to hold for stochastic extensions of the same system. As we will see, the results and lessons learned from the deterministic model will prove very useful for studying the more realistic stochastic scheduling models of Chapters 5 and 6.

Continuous-Material Production

While it is true that many manufacturing processes are continuous (e.g., in the chemical industries), our formulation can adequately represent discrete-parts processes as well. By choosing to model a discrete process as continuous, we are assuming that a large number of parts are produced between any other events of interest in the factory (such as setup changes or random breakdowns). Thus, over a long time scale the discreteness of the process is hidden and cumulative processes appear as continuous (e.g., if the processing times of the discrete material units form a sequence of i.i.d. random variables, this statement is justified mathematically by the functional strong law of large numbers, Chen and Yao 2001, p. 109).

Related to this assumption, a variable production rate $u_i(t)$ as the one we model here need not necessarily come from a machine with adjustable feed or production speed. Indeed, when the continuous-material assumption is a good approximation, the machine would appear (over a long-enough time scale) to be producing at a rate below μ_i if an appropriate amount of idling time occurs between production intervals (Moon et al. 1991). This concept is depicted in Fig. 2-4, where it is seen that the discrete process is increasing in a stepwise fashion, and each step takes longer than τ_i because of the idle time. The end result is that, when looked over a long time scale, the process appears to be increasing continuously at a rate lower than the maximum production rate μ_i.

Demand Process

As in the production process, the assumptions about the demand process (namely, constant rate and continuous material) model the case where a large number of discrete orders arrive between other significant events in the factory and orders accumulate at an average rate equal to the demand rate. Alternatively, in settings such as push systems, the demand $D_i(t)$ may not actually come from customer orders but rather from a long-term production goal

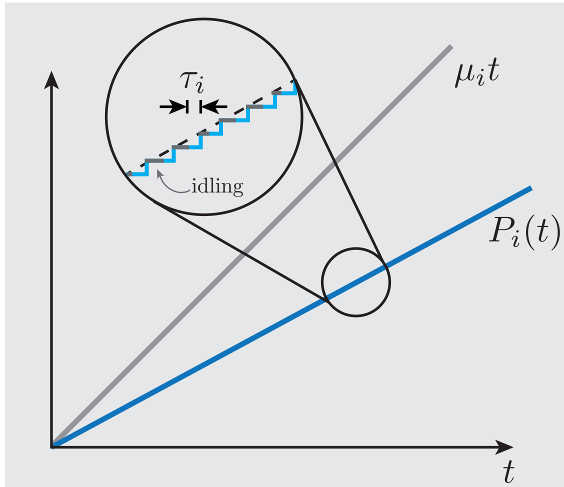


Figure 2-4: Schematic diagram of a portion of a discrete-material production process, with the zoomed-in segment showing the actual stepwise increase of $P_i(t)$. Notice how the insertion of idle time between production times makes the process appear to be linearly increasing at a rate that is lower than μ_i when looked over a long-enough time scale.

that is usually set by a planning department (see, e.g., Gershwin 2002). This production goal could be smoothed out and spread evenly over a long period, leading to a constant-rate target demand as the one we consider here.

2.2 The Clearing-Cruising (CC) Class

The dynamic equations that govern system Σ need to be complemented by the statement of the scheduling policy. While the form of this policy could be quite general, in this section we describe a popular *class* of policies that encompasses most of the heuristics that we will be considering in the thesis. We will refer to this class (which is formally defined below) as the *Clearing-Cruising (CC) Class*. The class consists of all policies in which the system produces at full capacity until it reaches a fixed base stock or surplus target level, and then it has the option of holding the surplus at this level for some time before switching over to a new part type. Thus, CC policies are *exhaustive* in the make-to-order setting (as discussed in Chapter 1) or *base stock* (Gallego 1994) in the make-to-stock setting.

Definition 2.2.1 (Clearing-Cruising (CC) Class) *A CC Class policy establishes a* fixed *surplus target or base stock level* Z_j^{U} *for each part type* $j \in \mathcal{Q}$ *and has the following two properties.*

1. *At every* t *such that* $\sigma(t) = i$*, the control* $u_i(t)$ *must satisfy*

$$u_i(t) = \begin{cases} \mu_i & \text{if } x_i(t) < Z_i^{\mathrm{U}}, \\ d_i & \text{if } x_i(t) = Z_i^{\mathrm{U}}, \\ 0 & \text{if } x_i(t) > Z_i^{\mathrm{U}}. \end{cases}$$

2. *If at some time* t *the system switches from type* i *to type* j*, then necessarily* $x_i(t) = Z_i^{\mathrm{U}}$*.*

From the first property, we can see that a CC policy will never waste production time when the surplus of the current setup is below its target level.[3] However, once this level is reached, the policy *may* decide to underutilize the system capacity for some time by producing at the current setup's demand rate before executing a new changeover. In addition, the second property states that no changeover can take place, and thus no run can conclude, while the current part type's surplus is below its target level.

Figure 2-5 shows a typical surplus-versus-time plot for a CC Class policy. As shown in the figure, we say that a system is *sprinting* whenever the machine is producing at full capacity, and we use the term *cruising*, introduced by Lan and Olsen (2006), to refer to the period of time when the machine is producing at the demand rate. Notice that a CC policy need not always cruise but, if it does, it may only do so while $x_i = Z_i^{\mathrm{U}}$.

Clearly, a CC policy never overproduces. Therefore, any surplus state with $x_i > Z_i^{\mathrm{U}}$ for some i will be transient. For this reason, without loss of generality, we can restrict our analyses

[3] The reason for the superscript U in the surplus target (or base stock) symbol Z_i^{U} will become apparent in the next section.

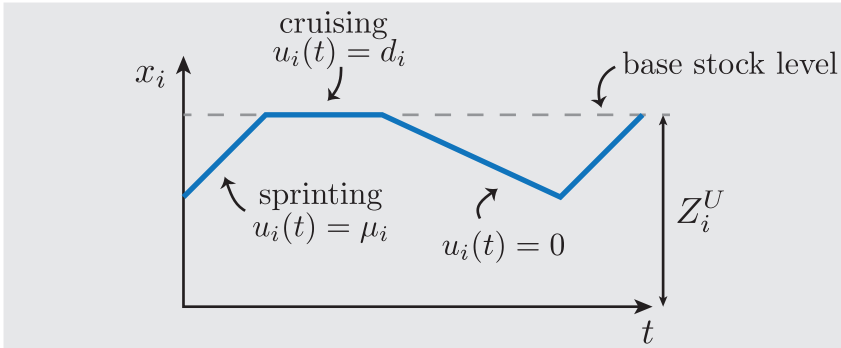


Figure 2-5: Schematic diagram of a typical surplus versus time plot for some part type i, under a CC Class policy.

of CC policies to the set of states $\mathcal{X}$, where

$$\mathcal{X} = (-\infty, Z_1^{\mathrm{U}}] \times \cdots \times (-\infty, Z_N^{\mathrm{U}}].$$

Moreover, since runs of any type i will always conclude with $x_i = Z_i^{\mathrm{U}}$, in the discrete-time view of the system all sampled states will belong to the set $\partial\mathcal{X}$, which is the boundary of $\mathcal{X}$, and is given by

$$\partial\mathcal{X} = \left\{ \boldsymbol{x} \in \mathcal{X} \;\middle|\; x_i = Z_i^{\mathrm{U}} \text{ for at least one } i \right\}.$$

The choice of base stock levels vector $\boldsymbol{Z}^{\mathrm{U}} = (Z_1^{\mathrm{U}}, Z_2^{\mathrm{U}}, \ldots, Z_N^{\mathrm{U}})^{\mathrm{T}}$ affects the amount of inventory (if any) that the system carries. If $\boldsymbol{Z}^{\mathrm{U}} = \mathbf{0}$, the system operates under a pure make-to-order setting, where all the production is triggered by existent customer orders and the (negative) surplus always corresponds to backlogged demand. This situation is equivalent to the case of a continuous-material polling model (see Section 1.2.2), shown in Fig. 2-6, where orders accumulate at the upstream buffers and the server tries to clear them as efficiently as possible.

On the other hand, if $\boldsymbol{Z}^{\mathrm{U}} > \mathbf{0}$, the system operates with inventory and, possibly at some times, backlogged demand. In this case, the average level of stocked product carried for each item should ideally balance the tradeoff between inventory costs and backlogged demand during periods where other items are being produced. It also serves as a hedging measure against random disruptions, such as machine failures or raw material shortages (thus, Z_i^{U} acts as a *hedging point* for type i).

In many of the policies that we will be considering, the actual value of the surplus $\boldsymbol{x}$ is not as important for making changeover decisions as the difference between the target vector $\boldsymbol{Z}^{\mathrm{U}}$ and $\boldsymbol{x}$. Thus it is useful to define the *surplus deviation* of each product i, denoted as $y_i(t)$, and given by

$$y_i(t) = Z_i^{\mathrm{U}} - x_i(t). \tag{2.6}$$

Since we are assuming that $\boldsymbol{x} \in \mathcal{X}$, the surplus deviations are nonnegative, and they indicate how far each surplus component is from its target level and how much is needed to produce to get back to this level (in this sense, they can be likened to the number of *production tokens* that have been released into the shop floor).

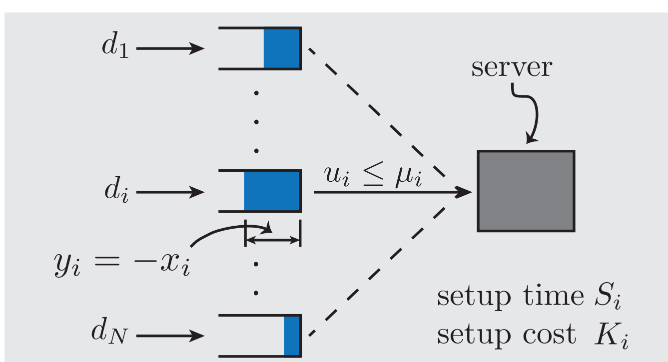


Figure 2-6: Schematic diagram of the continuous-material polling model discussed in Section 1.2.2. The buffer levels in this model follow the same dynamics as the quantities $y_i = Z_i^{\mathrm{U}} - x_i$ in our formulation.

The CC Class encompasses a variety of policies proposed in the literature, and there are several motivations behind its use. First of all, policies in the CC Class are simple to state and implement. For example, a CC policy implies that the operator would have to exhaust all outstanding orders for the current setup before considering a changeover, a rule that is easy to follow in the factory floor and that not requires constant measurement of all surplus levels. In addition, the analysis of CC policies is relatively simple due to the fact that, as mentioned, all production runs conclude at a state $\boldsymbol{x} \in \partial\mathcal{X}$ in which the item produced is at its target value. Thus, in the sampled surplus vector $\boldsymbol{x}_n$ there will always be exactly one component i with $x_i(t_n) = Z_i^{\mathrm{U}}$ for $n \geq 1$.

As discussed in Section 1.2.2, for many problems the optimal policy is characterized by a set of regions in which the control is constant and is such that it draws the state into a boundary. This behavior is mimicked in the CC Class, where the system applies a constant control (i.e., $u_i = \mu_i$) with current type i until the surplus reaches the boundary plane $x_i = Z_i^{\mathrm{U}}$, and then it possibly stays on the boundary for some time while cruising. (We will further justify these characteristics in Chapter 4).

Finally, one key property about CC policies is that when the surplus deviations vector $\boldsymbol{y}(t) = (y_1(t), y_2(t), \ldots, y_N(t))^{\mathrm{T}}$ is large (i.e., $\boldsymbol{x}(t)$ is far from $\boldsymbol{Z}^{\mathrm{U}}$), production runs tend to be longer and changeovers are performed less frequently. This in turn means that the system has more capacity and is able to clear the deviations faster (although at the expense of larger surplus swings). As $\boldsymbol{y}(t)$ gets smaller, less capacity is needed and setups are performed more frequently. Therefore, the balance between setup frequencies and system capacity is self-correcting, a feature that improves the stability of CC policies.

2.3 The Hedging Zone Policy (HZP)

The Hedging Zone Policy (HZP) is a prioritized and tunable policy that belongs to the CC Class. The policy splits the domain of $\boldsymbol{x}$ into two regions and applies on each region a different rule for selecting changeovers. On the first region, characterized by long production runs and large surplus deviations, the policy implements a prioritization-based rule, while on the second region, characterized by short and frequent runs, the policy either cruises while it waits for more demand to build up or selects the part type that is furthest behind in an appropriate sense.

We begin this section by describing the HZP in its simplest form with cruising, and we then progressively extend it to more general cases.

2.3.1 Unique Priorities

Consider the case in which each part type has been assigned a unique and fixed priority; these priorities induce a strict total order on the set of part types. That is, let $\mathsf{P}(\cdot) : \{1, 2, \ldots, N\} \mapsto \mathbb{R}$ be an injective function such that, for any two distinct part types i and j, if type i has a greater priority than type j, then $\mathsf{P}(i) > \mathsf{P}(j)$. We say then that part type i has priority $\mathsf{P}(i)$. (The term *priority* here should be solely interpreted as a fixed rank ordering. The changeover decision will be influenced by this ordering but, as we will see, it is also a function of the current surplus level.)

The HZP starts by defining two parameters for each part type. An *upper hedging point* Z_i^{U} that, as in any CC policy, plays the role of the base stock level, and a *lower hedging point* Z_i^{L}, which is used for triggering new setup changes. The vectors of upper and lower hedging points are denoted by $\boldsymbol{Z}^{\mathrm{U}}$ and $\boldsymbol{Z}^{\mathrm{L}}$ (respectively), while $\boldsymbol{\Delta Z} = \boldsymbol{Z}^{\mathrm{U}} - \boldsymbol{Z}^{\mathrm{L}}$ denotes the vector of *hedging points differences*.[4]

According to the HZP, whenever it is time to make a changeover decision, *the policy looks at all part types whose surplus lies below their lower hedging point*, and selects the one with the highest priority. If no part type is below its lower point, then the system cruises. To aid in the specification of the policy, we define the *ready set* $\mathcal{R}(\boldsymbol{x})$ as the set of all items $j \in \mathcal{Q}$ such that $x_j < Z_j^{\mathrm{L}}$. Note that this set can also be written as

$$\mathcal{R}(\boldsymbol{x}) = \left\{ j \in \mathcal{Q} \;\middle|\; \frac{Z_j^{\mathrm{U}} - x_j}{Z_j^{\mathrm{U}} - Z_j^{\mathrm{L}}} > 1 \right\},$$

which is the form that we will adopt throughout for convenience. The steps of the policy are formally stated in Policy 2.1 (see box below).

Policy 2.1: Hedging Zone Policy with Unique Priorities and Cruising

Let i represent the current setup, suppose $\boldsymbol{x}_0 \leq \boldsymbol{Z}^{\mathrm{U}}$, and define

$$\mathcal{R}(\boldsymbol{x}) = \left\{ j \in \mathcal{Q} \;\middle|\; \frac{Z_j^{\mathrm{U}} - x_j}{Z_j^{\mathrm{U}} - Z_j^{\mathrm{L}}} > 1 \right\}.$$

Then, follow these steps:

1. *Sprint with type i until $x_i = Z_i^{\mathrm{U}}$.*
2. *Cruise with type i until $\mathcal{R}(\boldsymbol{x})$ is nonempty.*
3. *Change over to the highest priority part type j^* belonging to $\mathcal{R}(\boldsymbol{x})$. That is,*
$$j^* = \operatorname*{argmax}_{j \in \mathcal{R}(\boldsymbol{x})} \{ \mathsf{P}(j) \}.$$
4. *Set $i \leftarrow j^*$ and go to Step 1.*

[4]Just as in a typical heating system there are two temperature set points that prevent the unit for turning on and off very frequently, the two hedging points provide hysteresis in the HZP by restricting the frequency of changeovers.

Analysis and Examples

For the current surplus state $\boldsymbol{x}$, the ready set $\mathcal{R}(\boldsymbol{x})$ will contain all part types whose surplus level satisfies $x_j < Z_j^{\mathrm{L}}$ and that, according to the policy, are ready for a new production run. Now, during the first step of the policy, all capacity is devoted to producing the current setup (i.e., sprinting) and, consistent with the CC Class specification, no other decision takes place until the type's surplus reaches its corresponding target value. If upon finishing Step 1 there is at least one part type in the ready set, we proceed immediately to Step 3 and change setups into the highest priority member of the ready set (since all priorities are different, this member is unique). Finally, the current-setup variable i is updated and we repeat the procedure.

When we reach Step 2 of the policy, it is possible that the ready set is empty (i.e., all surplus levels are above their lower hedging point). In this case, we say that the system is in the *hedging zone* $\mathcal{Z}_N$, which is the set of states in which all surplus components are above their lower hedging points. Mathematically,

$$\mathcal{Z}_N = \left\{ \boldsymbol{x} \in \partial\mathcal{X} \;\middle|\; x_i \in [Z_i^{\mathrm{L}}, Z_i^{\mathrm{U}}], \quad i = 1, 2, \ldots, N \right\}. \tag{2.7}$$

When the system is in the hedging zone, Step 2 states that the part type currently set up should be produced at its demand rate, until some part type gets below its lower hedging point and the system exits $\mathcal{Z}_N$ (making $\mathcal{R}(\boldsymbol{x})$ nonempty). Since the demand keeps arriving for all part types, this will occur eventually, and thus the system will only cruise for a finite amount of time.

We can see the typical behavior of the policy in a plot of *cumulative production over time*,

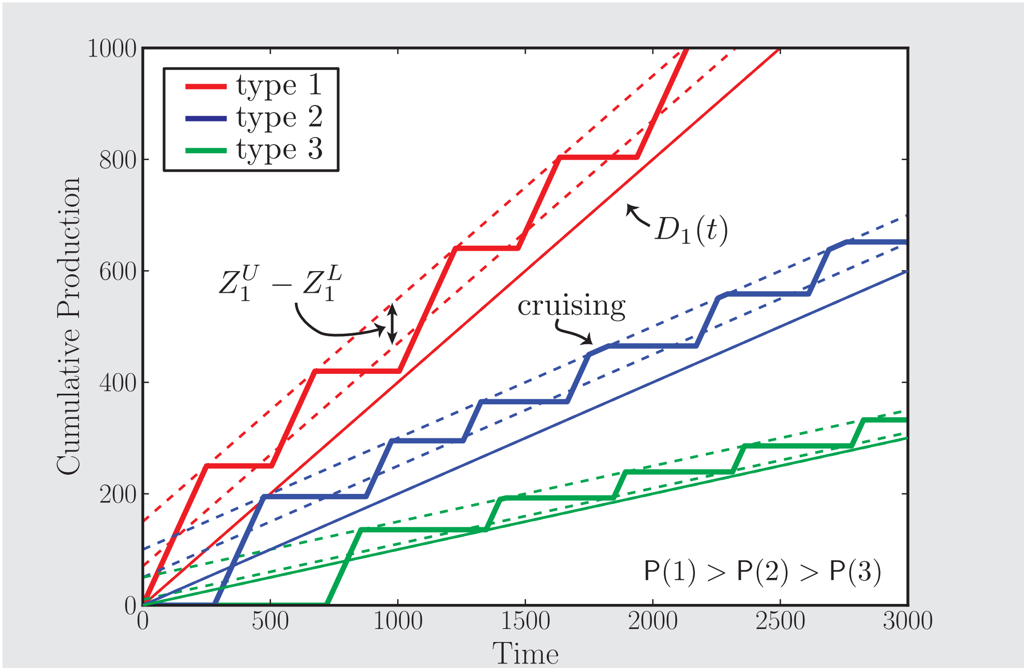


Figure 2-7: Plot of the cumulative production versus time for a three-part-type system operated under the HZP.

as shown in Fig. 2-7 for a system with the following parameters:

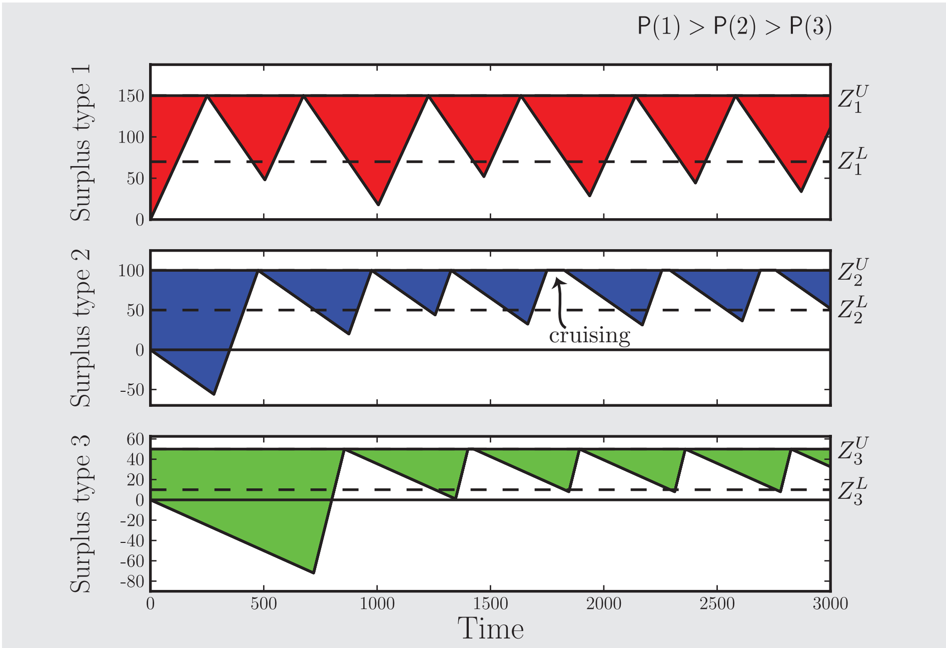
$$\boldsymbol{d} = \begin{pmatrix} 0.4 \\ 0.2 \\ 0.1 \end{pmatrix} \quad \boldsymbol{\mu} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \boldsymbol{Z}^{\mathrm{L}} = \begin{pmatrix} 70 \\ 50 \\ 10 \end{pmatrix} \quad \boldsymbol{Z}^{\mathrm{U}} = \begin{pmatrix} 150 \\ 100 \\ 50 \end{pmatrix} \quad \boldsymbol{S} = \begin{pmatrix} 0 & 30 & 45 \\ 30 & 0 & 20 \\ 45 & 20 & 0 \end{pmatrix}.$$

Notice how for each part type i, the upper hedging point establishes a target line that is parallel to the cumulative demand line and shifted up by Z_i^{U}. During the sprinting portion of the run, the cumulative production approaches this target line as fast as possible and, for runs that conclude with some cruising time, the cumulative production will then move along the target line during that period. It can also be seen that the two hedging points establish an interval for each item (the hedging zone), so that a given part type is only selected for a new run if every item with a higher priority is within this interval at the time the changeover decision is made.

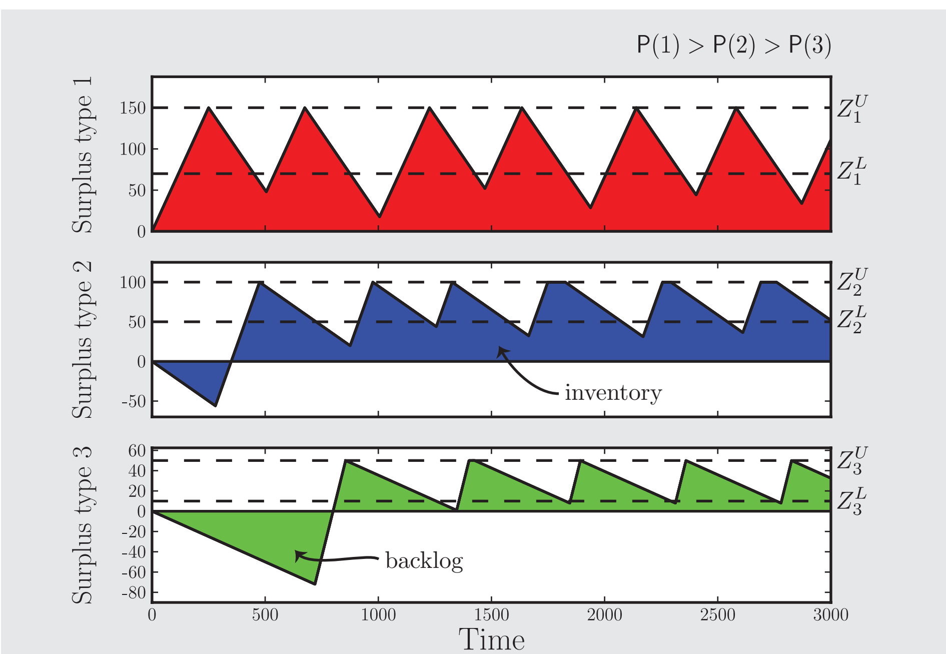
We can also illustrate the policy's behavior in terms of a *surplus-versus-time* plot. Figure 2-8a shows the trajectories of $x_i(t)$ over time for the same system parameters, with the area between $x_i(t)$ and Z_i^{U} shaded. This area is related to the work-in-process costs in the system (assuming that all the production orders given by $y_i(t)$ are immediately released into the shop floor) and should be minimized in the make-to-order case. Figure 2-8b shows the same trajectories, but now we have shaded the area between $x_i(t)$ and the zero-surplus line, which is related to the inventory and backlog costs in the system and should be minimized in the make-to-stock case. Notice that, in this example, after time $t = 800$ the system is operating with pure inventory for all part types.

If we were to change the value of $\boldsymbol{Z}^{\mathrm{U}}$ in the system while holding $\boldsymbol{\Delta Z}$ constant, the long-term surplus fluctuations shown in the previous plots would remain unchanged. This is true because, in the HZP, the dynamics of the surplus deviations $y_i(t)$ do not depend on the actual value of $\boldsymbol{Z}^{\mathrm{U}}$ but only on the vector $\boldsymbol{\Delta Z}$. In fact, as discussed in more detail in later chapters, a key fact about the analysis of the HZP is that we can separate the parameter selection process of the policy into two parts: a selection of $\boldsymbol{\Delta Z}$ based on the desired dynamics for the surplus deviations, and a selection of $\boldsymbol{Z}^{\mathrm{U}}$ based on the tradeoff between inventory and backlog costs.

Yet another way of studying the behavior of the system is in terms of surplus-space plots, where each of the coordinate axes corresponds to one of the components of the surplus vector and the surplus trajectory evolves in an N-dimensional space. Such a plot is shown in Fig. 2-9 for the same three-part-type system. The first thing to notice is the set of three colored planes of the form $x_i = Z_i^{\mathrm{U}}$, which together comprise $\partial\mathcal{X}$ (for orientation, the corner point near the center of the figure corresponds to the point $\boldsymbol{x} = \boldsymbol{Z}^{\mathrm{U}}$). The colored areas on each of these planes represent *setup zones*, which are the set of states in which a certain changeover is triggered (e.g., the blue setup zone indicates all states that, when reached,



(a) The shaded area (enclosed by $x_i(t)$ and Z_i^{U}) is related to the work-in-process costs.



(b) The shaded area (enclosed by $x_i(t)$ and the zero-surplus line) is related to the inventory and backlog costs.

Figure 2-8: Surplus versus time plots for the same three-part-type system of Fig. 2-7 trigger a changeover into part type 2). The set of states where the system cruises, the hedging zone, is indicated in yellow. The planes in the figure are translucent, so that the surplus trajectory, which lies *inside* them, can be seen.

Looking at the surplus trajectory shown in Fig. 2-9, consider the segment labelled $\mathrm{A} - \mathrm{A}'$. On this segment, part type 1 is being produced at full capacity and the production rate vector is given by

$$\boldsymbol{u}(t) = (\, \mu_1, 0, 0 \,)^{\mathrm{T}} .$$

By taking the time derivative of (2.4), we see that during this time the surplus vector is changing at rate

$$\dot{\boldsymbol{x}}(t) = (\, \mu_1 - d_1, -d_2, -d_3 \,)^{\mathrm{T}} ,$$

so that the surplus trajectory moves in the positive direction of the x_1-axis and in the negative direction for the other two. Once the trajectory hits the blue zone in the plane $x_1 = Z_1^{\mathrm{U}}$, a decision to change into type 2 is made. The trajectory will then move at rate $(\, -d_1, -d_2, -d_3 \,)^{\mathrm{T}}$ while the changeover is taking place, and then it will head towards plane $x_2 = Z_2^{\mathrm{U}}$.

In this way, the trajectory of $\boldsymbol{x}(t)$ in surplus space bounces on planes of the form $x_i = Z_i^{\mathrm{U}}$, and the discrete-time trajectory $\{\, \boldsymbol{x}_n \; ; \; n \geq 1 \,\}$ will consist of points that lie on those planes (i.e., points in $\partial\mathcal{X}$). Furthermore, whenever the trajectory hits a plane on the yellow area (corresponding to the hedging zone), the surplus vector will stay on that plane for some time until it exits the hedging zone. Note also that the setup zones are essentially what differentiate policies within the CC Class, since they establish the rule for selecting changeovers (and cruising times) as a function of the current setup and the surplus level. At the end of this chapter, we will compare the HZP setup zones with those of other CC policies considered in this thesis.

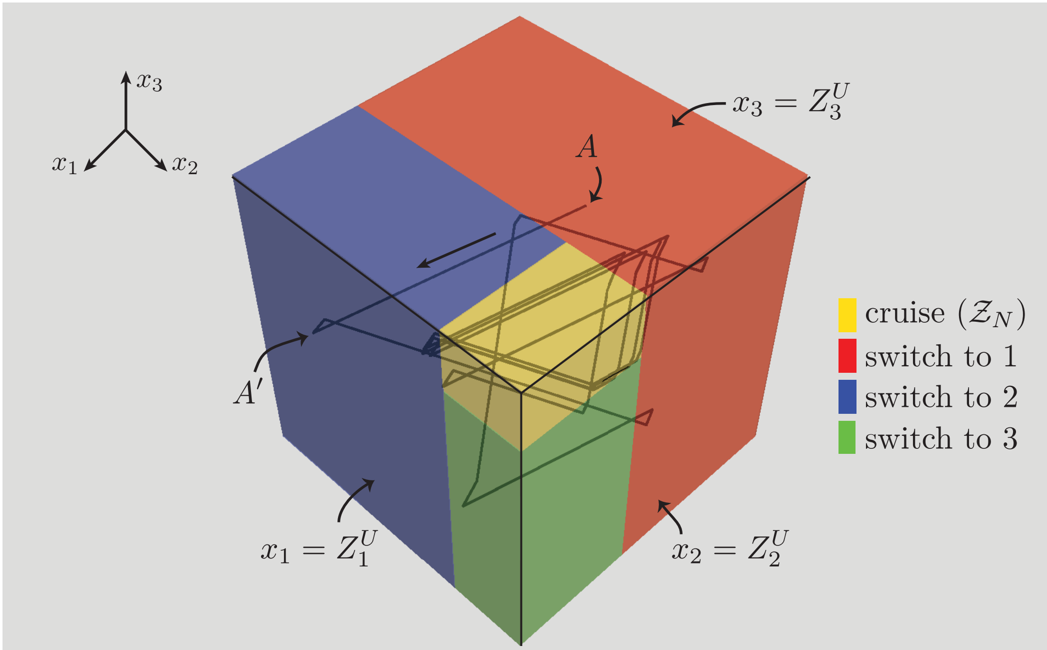


Figure 2-9: Plot of the surplus trajectory for the three-part-type system. Each axis corresponds to the surplus of one part type. The colored planes are of the form $x_i = Z_i^{\mathrm{U}}$ (i.e., set $\partial\mathcal{X}$) and indicate the setup zones, while the surplus trajectory lies *inside* these planes.

2.3.2 General Priorities

It is possible that, for a given problem, some part types have the same priority (e.g., if, as described in Section 4.4.2, the $c\mu$ rule were used for prioritizing parts, this would occur whenever $c_i\mu_i = c_j\mu_j$ for some $i \neq j$). If at the time of a changeover decision two or more part types with the same priority are in the ready set, the policy will need a way to choose one of them. While these ties could be resolved arbitrarily (e.g., randomly choose any of the parts), we will see from the numerical experiments of Chapters 4 and 5 that the actual rule employed can have an important effect on the performance of the system. For this reason, in this section we generalize the HZP for the case in which the prioritization function $\mathsf{P}(\cdot)$ is not necessarily an injection, by explicitly defining a rule that resolves ties. The modified policy is stated as Policy 2.2 (see box).

Policy 2.2: Hedging Zone Policy with General Priorities and Cruising

Let i represent the current setup, suppose $\boldsymbol{x}_0 \leq \boldsymbol{Z}^{\mathrm{U}}$, and define

$$\mathcal{R}(\boldsymbol{x}) = \left\{ j \in \mathcal{Q} \;\middle|\; \frac{Z_j^{\mathrm{U}} - x_j}{Z_j^{\mathrm{U}} - Z_j^{\mathrm{L}}} > 1 \right\}.$$

Then, follow these steps:

1. *Sprint with type i until $x_i = Z_i^{\mathrm{U}}$.*

2. *Cruise with type i until $\mathcal{R}(\boldsymbol{x})$ is nonempty.*

3. *Form the set $\mathcal{R}^*$ containing the highest priority items in $\mathcal{R}(\boldsymbol{x})$. That is,*

$$\mathcal{R}^* = \{\, j \in \mathcal{R}(\boldsymbol{x}) \;|\; \mathsf{P}(j) \geq \mathsf{P}(k),\; \forall\, k \in \mathcal{R}(\boldsymbol{x}) \,\}.$$

4. *Change over to j^* such that*

$$j^* = \operatorname*{argmax}_{j \in \mathcal{R}^*} \left\{ \frac{Z_j^{\mathrm{U}} - x_j}{Z_j^{\mathrm{U}} - Z_j^{\mathrm{L}}} \right\}.$$

5. *Set $i = j^*$ and go to Step 1.*

Analysis

Steps 1 and 2 remain the same as in the unique-priorities case. In Step 3 we form the set $\mathcal{R}^*$, which is a subset of the ready set whose elements have a higher or equal priority than any other member in $\mathcal{R}(\boldsymbol{x})$. Since the priorities are not necessarily unique, the cardinality of $\mathcal{R}^*$ may be greater than 1, and for this reason in Step 4 we select the element with the *largest weighted surplus deviation* $(Z_i^{\mathrm{U}} - x_i)/\Delta Z_i$.

As a side note, we point out that in Step 4 it is still possible for j^* to be non-unique for some values of the surplus vector (e.g., if $\boldsymbol{x} = \boldsymbol{Z}^{\mathrm{U}}$). However, since these cases are rare, we ignore them in the specification of the policy. (A simple way to resolve them is by selecting the part type with the smallest index that was not produced in the current run.)

2.3.3 Non-Cruising Version

When the surplus vector reaches the hedging zone $\mathcal{Z}_N$, we have seen that the HZP mandates that the system cruises (i.e., that the system produces the current setup at its demand rate). Cruising reduces the frequency of changeovers and increases the length of production runs. As we will discuss in more detail in Chapter 4, this can lead to lower costs, especially in systems with large setup costs and moderate ρ. However, for many systems, cruising is not beneficial and lower costs can be achieved by utilizing capacity fully and changing setups even when inside the hedging zone (in fact, our experimental results of Chapters 4 and 5 focus solely on this type of systems). This is why we also consider a non-cruising version of the HZP, stated as Policy 2.3.

Analysis

The non-cruising version of the HZP differs from the cruising version (Policy 2.2) only when the surplus vector is inside the hedging zone, which corresponds to the case where the ready set $\mathcal{R}(\boldsymbol{x})$ is empty in Step 2. When this occurs, rather than keeping the current setup at its upper hedging point, the policy populates $\mathcal{R}^*$ with *all* the part types; it then selects the next changeover based on the largest weighted surplus deviation. Thus, inside the hedging zone, the non-cruising HZP implements a clear-the-largest-weighted-deviation rule.

It is important to note that, with an appropriate selection of the hedging zone, it may be possible to avoid cruising altogether without the need to use a different version of the HZP. However, in order to do so, we might need to significantly reduce the size of $\boldsymbol{\Delta Z}$ so that the system never reaches $\mathcal{Z}_N$. As we will see in Chapter 3, the HZP becomes unstable when the hedging zone is small. Therefore, we have found that for systems in which cruising does not

Policy 2.3: Hedging Zone Policy with General Priorities and No Cruising

Let i represent the current setup, suppose $\boldsymbol{x}_0 \leq \boldsymbol{Z}^{\mathrm{U}}$, and define

$$\mathcal{R}(\boldsymbol{x}) = \left\{ j \in \mathcal{Q} \;\middle|\; \frac{Z_j^{\mathrm{U}} - x_j}{Z_j^{\mathrm{U}} - Z_j^{\mathrm{L}}} > 1 \right\}.$$

Then, follow these steps:

1. *Sprint with type i until $x_i = Z_i^{\mathrm{U}}$.*
2. *If $\mathcal{R}(\boldsymbol{x})$ is empty, then let $\mathcal{R}^* = \mathcal{Q}$. Otherwise, let*
$$\mathcal{R}^* = \{\, j \in \mathcal{R}(\boldsymbol{x}) \;\mid \mathsf{P}(j) \geq \mathsf{P}(k),\; \forall\, k \in \mathcal{R}(\boldsymbol{x}) \,\}.$$
3. *Change over to j^* such that*
$$j^* = \underset{j \in \mathcal{R}^*}{\operatorname{argmax}} \left\{ \frac{Z_j^{\mathrm{U}} - x_j}{Z_j^{\mathrm{U}} - Z_j^{\mathrm{L}}} \right\}.$$
4. *Set $i = j^*$ and go to Step 1.*

lead to better performance, it is better to use the non-cruising HZP described in this section as opposed to the non-cruising HZP with a small $\boldsymbol{\Delta Z}$. (To select between the cruising or non-cruising version of the policy for a given problem, we will use the criterion proposed by Lan and Olsen 2006 and discussed in Section 4.4.)

2.3.4 General Form

We have developed the Hedging Zone Policy incrementally. Starting with the simplest case, we generalized it to deal with different prioritization schemes and distinguished between the cruising and non-cruising versions. Now, in Policy 2.4, we condense both versions into a complete policy that captures all the qualities of the HZP. An advantage of stating the policy in this form is that, through the variation of a single parameter, a gradual transition between the cruising and the non-cruising versions of the policy can be obtained. This feature is attractive for stochastic systems, where it may only be desirable to implement cruising during certain cases in which the system departs from its typical behavior (as motivated by the results of Lan and Olsen 2006).

Policy 2.4: Hedging Zone Policy in General Form

Let i represent the current setup and suppose $\boldsymbol{x}_0 \leq \boldsymbol{Z}^{\mathrm{U}}$. Define the ready set as

$$\mathcal{R}(\boldsymbol{x}, r) = \left\{ j \in \mathcal{Q} \;\middle|\; \frac{Z_j^{\mathrm{U}} - x_j}{Z_j^{\mathrm{U}} - Z_j^{\mathrm{L}}} > r \right\} \quad \text{for any } r > 0,$$

and fix a parameter r^* in the range $[0, 1]$.
Then, follow these steps:

1. *Sprint with type i until $x_i = Z_i^{\mathrm{U}}$.*

2. *Cruise with type i until $\mathcal{R}(\boldsymbol{x}, r^*)$ is nonempty.*

3. *If $\mathcal{R}(\boldsymbol{x}, 1)$ is nonempty, then*

$$\mathcal{R}^* = \{\, j \in \mathcal{R}(\boldsymbol{x}, 1) \;|\; \mathsf{P}(j) \geq \mathsf{P}(k),\ \forall\, k \in \mathcal{R}(\boldsymbol{x}, 1) \,\},$$

otherwise $\mathcal{R}^* = \mathcal{Q}$.

4. *Change over to type j^*, where*

$$j^* = \underset{j \in \mathcal{R}^*}{\operatorname{argmax}} \left\{ \frac{Z_j^{\mathrm{U}} - x_j}{Z_j^{\mathrm{U}} - Z_j^{\mathrm{L}}} \right\}. \tag{2.8}$$

5. *Set $i = j^*$ and go to Step 1.*

Analysis

The parameter r^* is a *cruising parameter* that must be selected beforehand. Consider first the case where $r^* = 1$, which corresponds to the cruising version of the policy. In this case $\mathcal{R}(\boldsymbol{x}, r^*) = \mathcal{R}(\boldsymbol{x}, 1)$ is exactly the same as $\mathcal{R}(\boldsymbol{x})$ in Policy 2.2 and thus we recover the cruising version of the HZP. For the case $r^* = 0$, which corresponds to the non-cruising HZP, we see that $\mathcal{R}(\boldsymbol{x}, 0)$ is always non-empty and thus the policy never cruises in Step 2.

The new scenario arises when r^* is strictly greater than 0 and less than 1. In this case, the system experiences three types of behavior: (1) it cruises whenever $\boldsymbol{x}$ lies inside a *reduced* hedging zone with sides lengths $r^* \Delta Z_j$, for $j \in \mathcal{Q}$; (2) it follows a clear-the-largest-weighted-deviation rule whenever $\boldsymbol{x}$ is between the *reduced* hedging zone and the original hedging zone; and (3) it chooses the highest-priority part that lies outside of the original hedging zone, if there is such a part at the changeover decision epoch.

We see then that the r^* parameter can be used to obtain a middle ground between the cruising and the non-cruising versions of the policy: a policy that does not implement cruising on a regular basis, but that may do so if the surplus deviations get sufficiently close to $\mathbf{0}$.

2.4 Other Policies

As mentioned in Chapter 1, several policies have been proposed in the literature that are applicable to the scheduling of system Σ. Since we will be comparing the performance of some of these policies in later chapters, in this section we present their statements in the context of our model formulation and notation. The first three policies belong to the CC Class and, while they were originally conceived with the polling server model of Fig. 2-6 in mind, they can easily be adapted to our formulation by stating them in terms of surplus deviations $y_i = Z_i^{\mathrm{U}} - x_i$. The last two policies do not belong to the CC Class.

2.4.1 Clear-the-Largest-Buffer Policy (CLB)

This is the most straightforward policy and has been studied by several researchers (see, e.g., Liu et al. 1992 and Perkins and Kumar 1989). For a make-to-order system with backlog accumulating on upstream buffers (i.e., a polling system), the policy exhausts at full capacity all orders in the current buffer and then switches to the buffer with the largest amount of material. Policy 2.5 states the steps in terms of surplus deviations.

Policy 2.5: Clear-the-Largest-Buffer Policy

Let i represent the current setup and suppose $\boldsymbol{x}_0 \leq \boldsymbol{Z}^{\mathrm{U}}$. Then, follow these steps:

1. *Sprint with type i until $x_i = Z_i^{\mathrm{U}}$.*
2. *Changeover to part type j^* such that*

$$j^* = \operatorname*{argmax}_j \left\{ Z_j^{\mathrm{U}} - x_j \right\}.$$

3. *Set $i = j^*$ and go to Step 1.*

The CLB policy belongs to the more general Clear-a-Fraction (CAF) class of policies, in which the next changeover is selected based on the part types whose buffer level exceeds a fraction of the sum of all buffer levels (there may be ties that need to be resolved). These policies do not implement cruising in their original definitions, but they do belong to the CC Class. Similar policies have also been defined in terms of the work required to clear a buffer, which in our case would correspond to $\tau_i(Z_i^{\mathrm{U}} - x_i)$ for buffer i (Lou et al. 1991).

2.4.2 Perkins-Kumar Policy (PKP)

The Perkins-Kumar Policy (Perkins and Kumar 1989) is a non-cruising policy that relies on the fact that, for systems with sequence independent setups, there exists a simple-to-evaluate lower bound on the long-term average surplus-deviation cost of the system. This lower bound gives rise to a set of ideal surplus deviations y_j^* for each part type j, which are used for selecting changeovers based on the actual deviations $y_j(t)$. While the derivation of the cost bound will be discussed in Chapter 4, Policy 2.6 states the steps involved in PKP.

Policy 2.6: Perkins-Kumar Policy

Let i represent the current setup, $\boldsymbol{x}_0 \leq \boldsymbol{Z}^{\mathrm{U}}$, and y_j^* be given by (4.23) of Chapter 4. Then, follow these steps:

1. *Sprint with type i until $x_i = Z_i^{\mathrm{U}}$.*
2. *Change over to part type j^* such that*

$$j^* = \underset{j,\ j \neq i}{\operatorname{argmax}} \left\{ \frac{Z_j^{\mathrm{U}} - x_j + S_j d_j}{y_j^*} \right\}.$$

3. *Set $i = j^*$ and go to Step 1.*

Notice that for selecting the next changeover among all possible part types j, the setup time S_j is taken into account together with the parameter y_j^*. (The constraint $j \neq i$ in the maximization of Step 2 does not form part of the policy's original statement; however, this constraint is necessary to ensure that the resulting j^* is not equal to the current setup i, a situation that can sometimes occur when $\boldsymbol{x}$ is close to $\boldsymbol{Z}^{\mathrm{U}}$.) This policy has been shown experimentally to work very well in many cases, particularly when the cost bound is tight.

2.4.3 Lan-Olsen Policy (LOP)

As mentioned in Chapter 1, based on the work of Perkins and Kumar (1989), Chase and Ramadge (1992) identified that cruising in some systems leads to a lower cost. Based on this fact, and on an improved cost bound, Lan and Olsen (2006) developed a policy (called Heuristic 1 in their publication) that refines the PKP and allows for cruising. The steps are outlined in Policy 2.7.

Policy 2.7: Lan and Olsen (LOP) Policy

> Let i represent the current setup and suppose $\boldsymbol{x}_0 \leq \boldsymbol{Z}^{\mathrm{U}}$. Let y_j^* be obtained as explained in Section 4.4.1 and define the ready set
>
> $$\mathcal{R}(\boldsymbol{x}, r) = \left\{ j \in \mathcal{Q} \;\middle|\; \frac{Z_j^{\mathrm{U}} - x_j + S_j d_j}{y_j^*} > r \right\} \quad \text{for any } r > 0.$$
>
> Fix a parameter r^* in the range $[0, 1]$ and then follow these steps:
>
> 1. *Sprint with type i until $x_i = Z_i^{\mathrm{U}}$.*
> 2. *Cruise with type i until $\mathcal{R}(\boldsymbol{x}, r^*)$ is nonempty.*
> 3. *Change over to type j^*, where*
>
> $$j^* = \underset{\substack{j \in \mathcal{R}(\boldsymbol{x}, r^*) \\ j \neq i}}{\operatorname{argmax}} \left\{ \frac{Z_j^{\mathrm{U}} - x_j + S_j d_j}{y_j^*} \right\}.$$
>
> 4. *Set $i = j^*$ and go to Step 1.*

The parameter r^* plays the same role as in the statement of the HZP (Policy 2.4), while y_j^* is determined from a cost bound like the one used in PKP but that incorporates the possibility of cruising (see Section 4.4.1). (Although the statement of LOP given here is slightly different than that in the original publication, the two forms are equivalent when

$\boldsymbol{Z}^{\mathrm{U}} = 0$. This can be seen by multiplying the numerator and denominator by τ_j on each of the ratios in Step 3.)

Lan and Olsen's policy has been shown to work very well in many cases (Lan and Olsen 2006). Motivated by these results, we will argue later that a very sensible choice of $\boldsymbol{\Delta Z}$ is based on the same cost bound used in LOP. With this choice, both policies behave very similarly inside the hedging zone, and their main difference occurs outside of it, where the HZP relies on the prioritization scheme for selecting changeovers.

2.4.4 Gallego's Recovery Policy (GRP)

Up to this point, all the policies considered generate the production sequence dynamically, by selecting at each decision epoch which item to produce next based on the current system state. Gallego (1990) proposes instead a policy that always follows a fixed periodic sequence of the items (e.g., for $N = 3$ one sequence could be 1-2-3-1-2), and varies the production times as a function of the current surplus state. Since the policy was designed to recover a target (open-loop) schedule in the absence of further disruptions, we will refer to this heuristic as Gallego's Recovery Policy (GRP).

Although for some systems GRP behaves like a base-stock policy (Gallego 1994), in general this policy is not in the CC Class. Nevertheless, it has the attractive property that in many cases it recovers optimally from a *single* disruption over all policies that follow the *same* fixed sequence. The statement of GRP requires some further derivations and notation, and thus we defer it to Chapter 4.

2.4.5 The Corridor Policy

For completeness, we outline here yet another non-CC Class policy that has been proposed in the literature. The Corridor Policy, developed by Sharifnia et al. (1991), defines a set

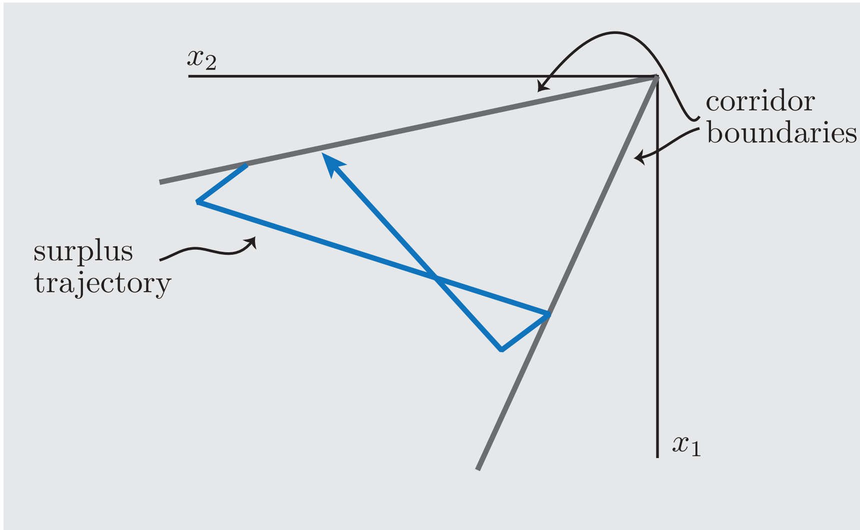


Figure 2-10: Schematic diagram of the Corridor Policy for a two-part-type system. The surplus trajectory bounces on the boundaries of the corridor.

of planes in surplus space that form a corridor along which the surplus vector is intended to move. Whenever the surplus trajectory hits one of these planes, a changeover is made and the system sprints with the new setup. Each plane has a label that indicates which changeover to do upon hitting it. Fig. 2-10 shows a schematic diagram of a typical surplus trajectory generated by this policy for $N = 2$.

While the Corridor Policy does not incorporate cruising, when the corridor planes are perpendicular to the surplus-space axes, the policy behaves like a CC policy. Also, an attractive feature of this policy is that the maximum length of any production run can be limited through an adequate corridor design (compare with CC Class policies, where runs get longer as the surplus deviations vector $\boldsymbol{y}$ increases). On the other hand, due to its general form, the policy is hard to analyze and implement in practice, and there are few guidelines on how to select the corridor planes to obtain good performance.

2.4.6 Comparison of Setup Zones

We have seen that all CC Class policies make changeover and cruising decisions at planes of the form $x_i = Z_i^{\mathrm{U}}$ (i.e., when the surplus is in $\partial\mathcal{X}$). Thus, as mentioned earlier, CC policies distinguish themselves by their setup zones. To get a better understanding of the differences between all the CC Class policies discussed in this chapter, we depict in Figure 2-11 their corresponding setup zones at the plane $x_1 = Z_1^{\mathrm{U}}$ for a three-part-type system with $\mathsf{P}(2) > \mathsf{P}(3)$.

We can see in the figure that the setup zones in (a) and (b) only differ inside a box located on the upper right corner, which corresponds to the hedging zone. For the first case, the cruising HZP states that the system should produce at the demand rate when the surplus lies inside this box, while the non-cruising HZP selects the part type with the largest weighted surplus deviation. The general-form HZP setup zones, shown in (c), are drawn for the same hedging zone as in the previous cases but now, since $r^* = 0.5$, cruising only occurs when the surplus is much closer to the upper right hand corner than in (b). The CLB policy in (d) always splits the plane symmetrically into two setup zones, while PK in (e) sets the slope and location of the boundary separating the two setup zones based on the cost bound to be discussed in Chapter 4. Finally, LOP in (f) has a similar structure as PK but calculates the boundary slightly differently and incorporates cruising. Comparing LOP with the general form HZP, we note that their setup zones are very similar *inside the hedging zone* when $\boldsymbol{\Delta Z}$ is selected as described in Section 4.4.2. Thus, the HZP can be considered as a hybrid policy that combines the qualities of LOP close to the hedging zone and applies a prioritization-based rule far from it.

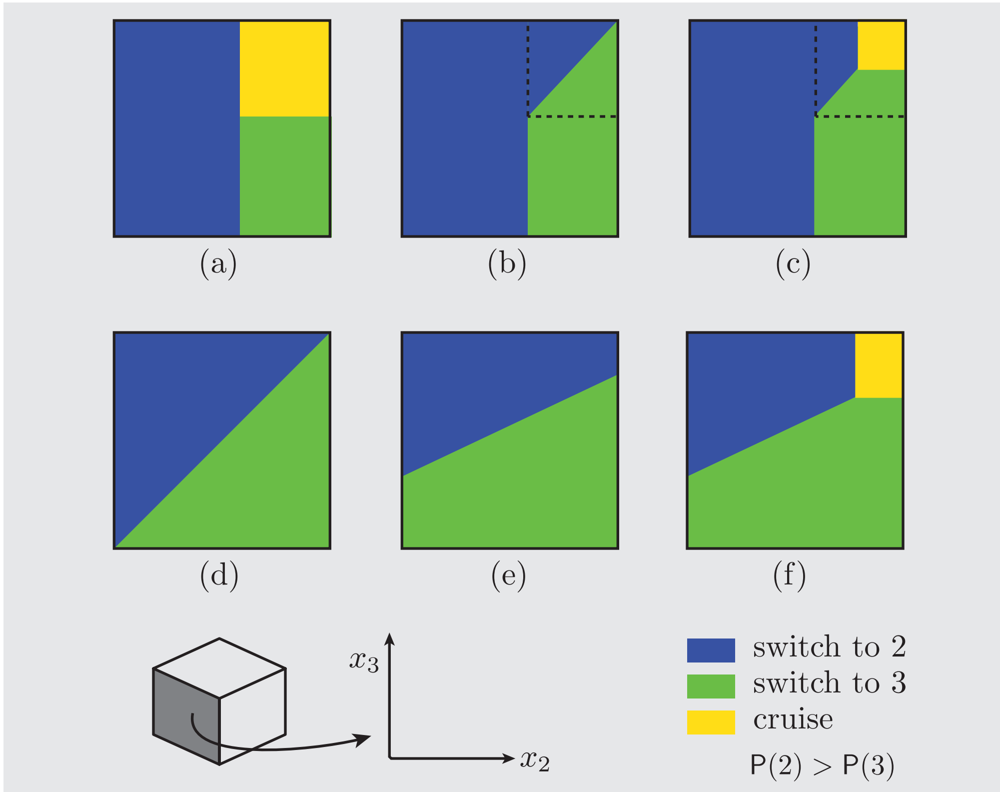


Figure 2-11: Comparison of the setup zones on the plane $x_1 = Z_1^{\mathrm{U}}$ for a three-part-type system. The CC-policies compared are: (a) HZP with cruising, (b) Non-cruising HZP, (c) HZP General Form with $r^* = 0.5$, (d) CLB, (e) PKP, and (f) LOP with $r^* = 0.5$.

2.5 Summary

In this chapter we have formally stated the scheduling problem through the description of a deterministic system. We justified the choice of model by arguing that controllers are frequently designed assuming deterministic dynamics with the intention of providing robustness to the actual stochastic system. After stating the model, we described a class of commonly-used policies called the Clearing Cruising Class. In this class, the current setup is always produced at full capacity (sprinting) until its base stock level is reached, and then the system may hold this level for some time before switching setups, by producing at the demand rate (cruising).

Based on the CC Class, we then developed the Hedging Zone Policy (HZP), which constitutes the central theme of this thesis. The HZP is a policy that uses two parameters per part type for selecting changeovers, as well as pre-assigned priorities. The first parameter plays the role of the base stock level, while the second parameter helps identify which part types are ready for a new production run. We stated a cruising and a non-cruising version of the policy, which can be selected based on the problem at hand, and we provided a compact expression of the policy that transitions smoothly between the two. Finally, we put the HZP in context by stating previously-studied policies that are also directly applicable to our problem, and whose performance will be compared with the HZP.

Before dealing with performance comparisons, in the next chapter we will address the important issue of stability. One of the facts about the HZP is that its stability is not guaranteed for all parameter values. Therefore, we need to develop simple ways to ensure that the choice of $\boldsymbol{\Delta Z}$ will not lead to a system that fails to meet the demand of all of its part types with bounded surplus.

Chapter 3

Stability Theory

In this chapter, we develop a stability theory for our model and the CC Class policies described in the previous chapter. After introducing the problem of unstable behavior in the HZP through an example, we use Lyapunov's direct method to state a sufficient condition that guarantees production of all part types with bounded surplus. We then show how this condition translates into a simple expression that constrains the set of values of the HZP parameters. Through a series of numerical experiments, we study the conservativeness of this condition and then formally show that it can be relaxed by considering only the $N-1$ highest-priority part types. We conclude by extending our results to the case of general priorities as well as to the other CC Class policies covered in Chapter 2.

3.1 Introduction

Stability is a central issue in the design of any control system, as it ensures that its response to a bounded input remains bounded (Dorf and Bishop 2005, p. 312). For our scheduling system, stability means that the system is able to meet the demand of all products without requiring an infinite amount of inventory or accumulating an increasing number of backlogged orders (a precise definition of stability is given in Section 3.2.1).

While, clearly, the inventory or backlog in an actual factory can never become infinite (after all, material is finite and there is only a certain number of orders that can be backlogged before the company loses its reputation!), an unstable policy that tends to ignore some of the part types or perform too many changeovers will soon become distrusted by the plant managers. Therefore, it is important to develop guidelines that guarantee from the outset that the intended policy behaves in a stable fashion.

As mentioned in Chapter 2, the Capacity Condition $\rho < 1$ is *necessary* for the stability of our model. To see why this is true, let $T_i(t)$ denote the accumulated production time of type i up to time t and $S_i(t)$ the accumulated setup time into that type. Assuming that the system never sits idle, it follows that

$$\sum_{i=1}^{N} \big(T_i(t) + S_i(t)\big) = t, \tag{3.1}$$

for all t. Furthermore, if the system always operates at full capacity, then $P_i(t) = T_i(t)\mu_i$ and, using the fact that $P_i(t) = d_i t + x_i(t)$, we can write (3.1) as

$$\sum_{i=1}^{N} \left(\frac{d_i t + x_i(t)}{\mu_i} + S_i(t)\right) = t.$$

Dividing the previous expression by t and replacing $\sum_i d_i/\mu_i$ by ρ, we get

$$\sum_{i=1}^{N} \left(\frac{x_i(t)}{\mu_i t} + \frac{S_i(t)}{t}\right) = 1 - \rho. \tag{3.2}$$

Now suppose $\rho > 1$. In this case, the left hand side in (3.2) is negative for all t and, since $S_i(t) \geq 0$, this means that the backlog of at least one of the part types must be growing linearly with t (i.e., $x_i(t)$ becoming more negative). Similarly, for the case in which $\rho = 1$, (3.2) implies that the sum of all terms $x_i(t)/\mu_i + S_i(t)$ is equal to 0 for all t. But, since the total accumulated setup time $\sum_i S_i(t)$ always grows without bound, this sum can only be 0 if the total backlog is also growing without bound.[1] Thus, we conclude that the system cannot be stable if $\rho \geq 1$.

While for some policies the Capacity Condition is both necessary and sufficient for stability (see, e.g., Takagi 1988, Perkins and Kumar 1989, and Section 3.4.1), this is not true in general. In the HZP, if the hedging zone is not chosen well, the system could ignore some of the lowest-priority part types; that is, at some point, it would no longer change over and produce these part types. This behavior is exemplified in Fig. 3-1, where we show the surplus versus time plots for the three-part-type example of Section 2.3.1 with a reduced hedging zone. Notice how types 1 and 2 get produced with bounded surplus but type 3 (the lowest-priority part) is never produced and its backlog grows without bound.

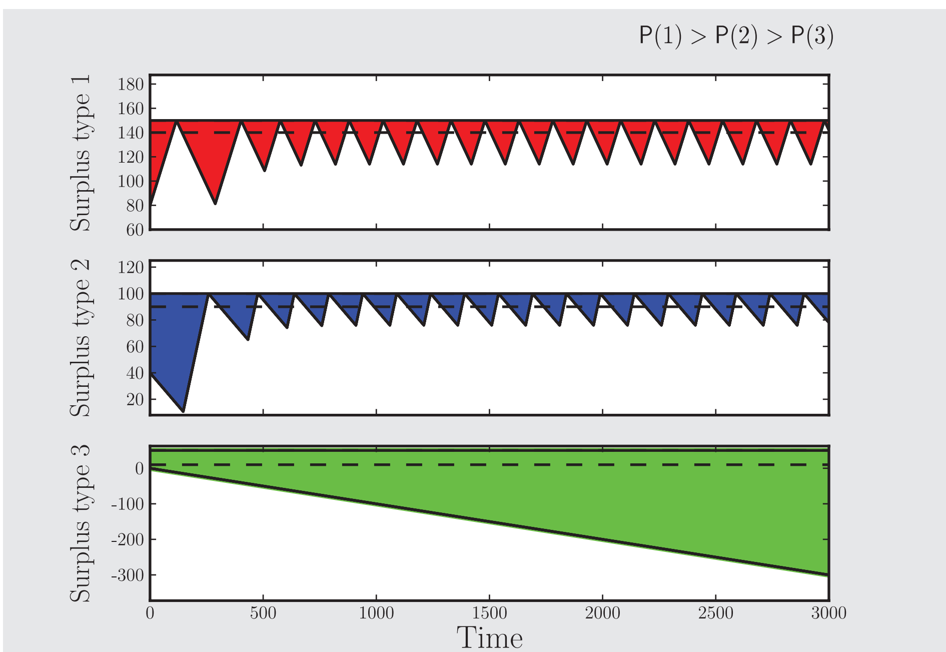


Figure 3-1: Surplus versus time plots for the three-part-type system of Section 2.3.1 with a smaller hedging zone. (Here, $\boldsymbol{\Delta Z} = (10, 10, 40)$ instead of $(80, 50, 40)$.)

[1]Given that we are considering systems with positive setup times, $\sum_i S_i(t)$ would only be bounded if the system stopped changing setups altogether, which also corresponds to an unstable system.

The cause of this behavior can be seen more clearly in Fig. 3-2, which shows a projection of the surplus trajectory in x_1-x_2 space. It is seen that the system settles into a bow-tie shaped limit cycle that does not touch the hedging zone $\mathcal{Z}_2$ formed by these two part types, where

$$\mathcal{Z}_2 = \left\{ \boldsymbol{x} \in \partial\mathcal{X} \;\middle|\; x_i \in [Z_i^{\mathrm{L}}, Z_i^{\mathrm{U}}], \quad i = 1, 2 \right\}.$$

Since part type 3 only gets selected if the higher-priority types 1 and 2 are within their hedging zones at the time of the changeover decision, the system never produces this part.

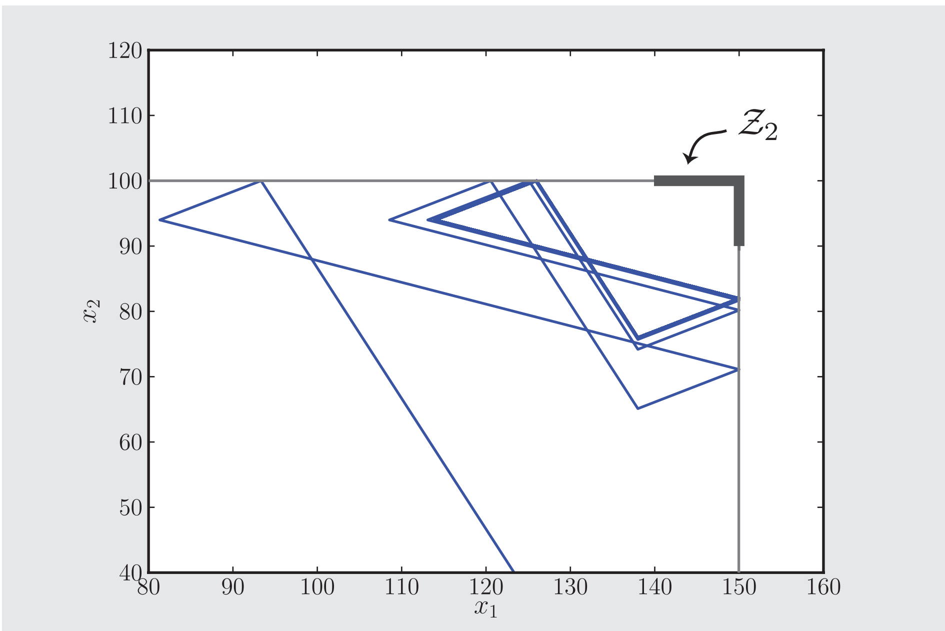


Figure 3-2: Surplus trajectory in x_1-x_2 space of the three-part-type system of Section 2.3.1 with a reduced hedging zone. Notice how the trajectory settles into a bow-tie limit cycle that lies outside of $\mathcal{Z}_2$.

Therefore, in order to produce all $i \in \mathcal{Q}$ under the HZP, the parameters ΔZ_i must be sufficiently large so that the surplus trajectory is able to reach the hedging zone formed by the $N - 1$ highest-priority part types (i.e., $\mathcal{Z}_{N-1}$), triggering in this way production of the lowest-priority part. However, as addressed in the subsequent chapters, the size of the hedging zone also affects the cost performance of the system, and thus it is desirable to determine just *how large* we need to make the hedging zone to avoid the behavior of Fig. 3-2.

Preventing Unstable Behavior

For systems with $N = 3$, we can determine exact conditions for preventing the system from ignoring the lowest-priority part type, thanks to the simple nature of the surplus trajectory generated by the other two part types (refer to Appendix A for a rigorous derivation). Suppose that parts are labelled such that $\mathsf{P}(1) > \mathsf{P}(2) > \mathsf{P}(3)$. Then, if the surplus trajectory of parts 1 and 2 satisfies either one of the two conditions depicted in Fig. 3-3, their limit cycle will have at least one point inside hedging zone $\mathcal{Z}_2$. This ensures that $\mathcal{Z}_2$ is reachable by the x_1-x_2 trajectory, which in turn means that part type 3 will get produced. As derived in Appendix A, the conditions shown in the figure can be expressed mathematically as

$$\begin{aligned} Z_1^U - Z_1^L &> \frac{S_{12}(1-\rho_1) + S_{21}\rho_2}{1-\rho_1-\rho_2} d_1, \text{ and/or} \\ Z_2^U - Z_2^L &> \frac{S_{21}(1-\rho_2) + S_{12}\rho_1}{1-\rho_1-\rho_2} d_2. \end{aligned} \tag{3.3}$$

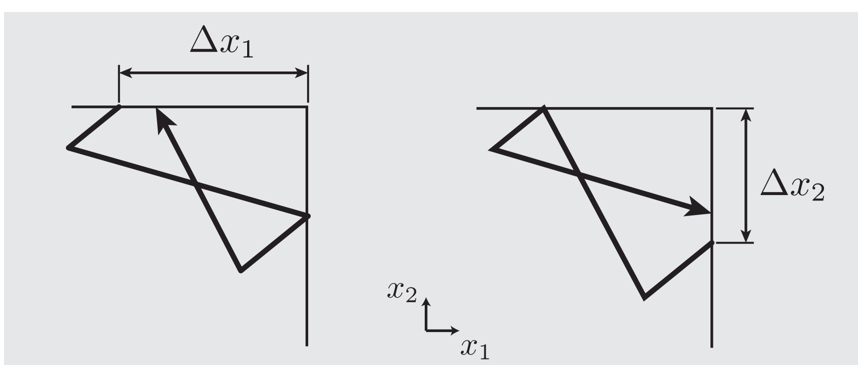


Figure 3-3: Conditions on the surplus trajectory of x_1 and x_2 that guarantee production of type 3. On the left diagram, a trajectory starting at $x_1 = Z_1^{\mathrm{L}}$, $x_2 = Z_2^{\mathrm{U}}$ ends up closer to Z_1^{U} after one full 1-2 cycle. (A similar explanation applies to the right diagram.)

The condition in (3.3) is both necessary and sufficient for the system to produce all three part types infinitely often. However, two important problems remain. First, as mentioned, our derivation takes advantage of the fact that the trajectory of the first two part types has a unique and easy-to-compute limit cycle (note that the production sequence for these parts will always be 1-2-1-2-1-...). For larger systems, however, we can no longer predict the production sequence *a priori* (it emerges from the dynamics of the system), making it hard to determine the long-term surplus trajectory. Additionally, the second problem is that even if we could find a necessary and sufficient condition to guarantee that no part type is ignored indefinitely, we would still need to show that the surplus remains bounded (i.e., that the system is stable).

To get a better idea of the complexity of finding a necessary and sufficient stability condition that addresses the problems mentioned above, consider Fig. 3-4, where we have computed experimentally the *stability boundary*, separating stable and unstable systems in $\boldsymbol{\Delta Z}$-space, for three different systems with $N = 3, 5$, and 10. On each plot, the axes correspond to the two components of the vector $\boldsymbol{\Delta Z}$ that were varied while the rest were held fixed at the values indicated. Each point in the plots represents a particular value of $\boldsymbol{\Delta Z}$ and, through simulation, we determined if such a value led to instability. Each simulation was long enough to include 10^5 production runs, which allowed the system to reach steady state. The simulation was then continued until we obtained two consecutive batches of 10^4 production runs each. For each of these two batches and each part type, we determined the maximum backlog level that the trajectory achieved; if this value did not increase by more than 0.1%, the part type was considered to be produced with bounded surplus. A system in which *all* part types were produced with bounded surplus was considered stable. In our experiments, all of the simulations started with an initial condition of $\boldsymbol{x}_0 = \boldsymbol{Z}^{\mathrm{U}}$. (Notice that the determination of the *true* stability boundary is further complicated by the fact that, even if steady state was reached, we would need to repeat this experiment for *all* initial conditions. However, the results provide an approximate representation of the general shape and location of the stability zone for different systems.)

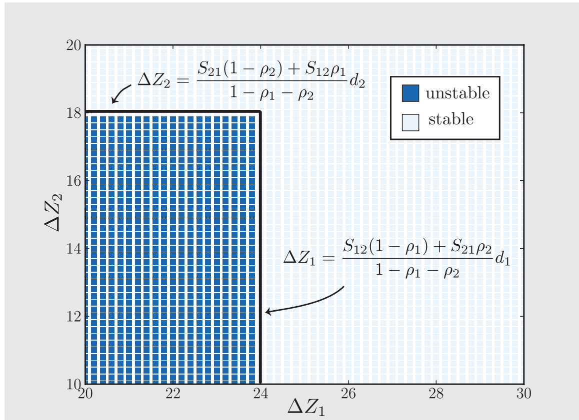
The irregular shape of the boundary separating stable and unstable systems for the cases $N = 5$ and $N = 10$ in Fig. 3-4 suggests that deriving a necessary and sufficient stability condition for large systems is an extremely complex problem. This motivates the search for an easy-to-evaluate approximation to the boundary, as developed in the next section. We can also see that the stability boundary for the system with $N = 3$ corresponds to a box whose sides agree exactly with (3.3). In other words, the boundary of stability coincides exactly with the boundary that separates systems that produce all three part types from systems that behave like in Fig 3-1. Moreover, it turns out that in all of our experiments (i.e., with N =3, 5, and 10), *the unstable systems were so because they ignored some of the low-priority part types*. That is, there was not a single instance in which the surplus of *all* part types was unbounded. This suggests that, as long as we guarantee that system does not ignore any of the part types indefinitely, the HZP will be stable (a claim that is proved rigorously in Section 3.3.3).

3.2 Lyapunov Stability Theory

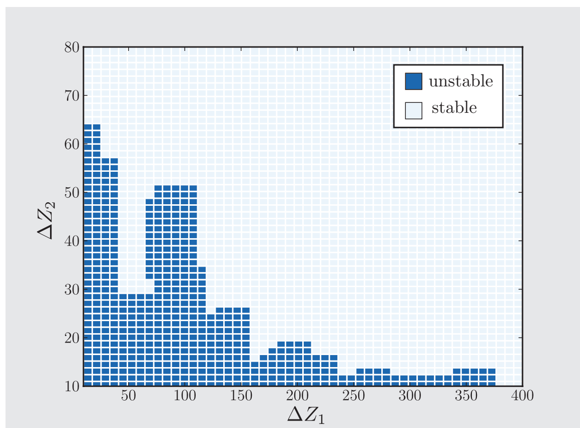
3.2.1 Notion of Stability

As stated earlier, a stable system is able to meet the demand for all of its products and requires finite inventory or backlog to do so. This means that the cumulative production process $P_i(t)$ tracks the cumulative demand $D_i(t)$ closely, for all $i \in \mathcal{Q}$.

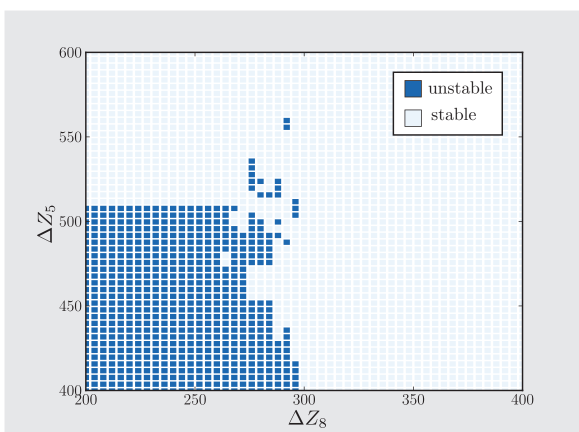
The ability for a system to behave in this way depends on both the structure of the policy used and the particular choice of parameters. For example, as we showed in the previous section, a system Σ can be stable under the HZP for some values of $\boldsymbol{\Delta Z}$ and unstable for



(a) $N = 3$ system of Section 2.3.1. ($\Delta Z_3 = 40$.)



(b) $N = 5$ system of Table D.1 in Section D.2



(c) $N = 10$ system of Table D.2 in Section D.2.

Figure 3-4: Experimental determination of the stability boundary for different systems. Each square marker corresponds to a different system configuration whose stability was determined as explained in the text.

others. Therefore, in order to make more explicit the distinction between the system and the policy used to control it, we define below the concept of a closed-loop system Σ_π.

Definition 3.2.1 (Closed-Loop System) *A closed-loop system Σ_π is a system Σ operated under some policy π, which has a particular choice of values for its parameters and specifies the instantaneous production rate $\boldsymbol{u}(t)$ and the changeover epochs as a function of the current state.*

We now provide a precise definition of stability of a closed-loop system Σ_π.

Definition 3.2.2 (Stable System) *A system Σ_π is stable if there exists an $M > 0$ such that, for any $\boldsymbol{x}(t_0)$, there is a time $\tilde{t} = \tilde{t}(\boldsymbol{x}(t_0))$ with the property that $\|\boldsymbol{x}(t)\| \leq M$ for all $t \geq \tilde{t}(\boldsymbol{x}(t_0))$.*

The above definition implies that, regardless of the initial condition, after some finite time the system's surplus is guaranteed to get within a finite-size box and stay there indefinitely. Therefore, we can say that all trajectories of the system are *ultimately bounded* with bound M (note that this bound does not depend on the initial state). While there exist many other related definitions of stability in the literature (see, e.g., Michel et al. 2008), Definition 3.2.2 is adequate for our purposes and will be the only one used throughout this chapter.

3.2.2 Lyapunov's Direct Method

Lyapunov's direct method is a commonly used technique for demonstrating the stability of nonlinear dynamic systems (Luenberger 1979, p. 319). The method consists of finding an appropriate function of the system's state that is known to be non-increasing over all trajectories. In physical applications, this function typically measures the energy in the system, and thus the Lyapunov method can be used to verify that the system dissipates energy and reaches an equilibrium. For general applications, the function can be simply considered to summarize the system's state description (which may be a large-dimension vector) by associating a representative scalar quantity to it. Lyapunov's method has also found wide use in the analysis of stochastic queueing networks, and a brief review of this literature is provided in Section 5.2.

We begin by defining the general form of the candidate Lyapunov functions to be considered. Recall that $\mathcal{X}$ denotes the set of all $\boldsymbol{x} \leq \boldsymbol{Z}^{\mathrm{U}}$ and that $\partial\mathcal{X}$ is the subset in which at least one surplus component is exactly at its base stock level. Consider a function $V : \mathcal{X} \mapsto \mathbb{R}$ of the form

$$V(\boldsymbol{x}) = \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{Z}^{\mathrm{U}} - \boldsymbol{x}), \tag{3.4}$$

where $\boldsymbol{\phi}$ is an $N \times 1$ vector with all components strictly greater than zero.

Notice that $V(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} < \boldsymbol{Z}^{\mathrm{U}}$, so that the function is positive over its domain except at point $\boldsymbol{x} = \boldsymbol{Z}^{\mathrm{U}}$, where it equals 0. Furthermore, we can see that $V(\boldsymbol{x}) \to \infty$ as $(\boldsymbol{Z}^{\mathrm{U}} - \boldsymbol{x}) \to \infty$, which implies that if $V(\boldsymbol{x})$ is bounded, then so is $\boldsymbol{x}$ (i.e., the function is *radially unbounded*). Thus, we can see that $V(\boldsymbol{x})$ serves as a measure of the surplus state's distance from $\boldsymbol{Z}^{\mathrm{U}}$.

Function $V(\boldsymbol{x})$ can be used to prove the stability of the system by considering the values that it takes along a typical trajectory of $\boldsymbol{x}$. If these values are known to be always decreasing between runs (except possibly for runs starting at a bounded set of states) then we can show that the system is stable. This idea is illustrated for the $N = 3$ case in Fig. 3-5 and stated formally in the next theorem, which constitutes a sufficient (but not necessary) stability condition. (Recall that $\boldsymbol{\psi}(\cdot)$ is the discrete-time map such that, omitting the setup state variable, we have $\boldsymbol{x}_{n+1} = \boldsymbol{\psi}(\boldsymbol{x}_n)$.)

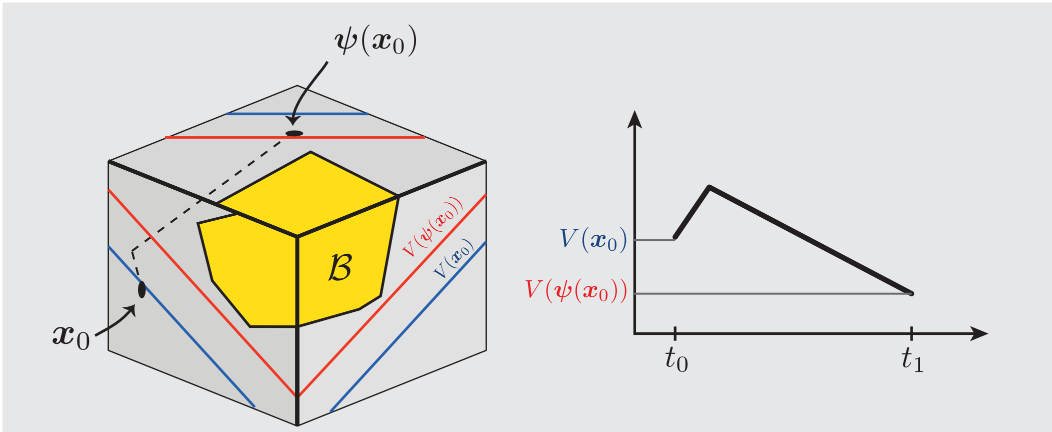


Figure 3-5: Schematic description of Stability Theorem 3.2.3. On the left, we see a typical run in surplus space (as before, each face corresponds to a plane of the form $x_i = Z_i^{\mathrm{U}}$) and on the right we see the corresponding values that $V(\boldsymbol{x}(t))$ takes over time. According to Theorem 3.2.3, every run that starts outside of $\mathcal{B}$ must conclude with a lower value of $V(\boldsymbol{x})$.

> **Theorem 3.2.3 (Stability Theorem)** *Let $\epsilon > 0$. A closed-loop system Σ_π operated under a CC Class policy π is stable if there exists a bounded set $\mathcal{B} \subset \partial\mathcal{X}$ and a function $V(\boldsymbol{x})$ as defined in (3.4) such that, for all $\boldsymbol{x}_0 \in \partial\mathcal{X} \setminus \mathcal{B}$,*
>
> $$V(\boldsymbol{\psi}(\boldsymbol{x}_0)) - V(\boldsymbol{x}_0) \leq -\epsilon.$$

Proof: *First note that for any $n \geq 0$ and $t \in [t_n, t_{n+1}]$,*

$$V(\boldsymbol{x}(t)) \leq \max\left\{V(\boldsymbol{x}_n) + \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d}\max_{i,j} S_{ij}\,,\ V(\boldsymbol{x}_{n+1})\right\}, \tag{3.5}$$

where we have used the fact that during a changeover the Lyapunov function increases at rate $\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d}$. The expression in (3.5) implies that if the discrete-time sequence $\{V(\boldsymbol{x}_n)\ ;\ n \geq 0\}$ is bounded, then $V(\boldsymbol{x}(t))$ will also be bounded for all $t \geq 0$.

Now suppose that the assumptions in the theorem are satisfied and define the following quan-

tities (which are finite because $\mathcal{B}$ is bounded)

$$V_{\mathcal{B}} = \sup_{\boldsymbol{x}\in\mathcal{B}} V(\boldsymbol{x}),$$

and

$$V_{\boldsymbol{\psi}(\mathcal{B})} = \sup_{\boldsymbol{x}\in\mathcal{B}} V(\boldsymbol{\psi}(\boldsymbol{x})).$$

$V_{\mathcal{B}}$ represents the maximum value that the Lyapunov function can take inside set $\mathcal{B}$ and $V_{\boldsymbol{\psi}(\mathcal{B})}$ is the maximum value it can take on the forward image of $\mathcal{B}$. Let $V^* = \max\{V_{\mathcal{B}}, V_{\boldsymbol{\psi}(\mathcal{B})}\}$ and consider the sequence $\{\, U_n \; ; \; n \geq 0 \,\}$, where

$$U_n = \max\{V(\boldsymbol{x}_n), V^*\}.$$

We know that $V(\boldsymbol{x}_n)$ must decrease by at least ϵ whenever $V(\boldsymbol{x}_n) > V^*$ because this means that $\boldsymbol{x}_n$ will necessarily be outside of set $\mathcal{B}$. Furthermore, notice that if for some $m > 0$ the surplus $\boldsymbol{x}_m$ lands in $\mathcal{B}$, then the Lyapunov function will remain bounded by V^* for all subsequent times (i.e., $V(\boldsymbol{x}_n) \leq V^*$ for all $n \geq m$). Thus, the sequence $\{\, U_n \; ; \; n \geq 0 \,\}$ is ultimately bounded by V^*.

Now, since for all n we have $V(\boldsymbol{x}_n) \leq U_n$, it follows that $\{\, V(\boldsymbol{x}_n) \; ; \; n \geq 0 \,\}$ is also ultimately bounded by V^* for any $\boldsymbol{x}_0$. Using (3.5)*, we see then that $V(\boldsymbol{x}(t))$ is bounded by $V^{**} = V^* + \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d}\max_{i,j} S_{ij}$ for sufficiently large t. Finally, let*

$$M = \sup_{\substack{V(\boldsymbol{x})\leq V^{**} \\ \boldsymbol{x}\in\mathcal{X}}} \|\boldsymbol{x}\|,$$

which is finite. We conclude then that $\|\boldsymbol{x}(t)\|$ will eventually become less than M for any $\boldsymbol{x}_0$ or, in other words, all surplus trajectories are ultimately bounded by M. □

The attractiveness of Theorem 3.2.3 lies in the fact that, in order to verify if Σ_π is stable, we only need to look one step forward in the discrete-time trajectory of any state that is not in $\mathcal{B}$ and ensure that the net change in the Lyapunov function is negative. This is certainly a problem more amenable to analysis than having to consider complete trajectories. (Furthermore, notice that the proof of the theorem prescribes a way to obtain a bound on the long-term trajectories of the system, a fact that may be useful for obtaining estimates of the required buffer sizes or base stock levels of the system.)

The following corollary to Theorem 3.2.3 will be useful for relaxing our stability condition in Section 3.3.4.

Corollary 3.2.4 (Reachability of $\mathcal{B}$) *For any $\alpha > 0$ and $\|\boldsymbol{x}(t_0)\| \leq \alpha$, with $\boldsymbol{x}(t_0) \in \partial\mathcal{X}$, there exists a finite time $\tilde{t}(\alpha)$ such that the surplus state will have visited $\mathcal{B}$ at least once during this period. That is,*

$$\boldsymbol{x}(t) \in \mathcal{B} \text{ for some } t \leq t_0 + \tilde{t}(\alpha).$$

Proof: *Let V_α be defined as*

$$V_\alpha = \sup_{\substack{\|\boldsymbol{x}\| \leq \alpha \\ \boldsymbol{x} \in \partial\mathcal{X}}} V(\boldsymbol{x}).$$

Since every run decreases the Lyapunov function by at least ϵ, we can upper bound the number of production runs that the trajectory of $\boldsymbol{x}(t_0)$ will need in order to reach $\mathcal{B}$ by $\tilde{n}(\alpha)$, where

$$\tilde{n}(\alpha) = \left\lceil \frac{V_\alpha}{\epsilon} \right\rceil.$$

Furthermore, we can bound all future values that the Lyapunov function takes along the trajectory since

$$V(\boldsymbol{x}_n) \leq \max\{V_\alpha, V^*\} \quad \forall\ n \geq 0,$$

where V^* is defined as in the proof of Theorem 3.2.3 and $\boldsymbol{x}_0 = \boldsymbol{x}(t_0)$. Now, consider the set of states $\boldsymbol{x} \in \partial\mathcal{X}$ such that $V(\boldsymbol{x}) \le \max\{V_\alpha, V^*\}$. Since this set is bounded, the time it takes to complete a single production run starting from any $\boldsymbol{x}$ in the set is finite (an expression for this quantity is given in (3.6), *in Section 3.3.1) and bounded by some $\Delta t(\alpha)$.*

Therefore, $\boldsymbol{x}(t_0)$ will perform no more than $\tilde{n}(\alpha)$ runs of duration no greater than $\Delta t(\alpha)$ before reaching $\mathcal{B}$. Thus, we can set

$$\tilde{t}(\alpha) = \tilde{n}(\alpha)\Delta t(\alpha)$$

as the upper bound on the time it takes to visit $\mathcal{B}$. □

We conclude this section by noting that the Lyapunov function considered in Theorem 3.2.3 and its corollary could be more general, as long as it satisfies the same properties that were discussed in connection with (3.4). While using a richer class of functions $V(\cdot)$ can lead to a less-conservative stability condition, this comes at the expense of a harder derivation and evaluation of the stability condition.

3.3 Stability of the HZP

3.3.1 Sufficient Stability Condition

We now use Stability Theorem 3.2.3 to obtain an easy-to-evaluate sufficient stability condition for the case in which the system is operated under the HZP. We will assume throughout this section that *priorities are unique* and that parts are ordered in terms of decreasing priority, so that $\mathsf{P}(1) > \mathsf{P}(2) > \cdots > \mathsf{P}(N)$.

The Stability Theorem gives us the liberty of selecting $\mathcal{B}$, as long as it is a bounded subset of

$\partial\mathcal{X}$. Therefore, we will define this set so that it contains all states that are in the hedging zone plus any other state that visits the hedging zone at some point during its next production run. Mathematically, $\mathcal{B}$ can be written as

$$\mathcal{B} = \mathcal{Z}_N \bigcup \{ \boldsymbol{x}(t_0) \in \partial\mathcal{X} \ \mid \boldsymbol{x}(t) \in \mathcal{Z}_N, \text{ for some } t \in [t_0, t_1] \},$$

where we see that, since $\mathcal{Z}_N$ is bounded, $\mathcal{B}$ is as well. Note also that by defining $\mathcal{B}$ in this way we make it clear that we are excluding possible cruising runs from our analysis. This follows because such runs necessary visit $\mathcal{Z}_N$ during $[t_0, t_1]$ and will conclude with $\boldsymbol{x}(t_1)$ *just outside* of this set (recall that, in the cruising version of the HZP, cruising only occurs inside the hedging zone and concludes at the instant when $\boldsymbol{x}$ exits this set).

Consider now a run that starts at state $\boldsymbol{x}_0 \notin \mathcal{B}$ with initial setup i and in which part j is produced (i.e., $\sigma_0 = i$ and $\sigma_1 = j$). The time it takes to complete the run, $t_1 - t_0$, will be given by

$$\Delta t(x_j(t_0), i, j) = S_{ij} + \frac{S_{ij} d_j}{\mu_j - d_j} + \frac{(Z_j^U - x_j(t_0))}{\mu_j - d_j}.$$

The above equation consists of the time it takes to complete the setup change from type i to type j, the time it takes to restore the surplus of type j back to the level $x_j(t_0)$ it had before the changeover, and the time it takes to bring type j to its upper hedging point. Simplifying, we get

$$\Delta t(x_j(t_0), i, j) = \frac{S_{ij}}{1 - \rho_j} + \frac{(Z_j^U - x_j(t_0))}{1 - \rho_j} \tau_j. \tag{3.6}$$

The expression in (3.6) allows us to compute the surplus state at the end of the run, $\boldsymbol{x}_1 = (x_1(t_1), x_2(t_1), \ldots, x_N(t_1))^{\mathrm{T}}$. Recalling that during the run of type j all other surpluses decrease at their demand rate, we have

$$x_k(t_1) = \begin{cases} x_k(t_0) - \Delta t(x_j(t_0), i, j) d_k & \text{if } k \neq j \\ Z_k^{\mathrm{U}} & \text{if } k = j, \end{cases} \tag{3.7}$$

The net change in the Lyapunov function at the end of the run is then

$$V(\boldsymbol{x}_1) - V(\boldsymbol{x}_0) = \Delta V(\boldsymbol{x}_0) = \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{x}_0 - \boldsymbol{x}_1),$$

which, upon substitution of (3.7), gives

$$\begin{aligned} \Delta V(\boldsymbol{x}_0) &= \sum_{k=1, k\neq j}^{N} \phi_k \Big(x_k(t_0) - \big(x_k(t_0) - \Delta t(x_j(t_0), i, j)\, d_k \big) \Big) + \phi_j (x_j(t_0) - Z_j^U) \\ &= \left[\sum_{k=1, k\neq j}^{N} \phi_k d_k \right] \Delta t(x_j(t_0), i, j) - \phi_j (Z_j^U - x_j(t_0)). \end{aligned}$$

The summation inside the brackets can be represented more compactly as $\boldsymbol{\phi}^T \boldsymbol{d} - \phi_j d_j$ and substituting (3.6), we get

$$\Delta V(\boldsymbol{x}_0) = (\boldsymbol{\phi}^T \boldsymbol{d} - \phi_j d_j) \left[\frac{S_{ij}}{1 - \rho_j} + \frac{(Z_j^U - x_j(t_0))\tau_j}{1 - \rho_j} \right] - \phi_j (Z_j^U - x_j(t_0)).$$

This expression simplifies to

$$\Delta V(\boldsymbol{x}_0) = \frac{\boldsymbol{\phi}^T \boldsymbol{d}\, \tau_j - \phi_j}{1 - \rho_j} (Z_j^U - x_j(t_0)) + \frac{\boldsymbol{\phi}^T \boldsymbol{d} - \phi_j d_j}{1 - \rho_j} S_{ij}, \tag{3.8}$$

which shows that the change in the Lyapunov function is only dependent on the initial surplus level of the part type being produced, $x_j(t_0)$, and that the relationship is linear.

Based on (3.8), we now transform the condition in Theorem 3.2.3 into a set of constraints on the hedging zone that will ensure that the maximum possible change in the Lyapunov function is less than $-\epsilon$. First note that Theorem 3.2.3 will be satisfied as long as

$$\sup_{\substack{\boldsymbol{x}_0 \in \partial \mathcal{X} \\ \boldsymbol{x}_0 \notin \mathcal{B}}} \Delta V(\boldsymbol{x}_0) \leq -\epsilon.$$

Rather than solving this optimization problem by first obtaining an expression for $\mathcal{B}$ (which will determine the set of feasible solutions), we will use the fact that $\mathcal{Z}_N \subset \mathcal{B}$. This implies that if the maximum change in Lyapunov function is negative outside of $\mathcal{Z}_N$, then it will also be outside of $\mathcal{B}$. Thus, it suffices to ensure that

$$\sup_{\substack{\boldsymbol{x}_0 \in \partial \mathcal{X} \\ \boldsymbol{x}_0 \notin \mathcal{Z}_N}} \Delta V(\boldsymbol{x}_0) \leq -\epsilon.$$

Using (3.8), we get

$$\sup_{\substack{\boldsymbol{x}_0 \in \partial \mathcal{X} \\ \boldsymbol{x}_0 \notin \mathcal{Z}_N}} \Delta V(\boldsymbol{x}_0) = \max_{x_j \leq Z_j^{\mathrm{L}}} \frac{\boldsymbol{\phi}^T \boldsymbol{d}\, \tau_j - \phi_j}{1 - \rho_j} (Z_j^U - x_j) + \frac{\boldsymbol{\phi}^T \boldsymbol{d} - \phi_j d_j}{1 - \rho_j} S_{ij}. \tag{3.9}$$

Now, consider the case in which the factor $\boldsymbol{\phi}^T \boldsymbol{d}\, \tau_j - \phi_j$ on the first term of the right hand side of (3.9) is positive. Given that the second term is always positive, it follows that $\Delta V(\boldsymbol{x}_0)$ cannot be less than $-\epsilon$ for any value of $x_j \leq Z_j^{\mathrm{U}}$. Thus, $\boldsymbol{\phi}^T \boldsymbol{d}\, \tau_j - \phi_j$ must be negative for the Lyapunov function to be valid. With this restriction, the solution to (3.9) is

$$\Delta V^* = -\frac{\phi_j - \boldsymbol{\phi}^T \boldsymbol{d}\, \tau_j}{1 - \rho_j} \Delta Z_j + \frac{\boldsymbol{\phi}^T \boldsymbol{d} - \phi_j d_j}{1 - \rho_j} S_{ij}, \tag{3.10}$$

which should be no greater than $-\epsilon$ to satisfy Theorem 3.2.3. This is the stability condition that we are looking for and is summarized below.

Condition 3.3.1 (Sufficient Stability Condition) *A closed-loop system Σ_π operated under the HZP will satisfy Theorem 3.2.3 and thus be stable if there exists a positive vector $\boldsymbol{\phi}$ such that, for all $j \in \mathcal{Q}$,*

$$\phi_j - \boldsymbol{\phi}^T \boldsymbol{d} \tau_j > 0, \tag{3.11}$$

and if the hedging zone satisfies for all $i, j \in \mathcal{Q}$

$$\Delta Z_j \geq \frac{\epsilon(1-\rho_j)}{\phi_j - \boldsymbol{\phi}^T \boldsymbol{d} \tau_j} + \frac{\boldsymbol{\phi}^T \boldsymbol{d} - \phi_j d_j}{\phi_j - \boldsymbol{\phi}^T \boldsymbol{d} \tau_j} S_{ij}. \tag{3.12}$$

Analysis of Condition 3.3.1

Equation (3.11) constitutes a constraint on the rate of change of $\Delta V(\boldsymbol{x})$ that ensures that *longer runs lead to larger reductions in the Lyapunov function*. The second constraint establishes a minimum length on all runs that start outside of the hedging zone. This length guarantees that the setup time is amortized properly over the production segment of the run, so that the system does not lose too much capacity by doing frequent setup changes.

It is interesting to note that (3.11) is not trivial, in the sense that not any positive vector $\boldsymbol{\phi}$ will satisfy it. However, if $\rho < 1$, we can always find at least one vector that satisfies this constraint. Indeed, let $\boldsymbol{\phi} = \boldsymbol{\tau}$, in which case $V(\boldsymbol{x})$ represents the time that the system needs to work in order to bring all part types to their targets if demand were suddenly interrupted. With this choice, (3.11) is clearly satisfied because

$$\tau_j - \boldsymbol{\tau}^{\mathrm{T}} \boldsymbol{d} \tau_j = (1-\rho)\tau_j > 0.$$

Given a vector $\boldsymbol{\phi}$ that satisfies (3.11) we can always make the hedging zone large enough to satisfy (3.12). We therefore have the following theorem.

Theorem 3.3.2 (Stabilizability of Σ under HZP) *Any system Σ with sufficient capacity (i.e., $\rho < 1$) is stabilizable under the HZP.*

On the other hand, $\boldsymbol{\phi} = \boldsymbol{\tau}$ only gives a subset of the values of $\boldsymbol{\Delta Z}$ that satisfy Condition 3.3.1. Since the performance of the HZP is heavily dependent on the parameters vector $\boldsymbol{\Delta Z}$, we would like to be able to select this vector from the largest possible set of values that satisfy this condition. Such a set will be determined in the next section.

3.3.2 Satisfiability Boundary of the Sufficient Stability Condition

Condition 3.3.1 is expressed in terms of the Lyapunov function vector $\boldsymbol{\phi}$, while the stability of the system is, of course, independent of this vector. It is thus desirable to translate the stability condition into an expression that solely depends on the vector of policy parameters $\boldsymbol{\Delta Z}$. Such an expression will allow us to establish a boundary (i.e., a hyper-surface in $\boldsymbol{\Delta Z}$-space) separating systems that satisfy the sufficient stability condition from systems that do not satisfy it. This surface is what we call the *satisfiability boundary* of Condition 3.3.1.

To simplify our developments, we will ignore possible sequence dependencies and state our equations in terms of the maximum setup time onto type j, denoted by S_{*j}. That is,

$$S_{*j} = \max_i S_{ij}$$

for the sequence-dependent case and

$$S_{*j} = S_j$$

for the sequence-independent case. This simplification will make our condition even more conservative for the case of sequence-dependent setups, especially when the changeover time S_{ij} depends strongly on i. However, we decide not to carry a more general analysis here because, as discussed in Chapter 6, in such cases it may be desirable to use a richer version of the HZP to begin with.[2]

Proceeding with our derivation of the satisfiability boundary, we express (3.11) as

$$(\boldsymbol{e}_j - \tau_j \boldsymbol{d})^{\mathrm{T}} \boldsymbol{\phi} > 0,$$

where $\boldsymbol{e}_j$ denotes the unit vector in the j-th direction. Now define a matrix $\boldsymbol{B}$ whose j-th row, $\boldsymbol{b}_j$, is given by

$$\boldsymbol{b}_j = \boldsymbol{e}_j^{\mathrm{T}} - \tau_j \boldsymbol{d}^{\mathrm{T}},$$

or, if $\mathbf{I}$ denotes the identity matrix,

$$\mathbf{B} = \mathbf{I} - \boldsymbol{\tau} \boldsymbol{d}^{\mathrm{T}}. \tag{3.13}$$

With this definition, (3.11) is written in matrix notation as $\mathbf{B}\boldsymbol{\phi} > \mathbf{0}$.

The inverse of matrix $\mathbf{B}$ is given by

$$\mathbf{B}^{-1} = \mathbf{I} + \frac{1}{1-\rho} \boldsymbol{\tau} \boldsymbol{d}^{\mathrm{T}}, \tag{3.14}$$

which can be verified by multiplying the above expression by (3.13). Now, letting $\mathbf{B}\boldsymbol{\phi} = \boldsymbol{\beta}$ for some positive vector $\boldsymbol{\beta}$, we express $\boldsymbol{\phi}$ as

$$\boldsymbol{\phi} = \mathbf{B}^{-1} \boldsymbol{\beta}. \tag{3.15}$$

Note that since the columns of $\mathbf{B}^{-1}$ have strictly positive components, the above expression implies that $\boldsymbol{\phi} > \mathbf{0}$ as required. Also, given that the actual magnitude of $\boldsymbol{\phi}$ is not impor-

[2] In particular, using sequence-dependent hedging points differences ΔZ_{ij} might make the HZP better suited to this problem. Each of these values could then be adjusted to satisfy the stability condition.

tant (only its direction), we will enforce an arbitrary constraint that ensures $\boldsymbol{\phi} \neq \mathbf{0}$ in the convenient form

$$\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d} = 1. \tag{3.16}$$

Now, from (3.15) and (3.14), it follows that

$$\begin{aligned}
\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d} &= (\mathbf{B}^{-1}\boldsymbol{\beta})^{\mathrm{T}}\boldsymbol{d} \\
&= \boldsymbol{\beta}^{\mathrm{T}}\left(\mathbf{I} + \frac{1}{1-\rho}\boldsymbol{d}\boldsymbol{\tau}^{\mathrm{T}}\right)\boldsymbol{d} \\
&= \frac{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{d}}{1-\rho}.
\end{aligned}$$

Substituting this result in (3.16), we obtain that

$$\frac{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{d}}{1-\rho} = 1. \tag{3.17}$$

Now consider the constraint given in (3.12). For any given $\epsilon > 0$, the smallest possible ΔZ_j that will still satisfy Theorem 3.2.3 is obtained by making this constraint active. Since ϵ is arbitrary, it then follows that at the boundary of satisfiability

$$\Delta Z_j = \frac{\boldsymbol{\phi}^{T}\boldsymbol{d} - \phi_j d_j}{\phi_j - \boldsymbol{\phi}^{T}\boldsymbol{d}\,\tau_j} S_{*j}, \tag{3.18}$$

for all j. Substituting (3.16), the equation reads

$$\begin{aligned}
\Delta Z_j &= \frac{1 - \phi_j d_j}{\phi_j - \tau_j} S_{*j} \\
&= \frac{1 - \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{e}_j d_j}{(\boldsymbol{\phi} - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{e}_j} S_{*j}.
\end{aligned}$$

We replace $\boldsymbol{\phi}$ by $\mathbf{B}^{-1}\boldsymbol{\beta}$ in the last equation, and use (3.14) to obtain

$$\begin{aligned} \Delta Z_j &= \frac{1 - \boldsymbol{\beta}^{\mathrm{T}}\mathbf{B}^{-T}\boldsymbol{e}_j d_j}{(\mathbf{B}^{-1}\boldsymbol{\beta} - \boldsymbol{\tau})^{\mathrm{T}}\boldsymbol{e}_j} S_{*j} \\ &= \frac{1 - \boldsymbol{\beta}^{\mathrm{T}}(\mathbf{I} + \frac{\boldsymbol{d}\boldsymbol{\tau}^{\mathrm{T}}}{1-\rho})\boldsymbol{e}_j d_j}{\left[\boldsymbol{\beta}^{\mathrm{T}}\left(\mathbf{I} + \frac{\boldsymbol{d}\boldsymbol{\tau}^{\mathrm{T}}}{1-\rho}\right) - \boldsymbol{\tau}^{\mathrm{T}}\right]\boldsymbol{e}_j} S_{*j}, \end{aligned}$$

which upon substitution of (3.17) simplifies to

$$\Delta Z_j = \frac{1 - (\beta_j + \tau_j)d_j}{\beta_j} S_{*j}. \tag{3.19}$$

Solving for β_j, we get

$$\beta_j = \frac{1 - \rho_j}{\Delta Z_j + d_j S_{*j}} S_{*j}, \tag{3.20}$$

which is positive for all j as required.

Finally, substituting (3.20) into (3.17), we obtain the expression for the satisfiability boundary of Condition 3.3.1, resumed in the box below. Notice how the apparently complex statement in the stability condition reduced to a simple-to-evaluate equation in terms of the hedging-zone parameters.

Satisfiability Boundary of Condition 3.3.1

$$\sum_{j=1}^{N} \frac{1 - \rho_j}{\Delta Z_j + d_j S_{*j}} S_{*j} d_j = 1 - \rho. \tag{3.21}$$

Comparison of Condition 3.3.1 with Dai and Jennings (2004)

Dai and Jennings (2004) derived conditions for the stochastic stability of queueing networks with setups. To compare their results with ours, we rearrange (3.21) into

$$\sum_{j=1}^{N}\left(\rho_j + \frac{S_{*j}d_j}{\frac{\Delta Z_j + S_{*j}d_j}{1-\rho_j}}\right) = 1$$

and denote by l_j the quantity $(\Delta Z_j + d_j S_{*j})/(1-\rho_j)$. For each type j, l_j corresponds to the total amount of material produced during a run that starts *exactly* at the boundary of the hedging zone, i.e., with $x_j = Z_j^{\mathrm{L}}$. Thus, l_j corresponds to the minimum amount of material produced on a run of j when the system is not in the hedging zone. We can amortize the changeover time during this run by dividing it over each unit of material that was produced, leading to an increased effective unit-production time of

$$\tilde{\tau}_j = \tau_j + \frac{S_{*j}}{l_j}. \tag{3.22}$$

With these definitions, our stability condition takes the attractive form

$$\tilde{\rho} = \sum_{j=1}^{N} \tilde{\tau}_j d_j < 1. \tag{3.23}$$

A policy that, among other reasonable properties, produces at least l_j units of material on each run of type j and satisfies (3.23) at each station in the network has been termed *sensible* by Dai and Jennings (2004), and they show that in many cases such policies lead to systems that are also stable in a stochastic sense. Surprisingly, our derivations leading to (3.21) confirm that Condition 3.3.1 is equivalent to the condition given in that reference, which came to our attention after this work was completed.

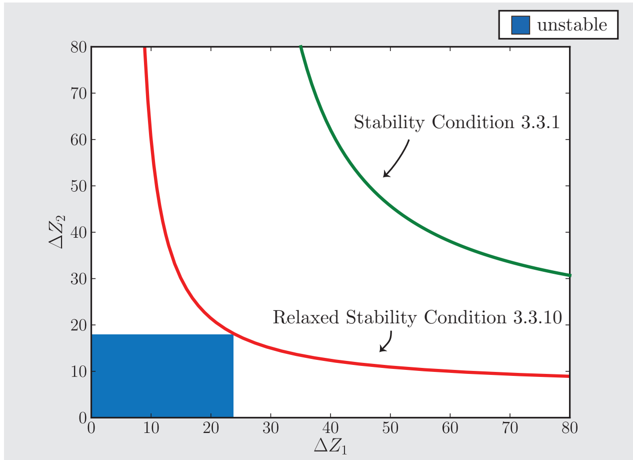
The equivalence between the two conditions is an interesting result that is not apparent from the derivations of Dai and Jennings. In particular, their developments were based on a slightly different fluid model that arises from the limit of a scaled, discrete-material stochastic process (see Chapter 5). Moreover, the authors proved that (3.23) is sufficient for the stability of their policies by starting with a specific Lyapunov function, while here we have started with a *class* of functions given by (3.4). In principle, one could expect that the extra flexibility afforded by considering a set of candidate Lyapunov functions as opposed to a single function would lead to a sufficient condition that covers a larger stable zone. However, we have shown that this is not the case. Such considerations are important since, as we discuss in the next section, the stability condition given by (3.21) or (3.23) tends to be too conservative for our policy (even in the sequence-independent case).

Conservativeness of Condition 3.3.1 (Sequence-Independent Case)

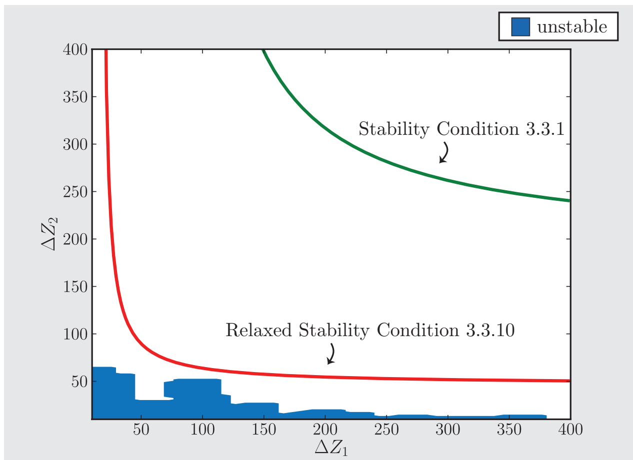
In this section, we focus exclusively on systems with sequence-independent setups, and thus the effect of sequence dependencies on the conservativeness of our stability condition is not addressed (see the comments at the beginning of Section 3.3.2). Starting from Theorem 3.2.3 and leading to the satisfiability boundary equation (3.21), we have made several simplifications in our derivations. These include the choice of linear and continuous Lyapunov functions given by (3.4), the selection of the set $\mathcal{B}$, and the requirement that the function decreases at the end of *every* run outside of this set. All of these simplifications add to the conservativeness of our stability condition, and thus it is natural to ask how close the satisfiability boundary is to the actual stability boundary of system Σ_π.

To partially answer this question, in Fig. 3-6 we compare the satisfiability boundary given by (3.21) with the experimental stability boundary described in Section 3.1 (the boundaries are given by the upper right hand side curves on each plot). As seen in this figure, Condition 3.3.1 seems to be too conservative, as a large set of stable choices of $\boldsymbol{\Delta Z}$ lies outside of the satisfiability boundary.

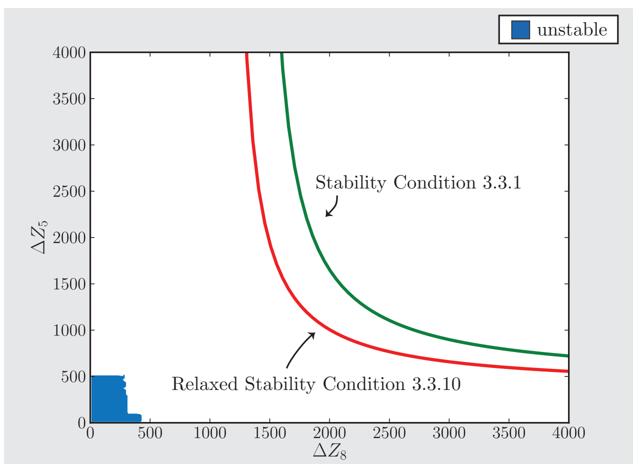
Of course, conservativeness may not be an important issue if the stable choices of $\boldsymbol{\Delta Z}$ not



(a) $N = 3$ system of Section 2.3.1.



(b) $N = 5$ system of Table D.1 in Section D.2.



(c) $N = 10$ system of Table D.2 in Section D.2

Figure 3-6: Comparison of the stability zones of Fig. 3-4 and the satisfiability boundaries of Conditions 3.3.1 and 3.3.10. (Sequence-independent setups.)

captured by the condition correspond to poor hedging zones values to begin with, in terms of some schedule cost measure. This consideration becomes more significant when we realize that relaxing the condition might come at the expense of increased complexity (for e.g., the computational costs of evaluating the condition could grow exponentially with the size of the system). On the other hand, even in such cases in which a well-performing value of $\boldsymbol{\Delta Z}$ satisfies our stability condition, for robustness purposes, it might still be desirable to have an idea of how far this vector is from the unstable zone. Without this knowledge, for example, one could worry that a choice of $\boldsymbol{\Delta Z}$ that is close to the satisfiability boundary could lead to instability in the presence of parameter uncertainty.

Fortunately, it turns out that it is possible to relax our stability condition—in some cases by a significant amount—without increasing the computational costs, giving us the confidence to select $\boldsymbol{\Delta Z}$ from a larger set of possible values without compromising stability. This relaxation will be addressed next, after taking a closer look at the dynamics of the HZP.

3.3.3 Complete-Production Property

In the introduction to this chapter, we presented an example of an unstable system operated under the HZP. There, we saw that the system became unstable by ignoring the lowest-priority part type indefinitely, leading to an ever-growing backlog for that item. Conceivably, a system could not ignore any of the part types and still be unstable. Such would be the case if the production runs of all part types were getting increasingly longer, so that even though no item is ignored, the system is never able to catch up with the demand.

It is useful to distinguish between these two types of unstable behaviors for the purposes of relaxing our stability condition. To this end, we define the concept of a complete-production system below, which states that for this type of system we can always obtain a bound on the time it takes to change over into any of the part types.

Definition 3.3.3 (Complete-Production System) *A system Σ_π is said to be a complete-production system if for any $\alpha > 0$, any $j \in \mathcal{Q}$, and any initial condition $\boldsymbol{x}(t_0)$ with $\|\boldsymbol{x}(t_0)\| \leq \alpha$, there exists a finite $\tilde{t}_j(\alpha)$ such that the system will have produced type j at least once during this period of time. That is,*

$$\sigma(t) = j, \textit{ for some } t \leq t_0 + \tilde{t}_j(\alpha).$$

Notice that the above definition implies that, as t grows, a complete-production system will change setups into each of the part types $j \in \mathcal{Q}$ infinitely many times. On the other hand, if there were trajectories that ignored some of the part types indefinitely, then it will not be possible to find a finite bound $\tilde{t}_j(\alpha)$ for all α, and thus Σ_π will not be a complete-production system.

The following terminology will be useful for proving the lemmas and theorem to follow.

Definition 3.3.4 (V-increasing(decreasing) run) *For a fixed Lyapunov function $V(\cdot)$, a production run that starts at some state $\boldsymbol{x} \in \partial\mathcal{X}$ is said to be V-increasing(decreasing) if the change in value of the function at the completion of the run is positive(negative). That is, if $V(\boldsymbol{\psi}(\boldsymbol{x})) - V(\boldsymbol{x})$ is greater(less) than 0.*

Definition 3.3.5 (V-neutral run) *A run that starts with some initial deviation $\boldsymbol{y}^V$, with $\boldsymbol{Z}^{\mathrm{U}} - \boldsymbol{y}^V \in \partial\mathcal{X}$, and that is neither V-increasing nor V-decreasing is called V-neutral.*

Setting $\Delta V(\boldsymbol{x})$ to 0 in (3.8), we see that the surplus deviation $\boldsymbol{y}^V$ corresponding to a V-neutral run of part type j with initial setup i is given by

$$y_j^V = \frac{\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d} - \phi_j d_j}{\phi_j - \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d}\tau_j} S_{ij}. \tag{3.24}$$

Furthermore, any run of type j that starts with $Z_j^U - x_j$ greater than y_j^V will be V-decreasing and in the opposite case will be V-increasing.

Based on these definitions and a closer analysis of the dynamics of the HZP, we state below a theorem that allows us to relax the sufficient condition of the previous section. This theorem says that any *HZP-controlled system with the complete-production property and sufficient capacity must be stable.* The proof is somewhat involved, and we will first state and prove some lemmas that will simplify our arguments.

Theorem 3.3.6 (Complete-Production Theorem) *Let Σ_π be a system operated under the HZP with $\rho < 1$. Then, a necessary and sufficient condition for the system to be stable is that Σ_π is a complete-production system.*

The intuition behind the previous theorem is conceptually straightforward. Suppose a system with $\rho < 1$ is unstable but it manages to produce all part types infinitely often. Because of its instability, we know that the surplus deviations must be unbounded and so there will be increasingly longer production runs. Furthermore, the values that any Lyapunov function takes must be unbounded. However, since long runs always lead to large reductions in the Lyapunov function, the only way $\{ V(\boldsymbol{x}_n) \; ; \; n \geq 0 \}$ can be unbounded is if the system is performing a growing number of short, V-increasing runs that offset the long V-decreasing runs. We will show that if the complete-production assumption property is satisfied in the HZP, the number of V-increasing runs that a system can make before it makes a V-decreasing run is bounded. Therefore, as illustrated on Fig. 3-7, although the Lyapunov function might not be decreasing after every *single* run, its net change must be negative after a finite number of runs.

The following supporting lemmas make our arguments more precise and lead to the statement

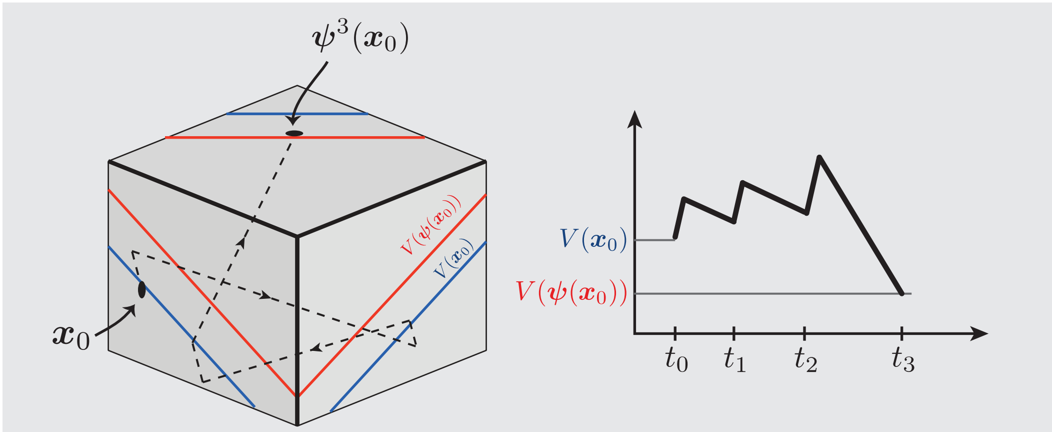


Figure 3-7: Depiction of the observation about the HZP that leads to the relaxation of Condition 3.3.1 (for $N = 3$). While $V(\boldsymbol{x})$ might not decrease after every run, we can show that in a complete-production system the net change in $V(\boldsymbol{x})$ is always negative after a finite number of runs.

and proof of the Complete-Production Theorem 3.3.6.

Lemma 3.3.7 *If a complete-production system is unstable, then the surplus deviations of all the part types will be unbounded (as opposed to only the deviations of a subset of the part types). That is,*

$$\limsup_{t\to\infty}(Z_j^U - x_j) \to \infty, \quad \textit{for all } j \in \mathcal{Q}.$$

Proof: *If the system is unstable, then by definition the surplus deviations of at least one of the part types, say type j, must be unbounded. Since by the complete-production assumption all part types are produced, this means that there will be increasingly longer runs of type j and, during these runs, the surpluses of all the other part types will drop. This leads to increasingly longer runs of those part types as well, and therefore unbounded surplus deviations for all $j \in \mathcal{Q}$.* □

Lemma 3.3.8 *Suppose Σ_π is an N-part-type, complete-production system operated under the HZP and with parts labelled so that $\mathsf{P}(1) > \mathsf{P}(2) > \cdots > \mathsf{P}(N)$. Let the vector $\boldsymbol{\phi}$ in $V(\cdot)$ satisfy $\mathbf{B}\boldsymbol{\phi} > \mathbf{0}$. Consider any state $\boldsymbol{x}_0$ with $\sigma_1 = N$, so that $\boldsymbol{x}_0$ corresponds to the* *surplus vector at the beginning of a run of the lowest priority type N. The minimum possible duration $t_1 - t_0$ of such a run increases linearly with $V(\boldsymbol{x}_0)$. Thus, when the value of the Lyapunov function is large, runs of the low-priority part are long.*

Proof: *In order for the HZP to have chosen the lowest-priority type, the surpluses of all other part types must have been within their hedging zones. Therefore, it follows that the value of the Lyapunov function at the beginning of the run must satisfy*

$$V(\boldsymbol{x}_0) \leq \sum_{i=1}^{N-1} \phi_i(Z_i^U - Z_i^L) + \phi_N(Z_N^U - x_N(t_0)).$$

This implies that the surplus deviation of type N at the beginning of the run is bounded from below by

$$Z_N^{\mathrm{U}} - x_N(t_0) \geq \frac{1}{\phi_N}\left(V(\boldsymbol{x}_0) - \sum_{i=1}^{N-1} \phi_i(Z_i^U - Z_i^L)\right).$$

Since the length of the run is proportional to the magnitude of the surplus deviation, we conclude that the larger the value of $V(\boldsymbol{x}_0)$, the longer it will take to complete the next run of type N. $\square$

The next lemma shows that when a system operated under the HZP has the complete-production property, the Lyapunov function cannot increase by very much before it starts decreasing.

Lemma 3.3.9 (Maximum Increase in V) *Suppose Σ_π is an N-part-type, complete production system operated under the HZP and with parts labelled so that $\mathsf{P}(1) > \mathsf{P}(2) > \cdots > \mathsf{P}(N)$. Let the vector $\boldsymbol{\phi}$ in $V(\cdot)$ satisfy $\mathbf{B}\boldsymbol{\phi} > \mathbf{0}$. Then, over any trajectory segment that involves production runs of no other than the first k part types, the increase in Lyapunov* *function is bounded by some finite number ΔV_k^+. That is, if*

$$\{\, \boldsymbol{x}_n \in \partial\mathcal{X} \;;\; n \in [n_1, n_2] \,\}$$

denotes some segment of an arbitrary trajectory with $\sigma_n \leq k$ for $n \in (n_1, n_2]$, then there exists a finite number ΔV_k^+ such that, for any such segment,

$$V(\boldsymbol{x}_{n_2}) - V(\boldsymbol{x}_{n_1}) \leq \Delta V_k^+. \tag{3.25}$$

Proof: *We prove this by induction. First, for some $\tilde{k} < N$ and finite numbers $A_1, A_2, \ldots, A_{\tilde{k}}$, let $\mathcal{P}(\tilde{k}; A_1, A_2, \ldots, A_{\tilde{k}})$ denote the optimization problem*

$$\sup_{\boldsymbol{x}_0, \sigma_0, n} \quad V(\boldsymbol{x}_n) - V(\boldsymbol{x}_0) \tag{3.26}$$

$$\textit{subject to} \quad \begin{array}{ll} \sigma_m \leq \tilde{k} & m = 1, 2, \ldots, n \\ 0 \leq Z_j^U - x_j(t_0) \leq A_j & j = 1, 2, \ldots, \tilde{k}, \\ \boldsymbol{x}_{m+1} = \boldsymbol{\psi}(\boldsymbol{x}_m) & m = 0, 1, \ldots, n-1 \\ \boldsymbol{x}_0 \in \partial\mathcal{X}. & \end{array}$$

This optimization problem finds, for a given Σ_π, the trajectory that leads to the maximum change in the Lyapunov function, over all trajectories that do not produce any part type with label greater than $\tilde{k}$ (as specified on the first set of constraints) and whose initial surpluses for the first $\tilde{k}$ part types are within A_j of their base stock levels (as specified on the second set of constraints). Only trajectories that are generated by policy π are considered (as enforced by the third set of constraints).

We note two important properties about problem $\mathcal{P}(\tilde{k}; A_1, A_2, \ldots, A_{\tilde{k}})$. First, assuming that $\boldsymbol{x}_0 \notin \mathcal{Z}_N$, the surplus levels of part types $\tilde{k}+1$ through N do play any role in the problem.

This follows because all feasible solutions to the problem involve only part types with label smaller or equal than $\tilde{k}$, and the decision to produce some type $i > \tilde{k}$ is independent of $x_{\tilde{k}+1}, \ldots, x_N$ in the HZP (outside of the hedging zone). The other property to note is that the solution to the problem is always finite. This follows because the set of initial conditions for the surplus of types 1 through $\tilde{k}$ is bounded by the A_j's coefficients and, since the system satisfies the complete-production assumption, there is a bound on the length of any feasible trajectory of the problem (i.e., a feasible trajectory cannot prolong indefinitely because it will eventually involve a run of some type $i > \tilde{k}$).

Now suppose that it is true that for some $\tilde{k}$ the bound in (3.25), $\Delta V_{\tilde{k}}^{+}$, can be obtained by solving $\mathcal{P}(\tilde{k}; A_1, A_2, \ldots, A_{\tilde{k}})$ for some known numbers $A_1, A_2, \ldots, A_{\tilde{k}}$. Then, we show next how we can use this fact to obtain the bound $\Delta V_{\tilde{k}+1}^{+}$. To this end, note that since large runs lead to large reductions in $V(\cdot)$, there exists a finite $A_{\tilde{k}+1}$ (which can be found from (3.8)) such that whenever $Z_{\tilde{k}+1}^{U} - x_{\tilde{k}+1} \geq A_{\tilde{k}+1}$ and type $\tilde{k}+1$ gets produced, the change in Lyapunov function will be less (i.e., more negative) than $-\Delta V_{\tilde{k}}^{+}$ (see Fig. 3-8). This means that a trajectory segment involving parts with label no greater than $\tilde{k}+1$ and that achieves the largest possible increment in Lyapunov function should have $Z_{\tilde{k}+1}^{U} - x_{\tilde{k}+1}(t) \leq A_{\tilde{k}+1}$ at all times t in which a run of type $\tilde{k}+1$ begins. Otherwise, the runs of type $\tilde{k}+1$ in the segment would offset any possible increments in $V(\cdot)$ from the previous runs and the trajectory would not achieve the greatest possible change in Lyapunov function. We conclude then that $\Delta V_{\tilde{k}+1}^{+}$ can be found from the solution to the problem $\mathcal{P}(\tilde{k}+1; A_1, A_2, \ldots, A_{\tilde{k}+1})$. This completes the inductive step of the proof.

We now show that we can find ΔV_2^{+} from the solution to $\mathcal{P}(2; A_1, A_2)$ for some known numbers A_1 and A_2 (this is the base case of our proof). Note that any trajectory that involves

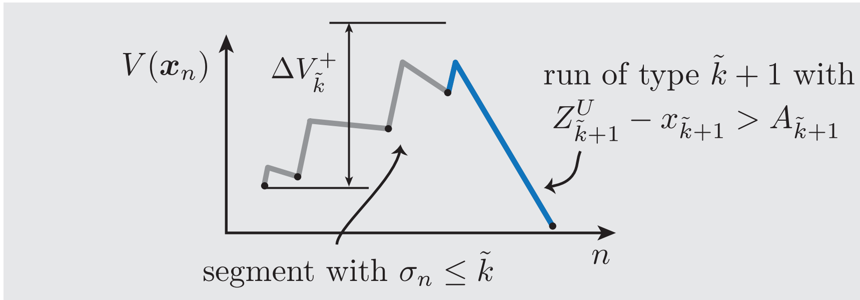


Figure 3-8: Depiction of the argument used to prove Lemma 3.3.9. Over any segment in which $\sigma_n \leq \tilde{k}$, the increase in Lyapunov function cannot be greater than $\Delta V_{\tilde{k}}^{+}$. Thus, if part type $\tilde{k}+1$ is produced next and its surplus deviation is greater than $A_{\tilde{k}+1}$, the decrease in $V(\boldsymbol{x})$ at the end of the run will offset any possible increments from the previous runs.

just part types 1 *and* 2 *will start having* V*-increasing runs only after either*

$$\begin{aligned} Z_1^U - x_1 <& y_1^V \quad \textit{or} \\ Z_2^U - x_2 <& y_2^V, \end{aligned}$$

where y_j^V *is the deviation of a* V*-neutral run, given by* (3.24)*. Thus, solving* $\mathcal{P}(2; A_1, A_2)$ *with* $A_1 = y_1^V$ *and* $A_2 = y_2^V$ *will give us the base case bound* ΔV_2^+ *that we seek.* □

Armed with the previous lemmas, we can now prove the Complete-Production Theorem 3.3.6.

Proof of Theorem 3.3.6: *That the condition is necessary follows directly from the definition of stability since, if the system were not a complete-production system, there would exist at least one trajectory that ignores one or more of the part types indefinitely. This would lead to an unbounded backlog for those items.*

Now, to prove sufficiency, note that since part type N *has the lowest priority, the system will generally perform several production runs of types* 1 *through* $N-1$ *before producing type* N*. Let then* $\{\, \boldsymbol{x}_n \in \partial\mathcal{X} \;;\; n \in (n_1, n_2] \,\}$ *be any segment of a sequence in which part type* N *is not* *produced and with $\sigma_{n_2+1} = N$. By Lemma 3.3.9, we know that $V(\boldsymbol{x}_{n_2}) - V(\boldsymbol{x}_{n_1}) \leq \Delta V_{N-1}^{+}$. Thus,*

$$V(\boldsymbol{x}_{n_2+1}) - V(\boldsymbol{x}_{n_1}) \leq \Delta V_{N-1}^{+} + (V(\boldsymbol{x}_{n_2+1}) - V(\boldsymbol{x}_{n_2}))$$

Furthermore, by Lemma 3.3.8, if $V(\boldsymbol{x}_{n_2})$ is sufficiently large, run n_2+1 involving type N will be long and thus it will reduce V by an amount greater than ΔV_{N-1}^{+}. Thus, for a large-enough $V(\boldsymbol{x}_{n_1})$, the expression above will always be negative.

The previous argument implies that there must exist some finite V^* such that, whenever $V(\boldsymbol{x}) \geq V^*$, the net change in $V(\cdot)$ over a segment that includes a large-enough number of runs will always be negative. Furthermore, since $V(\boldsymbol{x})$ cannot increase by more than ΔV_{N-1}^{+} over any such segment, we see that

$$V(\boldsymbol{x}(t)) \leq V^* + \Delta V_{N-1}^{+} + \boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{d} \max_{i,j} S_i j$$

for all t greater than some $\tilde{t}(\boldsymbol{x}(t_0))$. It thus follows that the system is stable. □

We conclude this section by noting that not all complete-production CC policies are necessarily stable. For example, consider a policy that generates a sequence of clearing runs of the form 12-3-1212-3-121212-3-.... While in this scenario the system could still satisfy the complete-production property (assuming $N = 3$), its surplus is unbounded because the time between successive runs of part type 3 is increasing. For this reason, our relaxation of Condition 3.3.1 has relied on the specific structure of the HZP. However, the methodology presented here should be useful for assessing the stability of other policies that one might conceive.

3.3.4 Relaxed Stability Condition

With part types labelled so that $\mathsf{P}(1) > \mathsf{P}(2) > \cdots > \mathsf{P}(N)$, we know that in order to produce part type N in the HZP, the surpluses of parts 1 through $N-1$ must be all within their hedging zones at some point. That is, $\sigma_n = N$ implies that $\boldsymbol{x}_{n-1} = \mathcal{Z}_{N-1}$, where

$$\mathcal{Z}_{N-1} = \left\{ \boldsymbol{x} \in \partial\mathcal{X} \;\middle|\; x_i \in [Z_i^{\mathrm{L}}, Z_i^{\mathrm{U}}], \quad i = 1, 2, \ldots, N-1 \right\}.$$

Thus, producing the lowest priority part becomes a *reachability problem*: as long as all trajectories are able to reach the set $\mathcal{Z}_{N-1}$, the system will produce all part types.

Theorem 3.2.3 and Corollary 3.2.4 already provide us with the tools that we need to address this reachability problem. Recall that if the Lyapunov function is known to decrease by at least some amount ϵ on every run outside of $\mathcal{B}$, then all trajectories reach $\mathcal{B}$ in finite time. This implies that we can ignore part type N and design the hedging zone of the first $N-1$ part types so that it satisfies Condition 3.3.1, and this will guarantee that the lowest-priority part type gets produced. Furthermore, using the Complete-Production Theorem, such a selection of the hedging zone also ensures that the system is stable. Thus, we have the following Relaxed Stability Condition.

Condition 3.3.10 (Relaxed Stability Condition) *Suppose Σ_π is an N-part type system operated under the HZP and with parts labelled so that $\mathsf{P}(1) > \mathsf{P}(2) > \cdots > \mathsf{P}(N)$ and $\rho < 1$. Let $\Sigma_\pi^\bullet$ denote the same system but with only the first $N-1$ part types. Then, Σ_π is stable if $\Sigma_\pi^\bullet$ satisfies Condition 3.3.1.*

Proof: *Denote by $\boldsymbol{x}^\bullet(t)$ the surplus state of $\Sigma_\pi^\bullet$. By the Reachability Corollary 3.2.4 and our definition of $\mathcal{B}$ that lead to Stability Condition 3.3.1, it follows that for any $\|\boldsymbol{x}^\bullet(t_0)\| \leq \alpha$, the system will reach $\mathcal{Z}_{N-1}$ at least once before some time $t_0+\tilde{t}(\alpha)$. Since whenever the system* *reaches $\mathcal{Z}_{N-1}$ part type N gets produced, it follows that Σ_π is a complete-production system. By the Complete-Production Theorem 3.3.6, this implies that Σ_π is stable.* $\square$

Conservativeness of the Relaxed Condition 3.3.10

Since the relaxation of our stability condition involves considering a system that lacks the lowest-priority part type, it follows that the amount of relaxation that we can achieve depends on the parameters of part type N. If ρ_N is large compared to the other utilizations, the relaxed boundary will give a significant larger set of feasible $\boldsymbol{\Delta Z}$ values.

The satisfiability boundaries of the relaxed condition for our previous experiments are overlaid in Fig. 3-6, which correspond to the bottom curves in upper right corner of each plot. With part type N being the lowest priority part, the ratios ρ_N/ρ for each of the simulated systems are summarized in Table 3.1. Notice that, as expected from the table, the example system with $N = 10$ showed the least amount of relaxation (i.e., the relaxed condition did not cover a much larger stability zone), while for the other two cases the amount of relaxation was very significant. The results also suggest that Relaxed Condition 3.3.10 becomes more conservative as N grows.

Table 3.1: Ratio of the utilization of the lowest-priority part type to the total utilization, for each of the systems in Fig. 3-6.

Case	ρ_N/ρ
$N = 3$	0.14
$N = 5$	0.18
$N = 10$	0.015

3.4 Other Stability Results

3.4.1 Stability of CLB, PKP, and LOP

The stability of policies CLB, PKP, and LOP has been already established in the literature (Perkins and Kumar 1989, Lan and Olsen 2006). However, for completeness, we show here how Theorem 3.2.3 can be applied to obtain a straightforward proof of their stability.

First of all, notice from the definitions of these policies (refer to Chapter 2) that, at least when $\boldsymbol{Z}^{\mathrm{U}} - \boldsymbol{x}$ is large, the next changeover j^* is always determined according to a rule of the form

$$j^* = \underset{j \in \mathcal{Q}}{\operatorname{argmax}} \left\{ \frac{Z_j^U - x_j + C_j}{D_j} \right\}, \tag{3.27}$$

where $C_j \geq 0$ and $D_j > 0$ for $j \in \mathcal{Q}$. Thus, we can group these policies into what we refer to as the clear-the-largest-weighted-deviation (CLD) type, defined below.

Definition 3.4.1 (Clear-the-Largest-Weighted-Deviation (CLD) Policy) *A policy is of the CLD type if it belongs to the CC Class and, for sufficiently-large vectors $\boldsymbol{Z}^{\mathrm{U}} - \boldsymbol{x}$, the changeover decision is given by* (3.27)*, for some fixed $C_j \geq 0$ and $D_j > 0$, $j \in \mathcal{Q}$.*

The next result shows that the policies in Definition 3.4.1 are always stable. The argument is also explained graphically for a three-part-type system in Fig. 3-9.

Theorem 3.4.2 (Stability of CLD Policies) *Clear-the-largest-weighted-deviation (CLD) policies are always stable, as long as $\rho < 1$.*

Proof: *Consider the hyperplane $x_i = Z_i^U$ for any $i \in \mathcal{Q}$ and the parametric curve $\boldsymbol{g}^i(s)$ in $\mathbb{R}^N$ given by*

$$g_k^i(s) = \begin{cases} Z_k^U - D_k s + C_k & \text{if } k \neq i, \\ Z_i^U & \text{if } k = i, \end{cases}$$

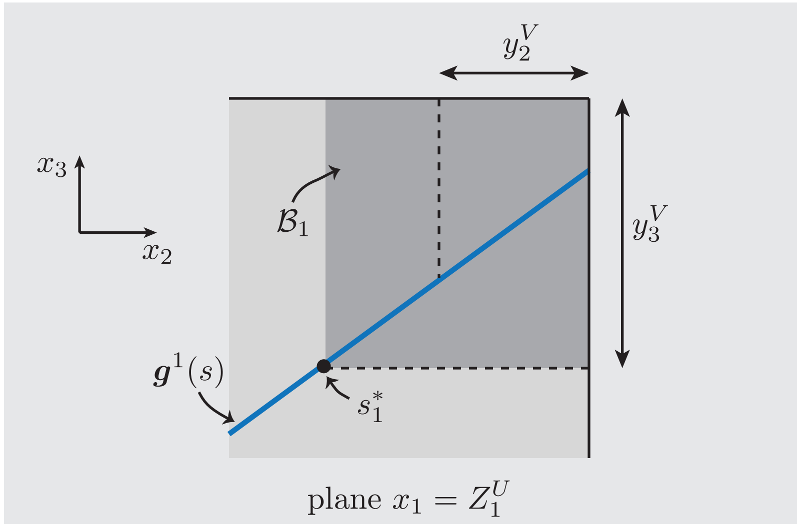


Figure 3-9: Depiction for $N = 3$ of the proof that CLD policies are always stable when $\rho < 1$. At any plane $x_i = Z_i^U$, there exists a point along the boundary separating the setup zones that defines a set $\mathcal{B}_i$ satisfying Stability Theorem 3.2.3 requirements. (Refer to the proof of Theorem 3.4.2 for an explanation of the notation used in the figure.)

for $s \geq 0$. *Curve* $\boldsymbol{g}^i(s)$ *represents the set of points on the hyperplane* $x_i = Z_i^U$ *for which decision rule* (3.27) *becomes ambiguous for sufficiently large* s *(i.e., all ratios in that equation with* $j \neq i$ *have the same value* s*).*

Now, fix some Lyapunov function with $\mathbf{B}\boldsymbol{\phi} > \mathbf{0}$. *Since* $D_j > 0$ *for all* j, *there must exist an* s_i^* *large enough so that*

$$Z_k^{\mathrm{U}} - D_k s_i^* + C_k > Z_k^{\mathrm{U}} - y_k^V \quad \forall\, k \neq i.$$

Thus, any time that the system is at plane $x_i = Z_i^U$ *and outside of the set defined by*

$$\mathcal{B}_i = \left[Z_1^U - D_1 s_i^* + C_1,\ Z_1^U\right] \times \cdots \times \left\{Z_i^U\right\} \times \cdots \times \left[Z_N^U - D_N s_i^* + C_N,\ Z_N^U\right],$$

runs will be V*-decreasing. Setting* $\mathcal{B} = \cup_{i=1}^N \mathcal{B}_i$, *we see that this set satisfies the requirement of Theorem 3.2.3 and that, therefore, the system is stable.* □

3.4.2 Stability of the HZP with General Priorities

Going back to Stability Condition 3.3.1, we can see that the priorities or rank ordering did not play any role in the derivations of that condition. The main requirement was that whatever part we selected for a new run had a large-enough surplus deviation such that its production run would be V-decreasing.

On the other hand, the arguments leading to Relaxed Stability Condition 3.3.10 relied on the fact that part type N had the lowest priority and was produced infinitely often. We now state this relaxed condition for systems in which priorities are not necessarily unique. As described in our definition of the HZP (see Chapter 2), when priorities are non-unique the policy employs a clear-the-largest-weighted-deviation (CLD) rule to resolve ties between potential changeovers. Given that this rule is always stable (as shown in the previous section), the next condition should not come as a surprise.

Condition 3.4.3 (HZP Relaxed Stability. General Priorities)

Suppose Σ_π is an N-part type system operated under the HZP with $\rho < 1$ and let $\mathcal{L}$ denote the set of part types with the lowest priority. That is

$$\mathcal{L} = \{\, j \in \mathcal{Q} \;\mid\; \mathsf{P}(j) \leq \mathsf{P}(k) \;\forall\; k \in \mathcal{Q} \,\}\,.$$

Denote by $\Sigma_\pi^\bullet$ a reduced system which includes only those part types in $\mathcal{Q}$ that are not in $\mathcal{L}$. Then, Σ_π is stable if $\Sigma_\pi^\bullet$ satisfies Condition 3.3.1.

Proof: *First note that, if $\mathcal{L} = \mathcal{Q}$, then all part types have the same priority and the HZP reduces to a pure CLD policy, which is always stable. Now, for the case in which $|\mathcal{L}| < N$, the policy consists of a mixture of priority-based changeover decisions and CLD-based decisions. Let $\boldsymbol{x}^\bullet$ denote the surplus of the reduced system $\Sigma_\pi^\bullet$. Condition 3.3.1 ensures that the surplus $\boldsymbol{x}^\bullet$ of $\Sigma_\pi^\bullet$ will reach its corresponding hedging zone, and thus that Σ_π will face the decision* *to select a part type in $\mathcal{L}$ infinitely often. Furthermore, using the same arguments as in the proof of the Maximum Increase in V Lemma 3.3.9, we can see that the Lyapunov function cannot increase by more than some number, say $\Delta V_{\mathcal{L}}^{+}$, over any trajectory segment in which no part in $\mathcal{L}$ gets produced. Finally, parts in $\mathcal{L}$ are always selected based on the CLD rule, and by the proof of Theorem 3.4.2 we know that this rule always leads to V-decreasing runs when $V(\boldsymbol{x})$ is sufficiently large. Thus, the stability of Σ_{π} follows from the same argument used to prove the Complete-Production Theorem 3.3.6.* □

3.5 Summary

In this chapter we have used the concept of Lyapunov functions to formally prove the stability of the HZP. We began by showing that, for small hedging zones, the system may settle into sequences that ignore indefinitely some of the low-priority part types, and we derived a necessary and sufficient condition that prevents this from happening for an $N = 3$ system. We then showed through numerical experiments that the development of a necessary and sufficient condition for larger systems is an extremely complex problem, which motivated the search for a not-too-conservative sufficient condition.

Using a simple class of linear and continuous Lyapunov functions, we developed such a condition for a system operated under the HZP and showed that the condition is very easy to evaluate. We then proved that the only way a system with $\rho < 1$ can become unstable in the HZP is by ignoring some of the part types, a fact that allowed us to relax our original condition by applying it to a system with only the $N-1$ highest-priority part types. Finally, using our Lyapunov methods, we verified the stability of the other CC Class policies discussed in Chapter 2 as well as that of the HZP with non-unique priorities.

While we will revisit the issue of stability when we develop our stochastic model in Chapter 5, our main focus now shifts towards the issue of performance. In this case, the question we seek to address is how to select values of $\boldsymbol{\Delta Z}$ that not only guarantee stable production but that also lead to good performance in terms of some relevant cost measure.

Chapter 4

Deterministic Performance Analysis

In this chapter, we complete the specification of the scheduling policies using cost-motivated arguments, and we compare their performance through numerical simulation. While our results and analyses are still purely deterministic, they will set the stage for the next chapter, where we incorporate random failures into our model.

We begin by defining and motivating the cost metrics that will be used for assessing the quality and robustness of the heuristics. We then derive equations for computing lower bounds on these metrics and later discuss how the HZP and other CC Class policies take advantage of these bounds for selecting good parameter values. We also describe the classic Economic Lot Scheduling Problem (ELSP) and the problem of optimizing a production schedule over a fixed sequence of changeovers. These concepts will lead to the statement of Gallego's Recovery Policy, the non-CC Class policy that we outlined in Chapter 2 and that will form part of our performance comparisons. The main thesis contribution of this chapter consists of an extensive series of numerical experiments evaluating the deterministic performance of the policies under a variety of system parameters. We present and interpret the results of these experiments in the last section.

4.1 Schedule Costs

We will study the performance of our policies in terms of three cost measures that capture some of the tradeoffs between inventory, capacity, and setup costs that were discussed in Chapter 1. These measures are: the (1) long-term average surplus-deviation cost, (2) long-term average inventory and backlog cost, and (3) recovery costs, all of which also include average setup costs.

4.1.1 Average Surplus Deviation Cost J

Recall that, for a CC Policy, the surplus deviation of type i is defined as

$$y_i(t) = Z_i^{\mathrm{U}} - x_i(t),$$

which is positive for all t, and let $\boldsymbol{y}(t) = (\, y_1(t), y_2(t), \ldots, y_N(t)\,)^{\mathrm{T}}$. The *instantaneous* deviation cost measures how far the system's surplus is from its base stock level $\boldsymbol{Z}^{\mathrm{U}}$ at time t by taking a weighted sum of all deviations $y_i(t)$. The long-term average of this instantaneous cost is what we will call cost measure J, as defined below.

Definition 4.1.1 (Average Surplus Deviation Cost J) *Let $\mathbf{c} > \mathbf{0}$ be an $N \times 1$ vector and $Q_{ij}(t)$ denote the number of completed runs up to time t that began with an i-to-j changeover. For any initial condition $(\boldsymbol{y}_0, \sigma_0)$, define*

$$J(\boldsymbol{y}_0, \sigma_0) = \limsup_{T\to\infty} \frac{1}{T} \left(\int_0^T \boldsymbol{c}^{\mathrm{T}} \boldsymbol{y}(t)\,\mathrm{d}t + \sum_{i,j\in\mathcal{Q}} Q_{ij}(T) K_{ij} \right).$$

Then, the long-term surplus deviation cost J is defined as

$$J = \sup_{\boldsymbol{y}_0, \sigma_0} J(\boldsymbol{y}_0, \sigma_0).$$

In Definition 4.1.1, the components c_i of $\boldsymbol{c}$ correspond to the cost per unit deviation of type i per time, and the product $Q_{ij}(T)K_{ij}$ corresponds to the i-to-j setup costs incurred up to time T. Note that $J(\boldsymbol{y}_0, \sigma_0)$ depends in general on the initial conditions, while J does not. We make this distinction explicit because in some cases the system may have more than one limit cycle (or, more generally, more than one *attractor*), implying that the long-term average costs will vary depending on the initial conditions. An example of this behavior will be given in Section 4.6.

In many settings, measure J can be directly related to costs of economic importance to a factory or organization. For example, when the system operates in a pure make-to-order fashion (i.e., $\boldsymbol{Z}^{\mathrm{U}} = \mathbf{0}$), the surplus deviations are equal to the amount of backlogged orders and J measures long-term average backlogging costs. Alternatively, when surplus deviations correspond to the number of production tokens that have been released into the factory floor, J measures average work-in-process (WIP) costs. As pointed out by Tang (2005), in some firms, WIP costs may be more important than finished-goods holding costs (e.g., if the finished goods are shipped out immediately from the factory), and thus a factory manager may want to schedule production with cost J in mind. In the polling model of Fig. 2-6, one of the most important performance measures considered is the mean wait time of a job that arrives into the system (Takagi 1988). This quantity is related to the average number of jobs in the system through Little's Law (Little 1961). When $c_i = 1$ and $K_{ij} = 0$ for all i, j, measure J corresponds to the sum of the long-term average number of jobs in the system (in our case, modeled as a continuous quantity).

Even in cases where optimizing finished-goods costs is more important than WIP costs, we claim that J can still be considered as a surrogate measure of schedule quality. Roughly speaking, systems with low J-costs tend to operate with surplus levels that are close to their base stock levels, which in turn implies that base stocks levels do not have to be set very high to obtain some specified service level (fraction of time with inventory in the system). This idea is illustrated in Fig. 4-1, where we depict two possible surplus trajectories of type i. The trajectory on the left has a larger J cost and also requires a larger base stock Z_i^{U}—and thus more inventory on average—to provide the same level of service as the trajectory on the right. These heuristic statements about J motivate our procedure for selecting the parameters in the HZP, which is described in Section 4.4.

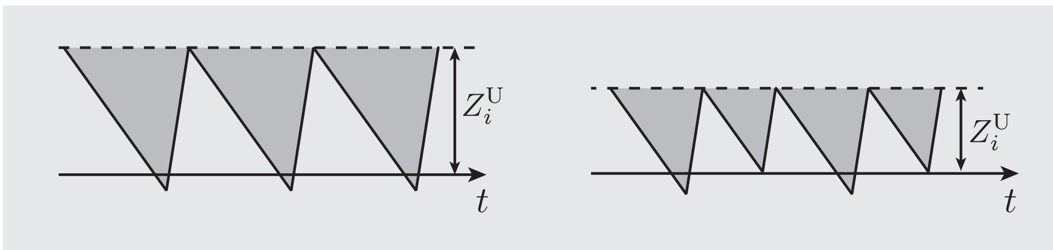


Figure 4-1: Example of two surplus trajectories of part type i. The trajectory on the left has a higher J cost than the trajectory on the right. Also, the trajectory on the left requires on average more inventory to provide the same service level (fraction of time with $x_i(t) > 0$).

To aid in our later discussions, we formally define the problem of scheduling a system with cost J as an objective. Note that this cost and its related optimization problem make sense mainly for CC Class policies, where there is a fixed base stock level $\boldsymbol{Z}^{\mathrm{U}}$ that we want to track closely, or for make-to-order systems (where $\boldsymbol{Z}^{\mathrm{U}} = \mathbf{0}$).

Definition 4.1.2 (J Optimization Problem (J-OP)) *For a given system Σ, the J-OP consists of finding a policy π that minimizes J over all valid policies satisfying the production rate constraint* (2.5).

Depending on the context, when talking about a feasible solution or a *solution* to J-OP, we will be referring to either a valid policy or the trajectory generated by that policy, and the J-costs associated with it. The *optimal solution* to J-OP is a feasible solution that optimizes J. Given that the closely-related ELSP problem (discussed in Section 4.3.1) is NP-hard (Hsu

1983), we do not expect to be able to find this optimal solution efficiently, except for very special cases.

4.1.2 Average Inventory and Backlog Cost I

Since cost J penalizes upstream inventory (assuming that, as mentioned, surplus deviations are immediately transformed into production orders and released into the factory floor), for optimizing this cost the system needs to be able to clear out WIP efficiently. In many settings, it is more important to penalize downstream inventory, which corresponds to the costs of stored finished goods.[1] In such cases, it is typical to balance the tradeoff between inventory and backlog; that is, while holding a large amount of inventory reduces the chance of stockouts or of machine starvation downstream in the production line (Gershwin 2002), too much inventory comes at the price of large holding costs and other expenditures required to store and manage it efficiently.

In this thesis, we will model these effects through a typically used measure that consists of the weighted sum of linear inventory and backlog costs. This cost, denoted as I, is defined below.

Definition 4.1.3 (Average Inventory and Backlog Cost I) *Let $\boldsymbol{h}, \boldsymbol{b} > \mathbf{0}$ be $N \times 1$ vectors and $Q_{ij}(t)$ denote the number of completed runs up to time t that began with an i-to-j changeover. Set $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ equal to the positive and negative parts of $\boldsymbol{x}$, respectively. That is, the i-th components of these vectors are $x_i^+ = \max(x_i, 0)$ and $x_i^- = \max(-x_i, 0)$.*

For any initial condition, define

$$I(\boldsymbol{x}_0, \sigma_0) = \limsup_{T \to \infty} \frac{1}{T} \left(\int_0^T \left(\boldsymbol{h}^{\mathrm{T}} \boldsymbol{x}^+(t) + \boldsymbol{b}^{\mathrm{T}} \boldsymbol{x}^-(t) \right) \mathrm{d}t + \sum_{i,j \in \mathcal{Q}} Q_{ij}(T) K_{ij} \right).$$

[1] The word "finished" here is relative, since it is from the point of view of the machine or station that we are scheduling. These goods could in fact be feeding some other downstream process.

Then, the long-term average inventory and backlog cost is given by

$$I = \sup_{\boldsymbol{x}_0, \sigma_0} I(\boldsymbol{x}_0, \sigma_0).$$

In the definition above, for each item i, the component h_i represents the holding costs per unit of inventory per time, and b_i represents the cost per unit of backlog per time. As with the previous cost measure, I is independent of the initial conditions.

We also define formally the problem of finding an optimal schedule with cost I as an objective.

Definition 4.1.4 (I Optimization Problem (I-OP)) *For a given system Σ, the I-OP consists of finding a policy π that minimizes I over all feasible policies satisfying the production rate constraint* (2.5)*.*

Relationship Between the Service Level and Backlog Costs

The service level θ_i is formally defined as the long-term fraction of time that there is inventory of item i in the system. That is,

$$\theta_i = \lim_{T \to \infty} \frac{1}{T} \int_0^T [\![x_i(t) > 0]\!] \, \mathrm{d}t, \tag{4.1}$$

where $[\![\cdot]\!]$ is equal to 1 if the condition inside the double brackets is true and zero otherwise. (Even though in the above definition we only consider those times when $x_i(t)$ is strictly greater than 0, in our continuous-material model a system operating with $x_i(t) = 0$ could still satisfy customers' orders immediately, as long as it produces exactly at the demand rate. This consideration is only important when the policy in question generates surplus trajectories that cruise at zero.)

It is shown in Sections B.1 and B.2 of Appendix B, that the long-term trajectory of $x_i(t)$

can always be perturbed into a lower-cost trajectory where the service rate satisfies

$$\theta_i^* = \frac{b_i}{h_i + b_i}\left(1 - \lim_{T\to\infty}\frac{1}{T}\int_0^T [\![x_i(t) = 0]\!]\,\mathrm{d}t\right). \tag{4.2}$$

In this expression, the integral corresponds to the long-term fraction of time that the trajectory is cruising at the zero-surplus level. For trajectories that are nowhere flat, the expression reduces to that derived by Gallego (1990),

$$\theta_i^* = \frac{b_i}{h_i + b_i}, \tag{4.3}$$

and the perturbation needed to achieve this service level consists of merely shifting up or down the trajectory (i.e., adjusting the base stock levels).

4.1.3 Recovery Costs

Costs J and I are steady-state metrics, in the sense that they measure the long-term behavior of the system and are independent of the initial conditions. However, these metrics do not give us any indication on the robustness of the policies to disruptions (in fact, the costs do not even distinguish between closed and open-loop schedules). While a policy that generates low-cost, steady-state schedules is clearly desirable, we would also like it to be able to recover efficiently from disruptions. This is the motivation behind the definitions of the *recovery costs* $C_J(\boldsymbol{y}_0, \sigma_0)$ and $C_I(\boldsymbol{x}_0, \sigma_0)$, stated below (also called *excess costs* by Gallego 1990).

Definition 4.1.5 (Recovery Costs $C_J(\boldsymbol{y}_0, \sigma_0)$ and $C_I(\boldsymbol{x}_0, \sigma_0)$) *For a given system Σ and initial state $(\boldsymbol{y}_0, \sigma_0)$ or $(\boldsymbol{x}_0, \sigma_0)$, the recovery costs are defined as*

$$C_J(\boldsymbol{y}_0, \sigma_0) = \limsup_{T\to\infty}\left(\int_0^T \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}(t)\,\mathrm{d}t + \sum_{i,j\in\mathcal{Q}} Q_{ij}(T)K_{ij} - J(\boldsymbol{y}_0, \sigma_0)T\right).$$

and

$$C_I(\boldsymbol{x}_0, \sigma_0) = \limsup_{T\to\infty} \left(\int_0^T \big(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}^+(t) + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}^-(t)\big)\,\mathrm{d}t + \sum_{i,j\in\mathcal{Q}} Q_{ij}(T)K_{ij} - I(\boldsymbol{x}_0, \sigma_0)T \right).$$

Consider the expression for $C_J(\boldsymbol{y}_0, \sigma_0)$. The integral in the first term measures the surplus-deviation cost accumulated since time 0 and up to time T and, as before, $Q_{ij}(T)K_{ij}$ corresponds to the total i-to-j setup costs at that time. The third term in the expression corresponds to the cost that would have accumulated during this period if the system had been following its long-term or steady state trajectory. Thus, $C_J(\boldsymbol{y}_0, \sigma_0)$ represents the extra cost that the system incurs for not starting in steady state. A similar explanation applies to $C_I(\boldsymbol{x}_0, \sigma_0)$.

4.2 Lower Bounds

Several lower bounds for variations of J-OP and I-OP have been developed in the literature, and researchers have leveraged these bounds for designing reasonable scheduling heuristics. The main approach for obtaining such bounds has consisted of relaxing the requirement that no two part types may be produced at the same time. For example, in the ELSP literature, a simple lower bound is obtained by computing the *economic manufacturing quantity* EMQ for each item, giving the so called Independent Solution (see Elmaghraby 1978). A better bound can be usually obtained by incorporating the long-term capacity constraint (4.9), as shown by Dobson (1987), who relies in this bound to compute target production frequencies for the items. Similarly, in the area of closed-loop scheduling, Perkins and Kumar (1989) developed a lower bound for J-OP by ignoring interferences between the production times of different items. This bound was later refined by Chase and Ramadge (1992), who incorporated the possibility of cruising, and by Lan and Olsen (2006), who considered the case of sequence-independent setup times and costs. (Readers interested in these bounds may also want to consult the work by Bertsimas and Nino-Mora 1999 and Adelman and Barz 2009.)

We will discuss in Section 4.4 how the lower bound on J forms the basis for policies PKP and LOP, and we will also motivate the bound's use in the selection of the parameters in the HZP. Thus, due to the important role that these bounds play on the specification and understanding of the scheduling policies, we present their derivation in detail next.

4.2.1 Long-term Properties

In order to obtain lower bounds on the steady-state costs J and I, we first develop a series of relations that describe the long-term behavior of our system. Suppose that a policy restricts its production rate $u_i(t)$ to the discrete set $\{0, d_i, \mu_i\}$ (this is justified in Section 4.2.2). Then, the cumulative production of type i at time t must satisfy

$$P_i(t) = T_i^{\mathrm{s}}(t)\mu_i + T_i^{\mathrm{c}}(t)d_i,$$

where $T_i^{\mathrm{s}}(t)$ denotes the total sprinting time up to time t and $T_i^{\mathrm{c}}(t)$ the total cruising time. Using the fact that $P_i(t) = d_i t + x_i(t)$ and dividing the previous expression by $\mu_i t$, we have that

$$\rho_i + \frac{x_i(t)}{\mu_i t} = \frac{T_i^{\mathrm{s}}(t)}{t} + \frac{T_i^{\mathrm{c}}(t)}{t}\rho_i. \tag{4.4}$$

Now, suppose the system is such that, for all $i \in \mathcal{Q}$,

$$\lim_{t\to\infty} \frac{x_i(t)}{t} = 0,$$

a condition that we will assume throughout. Taking the limit on both sides of (4.4), we get

$$\rho_i = \lim_{t\to\infty} \left(\frac{T_i^{\mathrm{s}}(t)}{t} + \frac{T_i^{\mathrm{c}}(t)}{t}\rho_i \right).$$

It thus follows from this equation that the sprinting and cruising time fractions must converge to a limit, which we denote by p_i^{s} and p_i^{c}, respectively. That is,

$$p_i^{\mathrm{s}} = \lim_{t\to\infty} \frac{T_i^{\mathrm{s}}(t)}{t} \quad \text{and} \tag{4.5}$$

$$p_i^{\mathrm{c}} = \lim_{t\to\infty} \frac{T_i^{\mathrm{c}}(t)}{t}, \tag{4.6}$$

Furthermore, for every part type i, we have the long-term frequency relation

$$\rho_i = \frac{p_i^{\mathrm{s}}}{1 - p_i^{\mathrm{c}}}. \tag{4.7}$$

This last relationship states that for systems with no cruising (i.e., systems with $p_i^{\mathrm{c}} = 0$) the long-term fraction of sprinting time must be equal to the utilization for that item. On the other hand, if the system cruises with type i, the relation states that the utilization must be equal to the ratio of sprinting time over non-cruising time for that item.

A similar relationship can be established between the production and setup times of all items. Let $S_{ij}(t)$ denote the total *time* spent performing i-to-j setups up to t. Then, we have that

$$\sum_{i}^{N} \left(\sum_{j}^{N} S_{ij}(t) + T_i^{\mathrm{s}}(t) + T_i^{\mathrm{c}}(t) \right) = t.$$

Dividing by t and using our previous results, we see that the limit

$$s_{ij} = \lim_{t\to\infty} \frac{S_{ij}(t)}{t} \tag{4.8}$$

must exist. This quantity represents the long-term fraction of *time* that the system spends performing i-to-j changeovers, and thus we have that

$$\sum_i^N \left(\sum_j^N s_{ij} + p_i^{\mathrm{s}} + p_i^{\mathrm{c}} \right) = 1.$$

Using (4.7) to replace p_i^{s} in the previous equation, we get the long-term capacity relation

$$\sum_i^N \left(\sum_j^N s_{ij} + p_i^{\mathrm{c}}(1 - \rho_i) \right) = 1 - \rho. \tag{4.9}$$

Note that (4.9) expresses the intuitive notion that, for a non-cruising system, the fraction of time available for setup changes in the long run is $1 - \rho$.

We can obtain yet another set of long-term relations as follows. Define $Q_i(t)$ to be equal to the number of *completed* production runs of type i up to time t and let $\Delta t_{i,n}$ denote the *production* time of the n-th run of type i (i.e., the total time spent sprinting and cruising during the run).[2]

Then, at any t we must have

$$\frac{\sum_{n=1}^{Q_i(t)} \Delta t_{i,n}}{t} \leq \frac{T_i^{\mathrm{s}}(t) + T_i^{\mathrm{c}}(t)}{t} \leq \frac{\sum_{n=1}^{Q_i(t)} \Delta t_{i,n}}{t} + \frac{\Delta t_{i,Q_i(t)+1}}{t}.$$

Now notice that since $x_i(t)/t$ tends to 0 and the length of any run depends linearly on $x_i(t)$,

$$\frac{\Delta t_{i,Q_i(t)+1}}{t} \to 0.$$

[2]Recall that, per our definition in Chapter 2, a run of type i is defined to include the time changing over into that part type. Thus, $\Delta t_{i,n}$ consists of the *non-setup* portion of the n-th run of type i.

Therefore, we can conclude that

$$\lim_{t\to\infty}\frac{\sum_{n=1}^{Q_i(t)}\Delta t_{i,n}}{t}=p_i^{\mathrm{s}}+p_i^{\mathrm{c}}.$$

Dividing and multiplying the above equation by $Q_i(t)$, we get that

$$\lim_{t\to\infty}\frac{\sum_{n=1}^{Q_i(t)}\Delta t_{i,n}}{Q_i(t)}\frac{Q_i(t)}{t}=p_i^{\mathrm{s}}+p_i^{\mathrm{c}},$$

and, since by assumption $Q_i(t)\to\infty$ for all i, this shows that the limits

$$\overline{\Delta t}_i=\lim_{m\to\infty}\frac{\sum_{n=1}^{m}\Delta t_{i,n}}{m}\quad\text{and}\tag{4.10}$$

$$n_i=\lim_{t\to\infty}=\frac{Q_i(t)}{t}\tag{4.11}$$

exist and that the long-term relation

$$\overline{\Delta t}_i n_i=p_i^{\mathrm{s}}+p_i^{\mathrm{c}}\tag{4.12}$$

must hold. The term $\overline{\Delta t}_i$ represents the average duration of the *production* portion of a run of type i and n_i represents the *production frequency* or long-term number of setups *into* type i per time.

Furthermore, as defined in Section 4.1, if $Q_{ij}(t)$ denotes the number of completed runs that involved an i-to-j setup, we can write

$$n_{ij}=\lim_{t\to\infty}\frac{Q_{ij}(t)}{t},$$

which represents the long-term frequency of changeovers from type i into type j. We then have that

$$\sum_{i \in \mathcal{Q}} n_{ij} = n_j$$

and

$$s_{ij} = n_{ij} S_{ij}.$$

Finally, the production time of the n-th run of type i can be split into its sprinting and cruising segments, denoted by $\Delta t^{\mathrm{s}}_{i,n}$ and $\Delta t^{\mathrm{c}}_{i,n}$. Averaging these segments over all runs, we get

$$\begin{aligned} \overline{\Delta t}^{\mathrm{s}}_i n_i &= p^{\mathrm{s}}_i \quad \text{and} \\ \overline{\Delta t}^{\mathrm{c}}_i n_i &= p^{\mathrm{c}}_i, \end{aligned}$$

where $\overline{\Delta t}^{\mathrm{s}}_i$ is the average time sprinting per run and $\overline{\Delta t}^{\mathrm{c}}_i$ is the average time cruising per run.

Equations (4.7), (4.9), and (4.12) will be used in the derivations of our lower bounds in the next sections.

4.2.2 Lower Bound for J-OP

We begin by justifying the statement that, under our modeling assumptions, it is not optimal to produce at any rate other than either the maximum production rate or the demand rate. The argument can be readily seen from Fig. 4-2, where we show a surplus trajectory of $x_i(t)$ generated by an arbitrary production rate $u_i(t)$, and a perturbed trajectory $x'_i(t)$ that satisfies $u_i(t) \in \{0, d_i, \mu_i\}$. Notice how $x'_i(t)$ encloses a smaller area with the target line $x_i = Z^{\mathrm{U}}_i$ (and thus has a smaller J cost) while still respecting the original production runs' start and end times. Therefore, this perturbation has no effect on the trajectories and setup frequencies of the other part types, which means that it is a feasible trajectory with a lower cost J. Furthermore, notice that there is no need for the system to cruise unless $x_i(t) = Z_i^{\mathrm{U}}$, since a lower cost can always be achieved through a perturbation that involves sprinting until either the run concludes or until Z_i^{U} is reached.

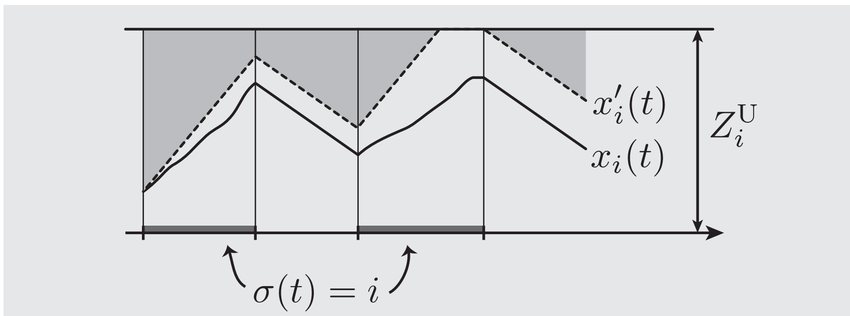


Figure 4-2: Justification of the restriction that $u_i(t) \in \{0, d_i, \mu_i\}$. Notice that the original trajectory $x_i(t)$ can be perturbed into a lower J-cost trajectory $x_i'(t)$ that always sprints except when $x_i = Z_i^{\mathrm{U}}$. This perturbation does not alter the length of the production runs.

The previous arguments support the use of the CC Class as a suitable class of policies, at least in terms of J-OP. However, while we have shown that an optimal trajectory should only sprint when $x_i < Z_i^{\mathrm{U}}$ or cruise at $x_i = Z_i^{\mathrm{U}}$, in the CC Class we also require that *changeovers* only occur at that base stock level. We will refer to this condition as the $\boldsymbol{Z}^{\mathrm{U}}$*-Switch Rule*, since it mirrors the Zero-Switch Rule in the Economic Lot Scheduling Problem (see Fig. 4-9 and accompanying discussion).

As discussed in Section 4.3.1, the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule is not always optimal; that is, in some cases it is better to conclude a run *before* the target level is reached. However, allowing for this could lead to a policy that is harder to state and implement in the shop floor. Moreover, determining when the $\boldsymbol{Z}^{\mathrm{U}}$-rule is optimal may not be an easy task. For example, notice that if we wanted to perturb the trajectory $x_i(t)$ in Fig. 4-2 so that it also satisfied the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule, we would need to alter the duration of the first run of type i shown in the figure, but this would affect the production times of the other part types. Thus, we would need to consider all N surplus trajectories as we search for a feasible perturbation that satisfies the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule and that has a lower cost. Fortunately, based on reported experience with the ELSP (see Maxwell 1964 and Gallego 1990) and on our simulation results to be discussed later, it is reasonable to expect that the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule will be close to optimal for J-OP in most cases.

Proceeding with the derivation of our bound, we now relax the constraints in the system and show that, *under these looser restrictions*, the optimal trajectory will in fact satisfy the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule. This relaxation consists of ignoring possible interferences between the items' production schedules and instead enforcing the long-term relation (4.9), as well as a balance constraint on the production frequencies.

Since the relaxed set of constraints only depends on the long-term production frequencies and time fractions, any trajectory perturbation that keeps constant the number of runs and sprinting times over a fixed period of time will be feasible. This fact allows us to conclude that the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule must be satisfied or, equivalently, that the *optimal trajectories in the relaxed constraint set are of the CC Class type.* To show this, consider Fig. 4-3, which depicts a trajectory $x_i(t)$ over some fixed period of time, and note that two of the runs terminate before reaching the target level. The lower-cost trajectory $x_i'(t)$, on the other hand, always terminates at the target level and respects both the number of runs and total sprinting/cruising time of $x_i(t)$ over the same period. Thus, by perturbing $x_i(t)$ into $x_i'(t)$, we have obtained a CC Class trajectory that has a lower cost and that meets the relaxed set of constraints.

Consider now Fig. 4-4, where we show the typical shape of a CC Class surplus trajectory during the segment of time starting at the conclusion of the $(n-1)$-st production run of type i and ending at the conclusion of the n-th run of that part type. The contribution of

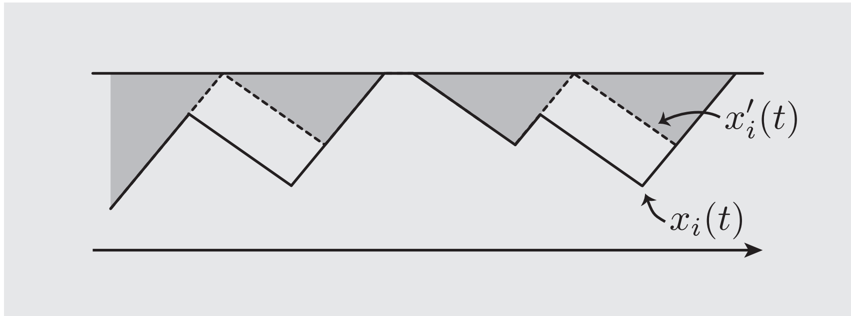


Figure 4-3: Proof that the CC Class is optimal for J-OP when interference between items is relaxed. Note that the original trajectory $x_i(t)$ has been perturbed into a lower J-cost trajectory that satisfies the $\boldsymbol{Z}^{\mathrm{U}}$-switch rule. The perturbed trajectory respects the original cruising and sprinting time fractions, as well as the production frequencies.

this segment to cost J depends on the area $A_{i,n}$ enclosed by the surplus trajectory and the Z_i^{U} target level line. If $\Delta t^{\mathrm{s}}_{i,n}$ denotes the *sprinting* portion of the n-th run of type i, this area is given by

$$A_{i,n} = \frac{1}{2}\left(\Delta t^{\mathrm{s}}_{i,n} + \frac{\Delta t^{\mathrm{s}}_{i,n}(\mu_i - d_i)}{d_i}\right)\Delta t^{\mathrm{s}}_{i,n}(\mu_i - d_i),$$

which simplifies into

$$A_{i,n} = \frac{(\mu_i - d_i)}{2\rho_i}\left(\Delta t^{\mathrm{s}}_{i,n}\right)^2.$$

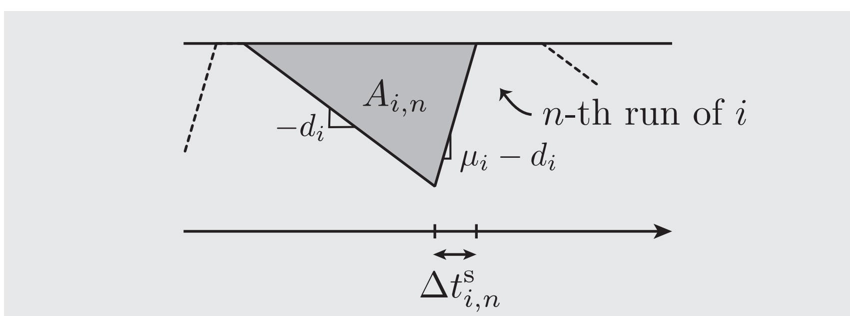


Figure 4-4: Segment comprising the period between the end of the $(n-1)$-st run and the n-th run of type i. The contribution to cost J during this period depends on the area $A_{i,n}$.

The above equation shows that each of the areas $A_{i,n}$ contributing to J depends on the squared sprinting time of the n-th run of i. It then follows that, in the optimal trajectory of the *relaxed* constraint set, *all sprinting times must be equal.* This is true because, for any fixed period of time, the sprinting times on each run are only constrained by their total sum and, thus, the sum of the areas $A_{i,n}$ is minimized when all sprinting segments have the same length.[3]

Now, given that all sprinting segments of type i have equal length $\overline{\Delta t}_i^{\mathrm{s}}$, and defining J_i to be the contribution of this item to cost J, we get

$$\begin{aligned}
J_i &= \lim_{t\to\infty} \sum_{n=1}^{Q_i(t)} c_i \frac{A_{i,n}}{t} \\
&= \lim_{t\to\infty} \sum_{n=1}^{Q_i(t)} \frac{c_i(\mu_i - d_i)}{2\rho_i} \frac{\left(\overline{\Delta t}_i^{\mathrm{s}}\right)^2}{t} \\
&= \frac{c_i(\mu_i - d_i)}{2\rho_i} \left(\overline{\Delta t}_i^{\mathrm{s}}\right)^2 \lim_{t\to\infty} \frac{Q_i(t)}{t} \\
&= \frac{c_i(\mu_i - d_i)}{2\rho_i} \left(\overline{\Delta t}_i^{\mathrm{s}}\right)^2 n_i,
\end{aligned}$$

Using the fact that $\overline{\Delta t}_i^{\mathrm{s}} n_i = p_i^{\mathrm{s}}$ and (4.7) we obtain that

$$J_i = \frac{c_i\rho_i(1-\rho_i)}{2\tau_i} \frac{(1-p_i^{\mathrm{c}})^2}{n_i}, \tag{4.13}$$

and cost J will be given by

$$J = \sum_i J_i + \sum_{i,j} n_{ij} K_{ij}.$$

Minimizing the previous expression, which solely depends on the variables n_i and p_i^{c}, gives the lower bound for J-OP. This minimization is subject to the long-term capacity constraint (4.9) and a balance relation that ensures that the frequency of changeovers *into* part type i

[3] In other words, we are minimizing an objective function of the form $\sum_n \left(\Delta t_{i,n}^{\mathrm{s}}\right)^2$, subject to a constraint $\sum_n \Delta t_{i,n}^{\mathrm{s}} = C$, which has as solution $\Delta t_{i,n}^{\mathrm{s}} = \text{constant}$.

is equal to the frequency of setups changeovers *from* part type i. That is,

$$\begin{aligned} &\textstyle\sum_{i=1}^{N} n_{ij} = n_j \quad \forall\, j \in \mathcal{Q}, \text{ and} \\ &\textstyle\sum_{i=1}^{N} n_{ji} = n_j \quad \forall\, j \in \mathcal{Q} \end{aligned} \tag{4.14}$$

We summarize below the nonlinear program for the lower bound to J-OP, denoted as J_{LB}. While J_{LB} can be easily obtained through a numeric optimization routine, for the sequence-independent case and without balance constraints (4.14) Lan and Olsen (2006) have obtained an analytical solution. Lan (2000) reports that in many cases the bound is not very sensitive to the balance constraints.

Lower Bound J_{LB}

$$\min_{n_i, n_{ij}, p_i^{\mathrm{c}}} \quad \sum_{i=1}^{N} \frac{c_i \rho_i (1-\rho_i)}{2\tau_i} \frac{(1-p_i^{\mathrm{c}})^2}{n_i} + \sum_{i,j \in \mathcal{Q}} n_{ij} K_{ij} \tag{4.15}$$

$$\text{subject to} \quad \begin{array}{ll} \sum_{i=1}^{N} \left(\sum_{j}^{N} n_{ij} S_{ij} + p_i^{\mathrm{c}} (1-\rho_i) \right) = 1 - \rho & \\ \sum_{i=1}^{N} n_{ij} = n_j & \forall\, j \in \mathcal{Q} \\ \sum_{i=1}^{N} n_{ji} = n_j & \forall\, j \in \mathcal{Q} \\ n_{ii} = 0 & \forall\, i \in \mathcal{Q} \\ n_i, n_{ij}, p_i^{\mathrm{c}} \geq 0 & \forall\, i, j \in \mathcal{Q}. \end{array}$$

It is important to emphasize that the above problem will give a set of production frequencies n_i^* and cruising time fractions $p_i^{\mathrm{c}*}$ that are optimal under the relaxed set of constraints but likely not feasible in the original problem where interference is not allowed. To make this distinction more evident, we will refer to n_i^* and $p_i^{\mathrm{c}*}$ as the *ideal* production frequencies and the *ideal* cruising time fractions, respectively. Only when a policy's trajectories are able to achieve these long-term values, the ideal schedule will be optimal.

Figure 4-5 depicts the typical shape of the ideal schedule obtained through the solution of J_{LB}. The period of time between any two consecutive runs of type i is denoted by T_i^*, where $T_i^* = 1/n_i^*$. Notice that the schedule also gives us the ideal *peak surplus deviation* y_i^*, a quantity that will be the basis for selecting the hedging zone later. It follows from the geometry of the figure that

$$T_i^* = y_i^* \left(\frac{1}{d_i} + \frac{1}{\mu_i - d_i} \right) + T_i^* p_i^{\mathrm{c}*},$$

and, therefore, the ideal deviation is given by

$$y_i^* = \frac{\rho_i(1-\rho_i)}{n_i^* \tau_i}(1 - p_i^{\mathrm{c}*}). \tag{4.16}$$

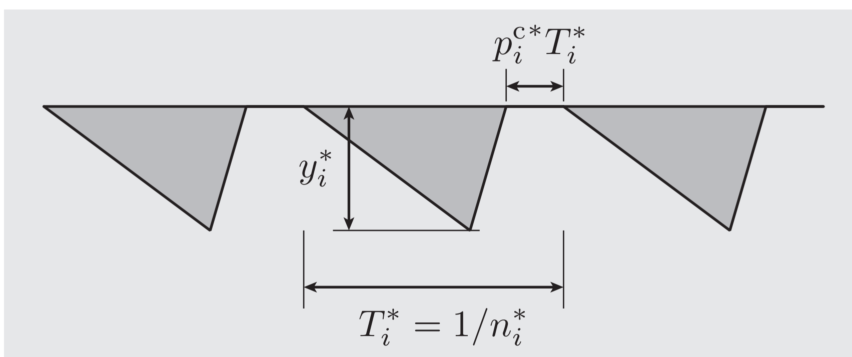


Figure 4-5: Schematic diagram of the *ideal* surplus trajectory for part type i. Notice that all runs are equal and the surplus deviations have the same peak value y_i^*. The cruising time on each run is given by the product $T_i^* p_i^{\mathrm{c}*}$.

4.2.3 Lower Bound for I-OP

The derivation of the lower bound for I-OP follows the same line of reasoning that led to J_{LB}. The main difference in this case is that the optimal level at which cruising should occur is $x_i(t) = 0$ (see Moon et al. 1991 or Elhafsi and Bai 1997). This fact can be verified through a perturbation argument, shown in Fig. 4-6, where we notice that the perturbation respects the original run's production times and that it has no effect on the trajectories of the other part types.

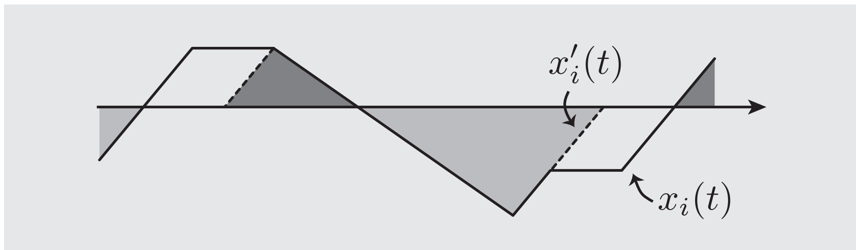


Figure 4-6: Justification of the fact that for I-OP it is optimal to cruise only when $x_i = 0$. Notice how the production runs in the perturbed trajectory $x_i'(t)$ start at the same times that in the original trajectory $x_i(t)$ but have a lower I cost.

As with J-OP, the lower bound is obtained by ignoring interferences between part types and enforcing instead the long-term capacity constraint (4.9) and the balance constraints (4.14). Optimization of the trajectories under this relaxed set of constraints leads to the conclusion that production of type i must never begin with $x_i(t) > 0$ or end with $x_i(t) < 0$. This conclusion is explained in Fig. 4-7, where a lower I-cost perturbation that respects the original production frequencies and the sprinting and cruising fractions is depicted.

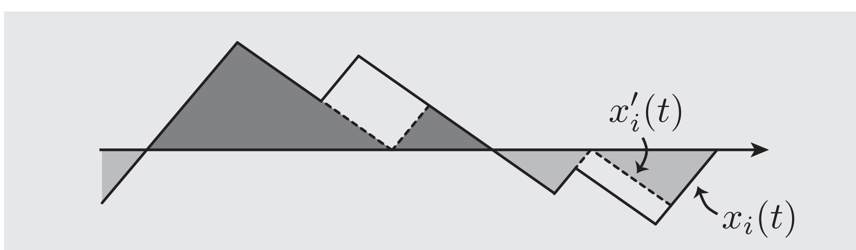


Figure 4-7: When we relax the no-interference condition between part types, it is optimal to never start producing an item with inventory or end a run with backlog, as verified through a perturbation argument. This property is called the Extended Zero-Switch Rule (EZSR).

In the ELSP, where backlogs are not allowed, it is customary to assume that solutions possess the property that production of any type i starts at the precise instant when $x_i(t) = 0$. This assumption corresponds to the previously-mentioned Zero-Switch Rule (ZSR). When backlogs are allowed, Gallego and Roundy (1992) refer to the condition that production of any item never starts with inventory or concludes with backlog as the Extended Zero-Switch Rule (EZSR).

In addition to satisfying the EZSR, we can also conclude (using the same arguments as in the previous section) that the optimal relaxed trajectories will have the same inventory and backlog peak values on each production run of type i and, therefore, equal run lengths. Let $\overline{\Delta t}_i^{\mathrm{s,I}}$ denote the length of the sprinting portion of any run of i during the period when there is inventory of that part type (i.e., during times when $x_i(t) > 0$), and $\overline{\Delta t}_i^{\mathrm{s,B}}$ the length of the sprinting portion when there is backlog. We can express the contribution of type i to cost I, denoted by I_i, as

$$I_i = \frac{h_i(\mu_i - d_i)}{2\rho_i}(\overline{\Delta t}_i^{\mathrm{s,I}})^2 n_i + \frac{b_i(\mu_i - d_i)}{2\rho_i}(\overline{\Delta t}_i^{\mathrm{s,B}})^2 n_i, \tag{4.17}$$

and hence

$$I = \sum_i I_i + \sum_{i,j} n_{ij} K_{ij}.$$

Now, with reference to Fig. 4-8, consider a typical segment of time starting at the end of a run of type i and concluding at the end of the subsequent run of this part type. The length of this period is $1/n_i$ and the surplus of type i will be positive during this period on an interval of duration $\overline{\Delta t}_i^{\mathrm{s,I}}(1 + \frac{\mu_i - d_i}{d_i})$. Thus, we must have that

$$\theta_i = \overline{\Delta t}_i^{\mathrm{s,I}}(1 + \frac{\mu_i - d_i}{d_i})n_i,$$

and it follows that

$$\overline{\Delta t}_i^{\mathrm{s,I}} = \frac{\theta_i \rho_i}{n_i}.$$

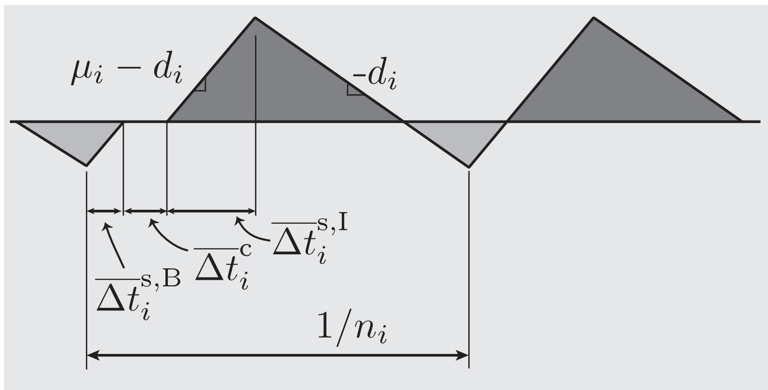


Figure 4-8: Optimal shape of trajectories for I-OP when the no-interference constraint is relaxed. Note that every run has the same inventory and backlog peak levels, and the run lengths are all equal.

Furthermore, the total sprinting time during each run, $\overline{\Delta t}_i^{\mathrm{s,I}}$, satisfies

$$\overline{\Delta t}_i^{\mathrm{s}} = (1/n_i) p_i^{\mathrm{s}} = \overline{\Delta t}_i^{\mathrm{s,I}} + \overline{\Delta t}_i^{\mathrm{s,B}}.$$

Therefore, solving for $\overline{\Delta t}_i^{\mathrm{s,B}}$ and using (4.7), it follows that

$$\begin{aligned}
\overline{\Delta t}_i^{\mathrm{s,B}} &= \frac{p_i^{\mathrm{s}} - \theta_i \rho_i}{n_i} \\
&= \frac{\rho_i (1 - p_i^{\mathrm{c}}) - \theta_i \rho_i}{n_i} \\
&= \frac{\rho_i}{n_i} (1 - p_i^{\mathrm{c}} - \theta_i).
\end{aligned}$$

Substituting in (4.17) the expressions for $\overline{\Delta t}_i^{\,\mathrm{s,I}}$ and $\overline{\Delta t}_i^{\,\mathrm{s,B}}$ just derived, we get that

$$\begin{aligned} I_i &= n_i \frac{\mu_i - d_i}{2\rho_i} \left[h_i \frac{\theta_i^2 \rho_i^2}{n_i^2} + b_i \frac{\rho_i^2}{n_i^2} (1 - p_i^{\mathrm{c}} - \theta_i)^2 \right] \\ &= \frac{\rho_i (1 - \rho_i)}{2 n_i \tau_i} \left[h_i \theta_i^2 + b_i (1 - p_i^{\mathrm{c}} - \theta_i)^2 \right]. \end{aligned}$$

Finally, the optimal trajectory should also satisfy the expression for the optimal service level (4.2). Given that the system only cruises when $x_i(t) = 0$, the long-term fraction of time with zero surplus is equal to p_i^{c}. Thus, we have that

$$\theta_i^* = \frac{b_i}{h_i + b_i} (1 - p_i^{\mathrm{c}}).$$

Substituting θ_i^* in our expression for I_i we finally get, after some minor algebra, that

$$I_i = \frac{\rho_i (1 - \rho_i)}{2\tau_i} \frac{h_i b_i}{h_i + b_i} \frac{(1 - p_i^{\mathrm{c}})^2}{n_i}.$$

The lower bound for cost I is obtained by optimizing the sum of the terms I_i and setup costs, subject to the same constraints used in J_{LB}. Notice that, J_{LB} and I_{LB} only differ by the weighting constants in the first terms of the objective function; in the former, the constants are the c_i's coefficients, while in the latter the constants are given by $h_i b_i / (h_i + b_i)$ for $i \in \mathcal{Q}$.

Lower Bound I_{LB}

$$\min_{n_i, n_{ij}, p_i^{\mathrm{c}}} \quad \sum_{i=1}^{N} \frac{\rho_i(1-\rho_i)}{2\tau_i} \frac{h_i b_i}{h_i + b_i} \frac{(1-p_i^{\mathrm{c}})^2}{n_i} + \sum_{i,j \in \mathcal{Q}} n_{ij} K_{ij} \tag{4.18}$$

$$\text{subject to} \quad \begin{array}{ll} \sum_{i=1}^{N} \left(\sum_{j}^{N} n_{ij} S_{ij} + p_i^{\mathrm{c}}(1-\rho_i) \right) = 1 - \rho & \\ \sum_{i=1}^{N} n_{ij} = n_j & \forall\, j \in \mathcal{Q} \\ \sum_{i=1}^{N} n_{ji} = n_j & \forall\, j \in \mathcal{Q} \\ n_{ii} = 0 & \forall\, i \in \mathcal{Q} \\ n_i, n_{ij}, p_i^{\mathrm{c}} \geq 0 & \forall\, i, j \in \mathcal{Q}. \end{array}$$

4.3 Open-Loop Schedules

4.3.1 The Economic Lot Scheduling Problem

The Economic Lot Scheduling Problem is a classic problem in the deterministic, open-loop production scheduling literature. Most of the early work on the problem is summarized in the excellent review by Elmaghraby (1978). Hsu (1983) showed that the problem is NP-Hard and, therefore, research on the ELSP has generally focused on comparing a wide variety of solution heuristics, including optimal solutions to restricted versions of the problem (Bomberger 1966), heuristics based on lower bounds (Dobson 1987), and the use of genetic algorithms (Moon and Silver 2002).

In its original form, the ELSP essentially consists of solving I-OP with infinite backlog costs and sequence-independent setup times and costs. The production rate $u_i(t)$ is restricted to the set $\{0, \mu_i\}$, which means that no cruising is allowed. However, *idling* (i.e., setting $u_i(t) = 0$ for some time even though $\sigma(t) = i$) is allowed for reducing changeover frequencies, which may be desirable in systems with low utilizations and large setup costs. The solutions to the ELSP are always in steady-state and typically follow a periodic sequence. As mentioned, it is also common to assume that the solution will satisfy the Zero-Switch Rule, which states that production of any item should begin at the instant when its inventory is depleted.

Figure 4-9 shows how any solution to the ELSP can be transformed into a valid solution to J-OP by applying vertical and horizontal reflections, and by shifting the trajectory up by $\boldsymbol{Z}^{\mathrm{U}}$. Notice how, under this transformation, the ZSR becomes the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule, which means that if the former is near optimal for the ELSP, the latter will be so for J-OP.

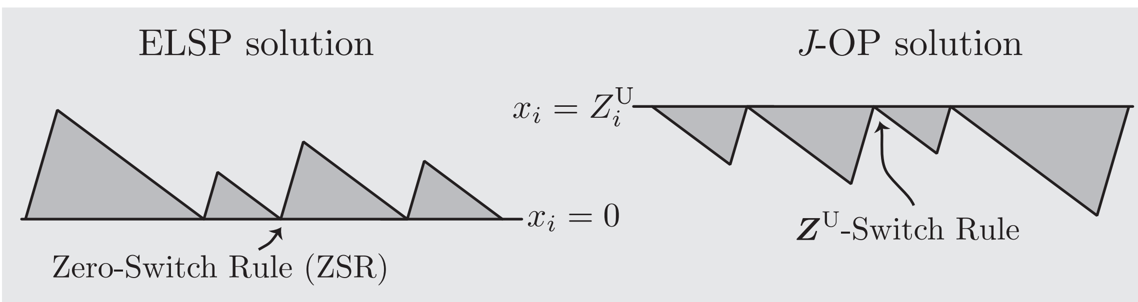


Figure 4-9: Schematic diagram showing the relationship between ELSP solutions and J-OP solutions. Notice how for the non-cruising case, the two solutions relate through a transformation consisting of horizontal and vertical reflections.

A solution to the ELSP will typically consist of a sequence of products and a set of production times and idle times. That is, let $\boldsymbol{f} = [f^1, f^2, \ldots, f^M]$ denote a periodic sequence of length M, where $f^n \in \mathcal{Q}$ corresponds to the item produced in the n-th position of the sequence. Then, the ELSP solution will consist of two M-dimensional vectors specifying the production and idling times at each position on the sequence. This information, together with the Zero-Switch Rule assumption, is enough to reconstruct the schedule. In Section 4.3.2, we discuss the problem of finding the optimal schedule for a *given* sequence $\boldsymbol{f}$, which is an easy problem to solve. To solve the ELSP, of course, we would need to find the best such schedule over *all* sequences $\boldsymbol{f}$.

The most severe limitation regarding the ELSP formulation lies in its open-loop nature. This means that solutions to the problem do not provide any guidance on how to recover the schedule in the event of a disruption, or even how to reach the prescribed steady-state inventory levels in the first place. Furthermore, maintaining zero backlogs (i.e., 100% service level) while still managing to follow the ZSR, is only possible in a perfectly-deterministic environment; in the real world, where unforeseen disruptions are bound to occur, a near perfect service level would require holding very large amounts of safety stock.

Some of the drawbacks mentioned above have been addressed by extending the ELSP original formulation. One of the most relevant extensions for our present discussions is that by Gallego and Roundy (1992), which relaxes the zero-backlog constraint, leading to the so-called Extended ELSP. Furthemore, Gallego (1990) derived a closed-loop recovery policy for this extended version of the problem. This policy has the attractive feature that it converges to the pre-determined schedule in the absence of disruptions, and it does so optimally with respect to any other recovery strategy that follows the same sequence. Gallego's policy is discussed in more detail in Section 4.5.

4.3.2 Optimal $\boldsymbol{f}$-Cyclic Schedules

If a sequence $\boldsymbol{f}$ is known, the optimal schedule to either J-OP or I-OP for that sequence can be obtained by solving a parametric quadratic programming problem. That is, for each value of the cycle's period $T_{\boldsymbol{f}}$, the optimal production and idling time vectors are found through the solution of a quadratic program. The optimal schedule for a given periodic sequence $\boldsymbol{f}$ is referred to as the *optimal $\boldsymbol{f}$-cyclic schedule* (Gallego 1990).

For completeness, we now state the parametric quadratic program for I-OP based on the formulation by Dobson (1987), but replacing idling by cruising and assuming the more general case of sequence-dependent setups. In what follows, the indices of the positions in the sequence $\boldsymbol{f}$ are to be interpreted in a circular fashion so that, if $|\boldsymbol{f}| = M$, then $f^{M+1} = f^1$ and $f^{-1} = f^M$. We also remark that, as stated in Lan and Olsen (2006), the term "cyclic" is often reserved in the polling models literature for sequences in which each item is produced exactly once in the sequence. This restriction is not assumed here, but we do require that the sequence contains all part types at least once and that it does not repeat items in consecutive positions. These properties are summarized in the definition below.

Definition 4.3.1 (Valid sequence $\boldsymbol{f}$) *A sequence $\boldsymbol{f}$ of length $M \geq N$ is called valid if for every $i \in \mathcal{Q}$ there is at least one position n such that $f^n = i$, and if for every position n we have $f^n \neq f^{n+1}$.*

For $i = 1, \ldots, N$, let G_i represent the set of locations in the sequence where product i is produced. That is,

$$G_i = \{\, n \leq M \ \mid f^n = i \,\} .$$

For any position n in the sequence, let $\nu(n)$ denote the next position in which the same part type gets produced again. That is, if $f^n = i$, then $f^{\nu(n)} = i$ and $f^m \neq i$ for $n < m < \nu(n)$. The set L_n is defined to contain all *positions* between n and $\nu(n)$. Mathematically, for $n = 1, 2, \ldots, M$, we write

$$L_n = \{\, m \ \mid n < m < \nu(n) \,\} .$$

To illustrate the previous definitions, suppose we have a three-part-type system and a sequence with 5 positions (i.e., $N = 3$ and $M = 5$), given by $f = [1, 2, 3, 1, 2]$. Then, if $n = 1$, where type 1 gets produced, we have that $\nu(n) = 4$. Also, for this sequence, the sets defined above will be

$$
\begin{aligned}
G_1 &= \{1,4\} \\
G_2 &= \{2,5\} \\
G_3 &= \{3\} \\
L_1 &= \{2,3\} \\
L_2 &= \{3,4\} \\
L_3 &= \{4,5,1,2\} \\
L_4 &= \{5\} \\
L_5 &= \{1\}
\end{aligned}
$$

Consider the period of time comprising the production of item i at the n-th position in the sequence (i.e., $f^n = i$) and concluding at the beginning of the next production run of that item (which occurs at position $\nu(n)$). The typical shape of type's i surplus trajectory is shown in Fig. 4-10. Note that we are assuming that the Extended Zero-Switch Rule is enforced so, to be more precise, the term optimal $\boldsymbol{f}$-cyclic sequence should be interpreted as optimal EZSR $\boldsymbol{f}$-cyclic sequence.

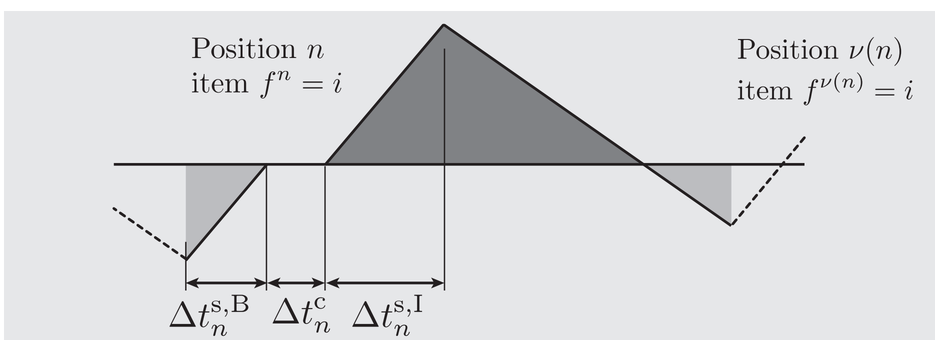


Figure 4-10: Surplus trajectory of type $f^n = i$ during the time period comprising the start of the n-th position in the sequence and concluding at the start of the $\nu(n)$-th position.

We begin by computing the cost contribution from the segment shown in Fig. 4-10, which is denoted as I_n and is given by

$$I_n = \frac{1}{T_{\boldsymbol{f}}}\left[h_i \frac{\mu_i - d_i}{2\rho_i}(\Delta t_n^{\mathrm{s,I}})^2 + b_i \frac{\mu_i - d_i}{2}\left((\Delta t_n^{\mathrm{s,B}})^2 + \frac{\mu_i - d_i}{d_i}(\Delta t_{\nu(n)}^{\mathrm{s,B}})^2\right)\right]. \tag{4.19}$$

In the above expression, $\Delta t_n^{\mathrm{s,I}}$ corresponds to the sprinting portion of this segment with type i inventory present, $\Delta t_n^{\mathrm{s,B}}$ the time sprinting with backlog present, $T_{\boldsymbol{f}}$ is the period of the cyclic sequence, and $f^n = i$. The cost we seek to optimize is obtained by summing the contributions over all positions in the sequence and adding setup costs. That is,

$$I = \sum_{n=1}^{M}\left(I_n + \frac{K_{f^{n-1}f^n}}{T_{\boldsymbol{f}}}\right).$$

Regarding the constraints of the problem, we need to ensure that the total amount produced for each item i during the cycle is equal to the total amount demanded during that same period. Since item i is produced at the positions in G_i, this constraint is expressed as

$$\sum_{n \in G_i}\left[\left(\Delta t_n^{\mathrm{s,I}} + \Delta t_n^{\mathrm{s,B}}\right)\mu_i + \Delta t_n^{\mathrm{c}} d_i\right] = T_{\boldsymbol{f}} d_i. \tag{4.20}$$

The solution should also have consistency between the amount of item i produced at the n-th position and the backlog level at the beginning of the next run of this part type, at position $\nu(n)$. Since L_n includes all positions between n and $\nu(n)$, we thus have that

$$\begin{aligned}\Delta t_n^{\mathrm{s,I}}(\mu_{f^n} - d_{f^n}) - \left[\sum_{m \in L_n}\left(\Delta t_m^{\mathrm{s,I}} + \Delta t_m^{\mathrm{s,B}} + \Delta t_m^{\mathrm{c}} + S_{f^{m-1}f^m}\right) + S_{f^{\nu(n)-1}f^{\nu(n)}}\right] d_{f^n} = \\ \Delta t_{\nu(n)}^{\mathrm{s,B}}(\mu_{f^n} - d_{f^n}),\end{aligned} \tag{4.21}$$

for each position $n = 1, 2, \ldots, M$. On the left hand side of this equation, the first term corresponds to the amount of inventory that is built up during the run, while the product of

d_{f^n} and the term inside the brackets corresponds to the demand generated between the two consecutive runs of product f^n. The term of the right hand side of the equation corresponds to the amount of backlog present just before type i starts being produced again.

In addition, the sprinting, cruising, and setup times on each position must add up to the total cycle period. This constraint is written as

$$\sum_{n=1}^{M} \left(\Delta t_n^{\mathrm{s,B}} + \Delta t_n^{\mathrm{s,I}} + \Delta t_n^{\mathrm{c}} + S_{f^{n-1}f^n}\right) = T_{\boldsymbol{f}}.$$

Finally, by enforcing nonnegativity constraints on all the variables, we ensure that the trajectory satisfies the Extended ZSR.

As shown by Dobson (1987), the constraints given by (4.20) are redundant, since substituting $T_{\boldsymbol{f}}$ in that equation gives (4.21) summed over all $n \in G_i$. Thus, the optimal $\boldsymbol{f}$-cyclic schedule is obtained by solving the problem shown below.

Optimal $\boldsymbol{f}$-Cyclic Schedule for I-OP

$$\min \quad \sum_{n=1}^{M} \left(I_n + \frac{K_{f^{n-1}f^n}}{T_{\boldsymbol{f}}}\right) \tag{4.22}$$

subject to

Constraint (4.21) for $n = 1, 2, \ldots, M$

$\sum_{n=1}^{M} \left(\Delta t_n^{\mathrm{s,B}} + \Delta t_n^{\mathrm{s,I}} + \Delta t_n^{\mathrm{c}} + S_{f^{n-1}f^n}\right) = T_{\boldsymbol{f}}$

$\Delta t_n^{\mathrm{s,I}}$, $\Delta t_n^{\mathrm{s,B}}$, $\Delta t_n^{\mathrm{c}} \geq 0$ for $n = 1, 2, \ldots, M$

$T_{\boldsymbol{f}} \geq 0$

Note: I_n is given by (4.19).

From the transformation between solutions to ELSP and J-OP depicted in Fig. 4-9, we see that the optimal $\boldsymbol{f}$-schedule for J-OP can be obtained by solving the above problem with backlogs constrained to zero and replacing the setup matrices by their transposes.[4]

The fact that the optimal schedule for a given sequence is, at least in theory, easy to find by solving the above optimization problem has lead to a typical approach for obtaining solutions to the Extended ELSP. This approach consists of two steps: (1) finding a good sequence $\boldsymbol{f}$ through some heuristic procedure, and (2) obtaining the optimal $\boldsymbol{f}$-cyclic schedule for that sequence. The heuristic for the first step is typically based on the ideal production frequencies that result from I_{LB}. We do not need to discuss this further here because, in the simulation experiments of Section 4.6, we will deal exclusively with three-part-type systems. For these small systems, it is feasible to simply perform an exhaustive search over all $\boldsymbol{f}$-cyclic schedules of up to some reasonably large number of positions, say, M^*, and choose the best one. This best sequence is referred to as the optimal $\boldsymbol{f}_{M^*}$-cyclic schedule, and is formally defined below.

Definition 4.3.2 (Optimal $\boldsymbol{f}_{M^*}$-cyclic schedule for J-OP (I-OP)) *Let $\mathcal{F}(M^*)$ denote the set of all valid sequences of length up to M^*. The optimal $\boldsymbol{f}_{M^*}$-cyclic schedule for J-OP (I-OP) corresponds to the optimal $\boldsymbol{f}$-cyclic schedule with the lowest J (I) cost among all schedules with sequences $\boldsymbol{f} \in \mathcal{F}(M^*)$.*

Thus, for example, the optimal $\boldsymbol{f}_{10}$-cyclic schedule for I-OP is found by solving (4.22) for *all* valid sequences with 10 or less positions and selecting the one with the lowest I-cost.

4.4 Parameter Selection in CC Class Policies

A scheduling policy can only be as good as its parameter selection procedure. While it is possible to devise a very general policy with many parameters, such a policy would not be very useful if it is not accompanied by a robust procedure for selecting good values for these

[4]This is necessary for the sequence-dependent case because, when we transform the solutions, the sequence of changeovers will be reversed.

parameters.

The CC Class policies described in Chapter 2 have different levels of complexity. For a fixed base stock level $\boldsymbol{Z}^{\mathrm{U}}$, the CLB policy is the least complex, since it has no parameters and thus can be readily implemented without any previous computations. Then, in order of increasing complexity, PKP requires a single parameter per part type, LOP requires an additional value for its cruising parameter, and the HZP requires in addition to the parameters in LOP a rank ordering or priority assignment. In this section, we describe and motivate the methods used to select the parameters in these policies. As will become evident, for both J-OP and I-OP, the methods described here *are independent of the base stock level* $\boldsymbol{Z}^{\mathrm{U}}$, whose selection is discussed in Section 4.4.3.

4.4.1 Parameter Selection in PKP and LOP

J-OP

We focus first on the parameter-selection procedure for problems in which performance is measured through cost J. After deriving their lower bound for J-OP, Perkins and Kumar (1989) suggested the use of the peak ideal deviations y_i^* given by (4.16) as a parameter for making changeover decisions. Their approach consists of selecting the part type i whose ratio of surplus deviation to y_i^* *after* the setup change is largest. (Note that this concept is similar to the heuristic used for finding a good sequence in the ELSP, where the ideal production frequencies n_i^* are used for determining $\boldsymbol{f}$.)

In Perkins and Kumar's original work, cruising and setup costs were not incorporated into their model, and they ignored the balance constraints (4.14). Under these conditions, they showed that J_{LB} can be found analytically and that the ideal peak deviations satisfy

$$y_i^* \propto \frac{\sqrt{d_i(1-\rho_i)}}{\sqrt{c_i}}. \tag{4.23}$$

Lan and Olsen (2006) provided an analytical solution for J_{LB} with cruising and setup costs (no balance constraints). For systems with no cruising, their solution is given by[5]

$$y_i^* = \frac{\sqrt{2d_i(1-\rho_i)(\beta S_i + K_i)}}{\sqrt{c_i}}, \tag{4.24}$$

where β solves

$$\sum_{i=1}^{N} S_i \sqrt{\frac{c_i d_i(1-\rho_i)}{2(\beta S_i + K_i)}} = (1-\rho).$$

In their original papers, the above expressions for y_i^* were substituted into the statements of the PKP and LOP policies (Policies 2.6 and 2.7 in Chapter 2), to obtain a closed-form expression for the changeover rule. In our implementation of LOP, rather than using (4.24), we compute y_i^* through the numerical solution of the tighter bound J_{LB} given by (4.15).

From their solution to J_{LB}, Lan and Olsen also obtained an analytical condition that determines if cruising is *ideal* for a given system. When this condition is met, $p_i^{\mathrm{c}*} > 1$ for at least one i, and the cruising parameter r_i^* should be 1 (or close to 1).

I-OP

As mentioned in Chapters 1 and 2, CLB, PKP, and LOP were originally stated as make-to-order policies, in which cost J is the natural metric to consider for performance evaluation. However, the policies can be also implemented in a make-to-stock formulation, which implies

[5] Refer to Lan and Olsen (2006), page 514, for the cruising solution. Also, note that (4.24) is expressed using *our notation*, which differs slightly from the one used in that manuscript; in particular, the authors' cost c_i actually corresponds to $c_i\mu_i$ in our notation.

that they behave as exhaustive, base stock policies.

In our experiments, the procedure we adopted for selecting the parameters of LOP for problems in which cost I was the objective differs only on the fact that the ideal deviations y_i^* now come from the solution of I_{LB}, given by (4.18).

4.4.2 Parameter Selection in the HZP

Hedging Zone and Cruising Parameter for J-OP and I-OP

Following the intuition behind PKP and LOP, the selection of the hedging zone parameters in the HZP also relies on the ideal peak surplus deviations y_i^*, computed from the solution to J_{LB} (for J-OP) and I_{LB} (for I-OP). Namely, we select the thresholds $\boldsymbol{\Delta Z}$ so that

$$\Delta Z_i = y_i^* - S_i d_i, \tag{4.25}$$

which means that ΔZ_i represents the ideal type i deviation *before* the changeover takes place.

Note that with ΔZ_i given by (4.25), the changeover rules are very similar *inside* the hedging zone for HZP and LOP. More specifically, consider a non-cruising trajectory that in the long-run always stays inside the hedging zone. In such trajectory, the HZP will select changeovers based on the ratio

$$\frac{Z_i^U - x_i}{y_i^* - S_i d_i}$$

while LOP relies on the ratio

$$\frac{Z_i^U - x_i + S_i d_i}{y_i^*}$$

We have no reason to believe that any of the two ratios gives a consistent advantage over the other and thus most of the performance differences between HZP and LOP are attributed to the shape of the setup zones *outside* the hedging zone. (Recall that, in the HZP, priorities are used outside of the hedging zone, while LOP implements the same changeover rule over all the surplus space.)

With respect to the cruising parameter r_i^*, we use Lan and Olsen's procedure. That is, if $p_i^{c*} > 0$ for at least one i, the cruising version of the HZP is used by selecting $r_i^* = 1$ (or close to 1); otherwise, the non-cruising version is used. It should be noted that in systems for which cruising is ideal, the HZP (as well as LOP) will cruise at the base stock level instead of at 0, as the ideal trajectory prescribes. This could make cruising less attractive for the system, at least in the deterministic case. Since in our experiments we will only focus on non-cruising systems, this issue will not be a problem in this thesis and needs to be addressed in the future.

Prioritization for J-OP

It is well-known in the scheduling and queuing theory literature that, in many settings, the so-called $c\mu$ Rule for part-type prioritization is optimal (see, e.g., Meyn 2008, p. 33, or Duenyas and van Oyen (1996)). For completeness, we show in Appendix B that, in our deterministic model with *no setups*, the recovery costs C_J are minimized when the part type with the largest value of $c_i\mu_i$ and with $y_i(t) > 0$ is produced.

It is important to highlight that the policy resulting from the $c\mu$ Rule for the case with no setups is *not* in the CC Class (see Appendix B). Thus, even as we let $S_{ij} \to 0$ in our system and decrease the size of the hedging zone, the HZP will not approach this optimal policy; this is due to the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule, which states that changeovers only occur after reaching the base stock level. However, it is reasonable to expect that, as setups become negligible, selecting changeovers based on the $c_i\mu_i$ coefficients is a good heuristic even under the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule restriction (e.g., when $\boldsymbol{d} = \boldsymbol{0}$ and there are no setups, the optimal policy for clearing an initial deviation follows both the $c\mu$ Rule and the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule). Furthermore, we can also expect that for the general case with positive setup times, a prioritization based on the $c\mu$ Rule will be effective for making changeover decisions for large values of $\boldsymbol{Z}^{\mathrm{U}} - \boldsymbol{x}$. This follows because the contribution of the setup change to the duration and total cost of the run will be relatively small over these long runs. These statements motivate the use of the $c\mu$ rule for prioritizing part types outside of the hedging zone.

Prioritization for I-OP

The prioritization assignment for make-to-stock problems requires some further consideration. As Perez and Zipkin (1997) point out, when the system is in the backlog area (i.e., $\boldsymbol{x}(t) < \mathbf{0}$), the item with the highest $b_i\mu_i$ coefficient maximizes the rate of decrease of the instantaneous cost $\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}^{+}(t) + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}^{-}(t)$. On the other hand, when the system is in the inventory area (i.e., $\boldsymbol{x} > \mathbf{0}$), the item with the highest $h_i\mu_i$ coefficient maximizes the rate of increase of inventory costs. However, these facts do not necessarily imply that a rank ordering based on the $h_i\mu_i$ coefficients will perform poorly because the system needs to build up inventory in order to satisfy the demand during periods when it is producing other items (or, in stochastic systems, during downtime periods). Suppose, for example, that item i has a large h_i cost. In the HZP, items with high priority will tend to have trajectories that are closer to their base stock level, compared to other part types. Thus, as in our arguments related to Fig. 4-1, giving high priority to item i could allow the system to achieve a desired service level with a smaller value of Z_i^{U}, which in turn should translate into lower inventory holding costs for that item.

More work and experimentation is needed to identify the best prioritization scheme for I-OP. In this thesis, we simplify matters somewhat by focusing exclusively on systems in which the ratio b_i/h_i is the same for all $i \in \mathcal{Q}$. (Because of (4.3), this is equivalent to considering systems in which the same target service level has been set for all items, which may be a realistic assumption for many practical applications.) Under this restriction, we do not need to decide between following a $h\mu$-Rule or a $b\mu$-Rule since the rank ordering given by the $h_i\mu_i$ coefficients is the same as the ordering based on $b_i\mu_i$. Of course it may be that, in general, a rank ordering in which the priority $\mathsf{P}(i)$ is some function of both h_i and b_i performs better (we could even consider functions that depend on Z_i^{U} as well). We propose in Section 6.1.3 two such rank orderings that should be investigated; these schemes have the attractive feature that they reduce to the $h\mu$-Rule (or, equivalently, the $b\mu$-Rule) when all items have the same ratio b_i/h_i.

Summary of HZP Parameter Selection

To summarize, the HZP makes changeover decisions according to two criteria: (1) when deviations are far from their ideal surplus deviation, it selects the part type with the largest $c_i\mu_i$ coefficient (for J-OP) and (2) when all surpluses are close to their ideal values (i.e., inside the hedging zone), it attempts to match the deviations to their ideal values.

The steps for setting the HZP parameters are summarized in the box below. Note that these steps assume sequence-independent setups and the same ratio b_i/h_i for all items. Refer to Chapter 6 for suggestions about how to deal with more general cases.

4.4.3 Base Stock Level Selection

The policies' parameter selection procedure for I-OP described in the previous sections are independent of the base stock level $\boldsymbol{Z}^{\mathrm{U}}$. This important fact highlights the approach that we have taken for solving I-OP, where we simplify the problem by splitting it into two parts. First, we generate good trajectories in terms of their production frequencies and, second, we adjust the location of these trajectories in surplus space in order to minimize inventory and backlog costs.

The selection of the base stock level $\boldsymbol{Z}^{\mathrm{U}}$ in the CC Class policies consists then of an iterative

HZP $\boldsymbol{\Delta Z}$ and $\mathsf{P}(\cdot)$ Selection Procedure for J-OP(I-OP)

1. Compute J_{LB} (I_{LB}) and obtain the ideal production frequencies n_i^* and ideal cruising fractions $p_i^{\text{c}*}$. Determine also the ideal peak surplus deviations y_i^*, using (4.16).

2. For each $i \in \mathcal{Q}$, set $\Delta Z_i = y_i^* - S_i d_i$.

3. If $p_i^{\text{c}*} > 0$ for some $i \in \mathcal{Q}$, use the cruising version (i.e., set $r^* = 1$). Otherwise, use the non-cruising version.

4. Prioritize part types according to the $c\mu$ ($b\mu$ or $h\mu$) Rule. That is, set

$$\mathsf{P}(i) = \begin{cases} c_i\mu_i & \text{for } J\text{-OP} \\ b_i\mu_i & \text{for } I\text{-OP} \end{cases}$$

procedure that requires observation of the actual system (or simulation experiments). Relying on (4.3), we adjust the base stock level until the optimal service level for each part type is matched, and this gives us the value of $\boldsymbol{Z}^{\text{U}}$ that minimizes I. (Notice that this procedure is somewhat representative of what a manager would actually do in a real setting, where he would raise or lower the base stocks values based on past performance in order to meet the desired service levels.)

4.5 Gallego's Recovery Policy

In Chapter 2, we outlined the policy proposed by Gallego (1990), which we called Gallego's Recovery Policy (GRP). This heuristic constitutes an interesting benchmark for comparison because it is not in the CC Class. In general, production runs in GRP do not conclude at the same surplus level every time and, furthermore, when recovering from a large disruption, runs will generally be shorter than in CC Class policies (which need to completely clear the current setup's deviation before changing setups).

Gallego's policy is based on the solution to the EELSP through an optimal $\boldsymbol{f}$-cyclic schedule. In order to apply the policy, a (presumably good) solution to this problem is required, and the idea consists of adjusting in real time the production times prescribed by the solution based on the departures from the schedule. Thus, one of the key features of GRP is that it always follows the same sequence $\boldsymbol{f}$, even when recovering from disruptions. This fact simplifies the analysis of the dynamics considerably and allowed Gallego to show that, under some restrictions, the optimal recovery policy from a *single* disruption can be found from the solution to a time-invariant Linear Quadratic Regulator (LQR) problem.

We will now state the steps involved in GRP. While the derivation of the LQR expressions that lead to GRP can be found in Gallego (1990), we include all the equations needed to implement the policy in Appendix D. The original work by Gallego did not consider cruising (but it did allow for idling) and, therefore, we will only consider the non-cruising version of GRP in this thesis. This corresponds to cases in which the pre-determined $\boldsymbol{f}$-cyclic schedule has $\Delta t_n^{\mathrm{c}} = 0$ at all positions $n = 1, 2, \ldots, M$.

Consider an f-cyclic optimal schedule with M positions and defined by its sprinting times

$$\boldsymbol{t} = [t^1, t^2, \ldots, t^M]$$

and its surplus level $\boldsymbol{w}$ at the instant *before* the changeover into item f^1. (Note that, according to our notation from Section 4.3.2, $t^n = \Delta t_n^{\mathrm{s,B}} + \Delta t_n^{\mathrm{s,I}}$.) GRP makes control decisions whenever it is about to start a new cycle of the production sequence, and the control consists of adjusted sprinting times for the runs on each position in the sequence.

Using Gallego's terminology, we define the k-th *control cycle* as the period of time comprising the k-th repetition of sequence $\boldsymbol{f}$. At the beginning of each control cycle, the policy reviews the state and, based on how far the current surplus is from $\boldsymbol{w}$, a modified sprinting times vector is obtained and followed during that cycle. The steps are summarized as Policy 4.1.

Policy 4.1: Gallego's Recovery Policy (GRP). Non-Cruising Version

For a non-cruising optimal $\boldsymbol{f}$-cyclic schedule with M positions, let $\boldsymbol{t}$ be the $M \times 1$ vector of sprinting times and $\boldsymbol{w}$ the $N \times 1$ vector of surplus levels at the instant before the changeover into type f^1. Also, let $\boldsymbol{G}$ be the matrix given by (D.2) in Section D.1.

Suppose the system is about to start the k-th control cycle with surplus level $\boldsymbol{x}_{k-1}$. Then, follow these steps:

1. *Determine the current deviation from the ideal surplus levels*

$$\boldsymbol{z}_{k-1} = \boldsymbol{w} - \boldsymbol{x}_{k-1}.$$

2. *Determine the control vector* $\boldsymbol{v}_k = \left(v_k^1, v_k^2, \ldots, v_k^M\right)^{\mathrm{T}}$, *where*

$$\boldsymbol{v}_k = \boldsymbol{G}\boldsymbol{z}_{k-1}.$$

3. *For* $n = 1, 2, \ldots, M$, *if* $t^n + v_k^n > 0$, *switch to part type* f^n *and produce it at full capacity for* $t^n + v_k^n$ *time units.*

4. *Set* $k \leftarrow k + 1$ *and go to Step 1.*

Note that GRP has two offline computational burdens. First, a good $\boldsymbol{f}$-cyclic solution to the EELSP must be found. Second, in order to obtain matrix $\boldsymbol{G}$, it is necessary to solve the Algebraic Matrix Riccati Equation, with enough accuracy to avoid numerical instabilities. On the other hand, for a single disruption $\boldsymbol{x}_0$, it can be shown that the policy always converges to the $\boldsymbol{f}$-cyclic schedule. Moreover, if the disruption is such that $t^n + v_k^n > 0$ always, then the policy is optimal in terms of C_I with respect to any other recovery strategy that follows the same sequence $\boldsymbol{f}$.

4.6 Simulation Experiments

This section describes in detail the set of simulation experiments that were carried out to compare the performance of the policies. Rather than choosing to simulate very complex systems, we decided to limit the dimensionality of the problem so that we were able to study a wide variety of cases and draw more meaningful conclusions about our results. For this reason, the experimental results in this chapter are limited to systems with

- Only three part types ($N = 3$).
- No setup costs.
- Sequence-independent setup times.
- Same ratio of backlog cost to inventory costs (b_i/h_i) for all items.
- No cruising (i.e., $p_i^{c*} = 0$).

As will be seen in the next section (and also in Chapter 5), even with these limitations, we are still able to make very interesting observations about the policies' performance. These results should also serve as a guide for exploring a larger space of system parameters, as discussed in Section 6.1.2.

4.6.1 Experimental Setup

Consider cost measure J. After limiting the system's parameter space, we have that this cost can be expressed as

$$J = g_J(\rho_1, \rho_2, \rho_3, c_1\mu_1, c_2\mu_2, c_2\mu_3, S_1, S_2, S_3),$$

for some function g_J. Notice that, in the above expression, all units of material have been transformed into monetary units through the cost coefficients c_i.

We can restrict the parameter space of the problem even further by making the g_J non-dimensional. For this purpose, we consider the parameters $S = S_1 + S_2 + S_3$ and $c\mu = c_1\mu_1 + c_2\mu_2 + c_3\mu_3$. Defining the non-dimensional cost $\tilde{J}$ as[6]

$$\tilde{J} = \frac{1-\rho}{c\mu S} J,$$

we can then write

$$\tilde{J} = \tilde{g}_J \left(\rho, \frac{\rho_1}{\rho}, \frac{\rho_2}{\rho}, \frac{\rho_3}{\rho}, \frac{c_1\mu_1}{c\mu}, \frac{c_2\mu_2}{c\mu}, \frac{c_3\mu_3}{c\mu}, \frac{S_1}{S}, \frac{S_2}{S}, \frac{S_3}{S} \right). \tag{4.26}$$

Each parameter in $\tilde{g}_J$ lies in the range $(0,1)$ and each triad of related parameters (i.e., the triads ρ_i/ρ, $c_i\mu_i/c\mu$, or S_i/S) must add up to 1. Therefore, it follows that for a fixed value of ρ, the parameter space is given by the cross product of three triangular planes, where each triangle corresponds to the set of possible values for a triad (see Fig. 4-11).

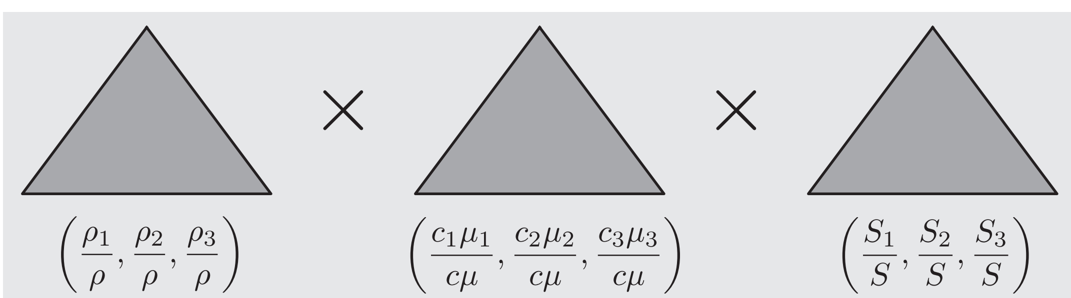


Figure 4-11: Depiction of the parameter space for our experiments. Since all the parameters are in $(0,1)$ and each triad adds up to 1, the set of possible values for each of these triads corresponds to a triangle with sides of length $\sqrt{2}$.

In our experiments, we performed a full-factorial exploration of the parameter space of $\tilde{g}_J$,

[6]The term $(1-\rho)$ is included for convenience.

using the point distribution patterns depicted in Fig. 4-12. The first data set (Dataset 1, Table D.3) had a distribution with moderate dispersion, in which the maximum ratio between any two related parameters in the triad was 7, while the second dataset (Dataset 2, Table D.4) had a higher dispersion, with a maximum ratio of 14. Notice that points lying close to the corners of a triangle correspond to cases in which one of the parameters from the triad is 7 or 14 times greater than the other two. Also, of all 7^3 possible systems that can be formed with this scheme, only 71 are unique; the rest correspond to equivalent systems obtained through permutations of the part type's labels. Only unique systems were simulated in our experiments.

We note that an alternate approach that is typically used in the literature for testing heuristics in systems with many parameters consists of assuming some probability model on the values of the parameters (usually in the form of independent uniform distributions) and sampling system instances according to this model (see, e.g., Dobson 1987, Gallego et al. 1994, or Moon and Silver 2002). The advantage of such approach is that, if the probability model is realistic, one may draw better generalizations from the results of the experiments. However, it is not clear how to define a realistic model for the parameters (Simchi-Levi et al. 2005, p. 9), and the correlations between them might be poorly represented. The full-factorial exploration that we propose here allows us to explore the relatively low-dimensional parameter space of our system systematically. Furthermore, as discussed in Section 6.1.2, it *may* still be possible to discover patterns and draw generalizations from the simulation results that are applicable to larger systems.[7]

For the evaluation of cost I, there is an extra parameter that consists of the ratio of inventory and backlog costs (assumed to be the same for all part types). Since this ratio can be related

[7]Such a generalization could consist, for example, of a function that predicts the best policy to use based on the values of the parameters of the system.

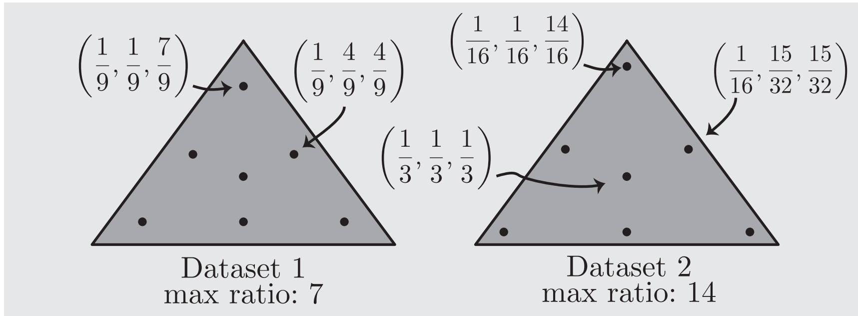


Figure 4-12: Depiction of the points explored for each triad. In Dataset 1, the maximum ratio between any two parameters was 7, while in Dataset 2 it was 14. The actual values can be found in Section D.2.

to the service level θ through (4.3), we can write that

$$\tilde{I} = \tilde{g}_I\left(\rho, \theta, \frac{\rho_1}{\rho}, \frac{\rho_2}{\rho}, \frac{\rho_3}{\rho}, \frac{h_1\mu_1}{h\mu}, \frac{h_2\mu_2}{h\mu}, \frac{h_3\mu_3}{h\mu}, \frac{S_1}{S}, \frac{S_2}{S}, \frac{S_3}{S}\right), \tag{4.27}$$

where

$$\tilde{I} = \frac{1-\rho}{h\mu S} I,$$

$h = h_1 + h_2 + h_3$, and $h\mu = h_1\mu_1 + h_2\mu_2 + h_3\mu_3$. Thus, for a fixed value of ρ and θ, we can explore the parameter space of $\tilde{g}_I$ in the same way as with $\tilde{g}_J$. Similarly, we define the non-dimensional recovery costs

$$C_{\tilde{J}} = \frac{1-\rho}{c\mu S^2} C_J$$

and

$$C_{\tilde{I}} = \frac{1-\rho}{h\mu S^2} C_I.$$

Our experiments consisted on simulating each unique system instance and comparing the costs performance of CLB, LOP, HZP, and GRP. The simulations were of the discrete-event type and were programmed in Fortran 95. The simulation code was wrapped with Python 2.6 with the aid of the scientific library SciPy (Jones et al. 2001). For LOP and HZP, the values of y_i^* were obtained through the numerical solution of J_{LB} and I_{LB} as given by (4.15) and (4.18), respectively, using AMPL (Fourer et al. 2003) and the MINOS Solver 5.1 to solve all nonlinear programs. For each system instance, we obtained the optimal f_{10} sequence by doing an exhaustive search over all sequences of length 10 or less and solving (4.22). This sequence was then used for implementing GRP, with matrix $\boldsymbol{G}$ was computed as described in Section D.1. Since, in general, GRP produced trajectories that were not of the CC Class type (i.e., they did not satisfy the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule), the J costs of this policy were not evaluated. (This does not mean, of course, that GRP could not be adapted to J-OP if desired.)

All averages in the simulations were taken after discarding transients from the trajectories. To ensure that the averages computed for each instance were in steady state, we evaluated cost J on two subsequent periods after transients, and verified that the values of this cost differed by less than 0.1% on both periods. (We were able to achieve this accuracy in all instances by computing the averages during the period $t \in [2.5 \times 10^6, 3.0 \times 10^6]$, for the case $\rho = 0.9$, and $t \in [1.0 \times 10^7, 1.5 \times 10^7]$, for $\rho = 0.99$, with $S = 10$.) The base stock values in all policies were adjusted in order to match the optimal service level (4.3) with an error of less than 5×10^{-4}.

It is interesting to note that, as mentioned in Section 4.1, the systems could converge to different values of J and I depending on the initial conditions. For example, Fig. 4-13 depicts the case of a system operated under LOP in which, depending on $(\boldsymbol{x}_0, \sigma_0)$, the system converged into (at least) one of three different limit cycles. This example shows that the dynamics of the policies can be quite complicated and difficult to predict (see also Fig. 3-4). (The possibility of having multiple limit cycles could also have an effect on the performance of the stochastic system; in particular, it could happen that the steady-state surplus trajectory spends most of the time in one of several sets of states and jumps infrequently between them. Such a behavior will likely increase the variability of the system and make it harder to control effectively.)

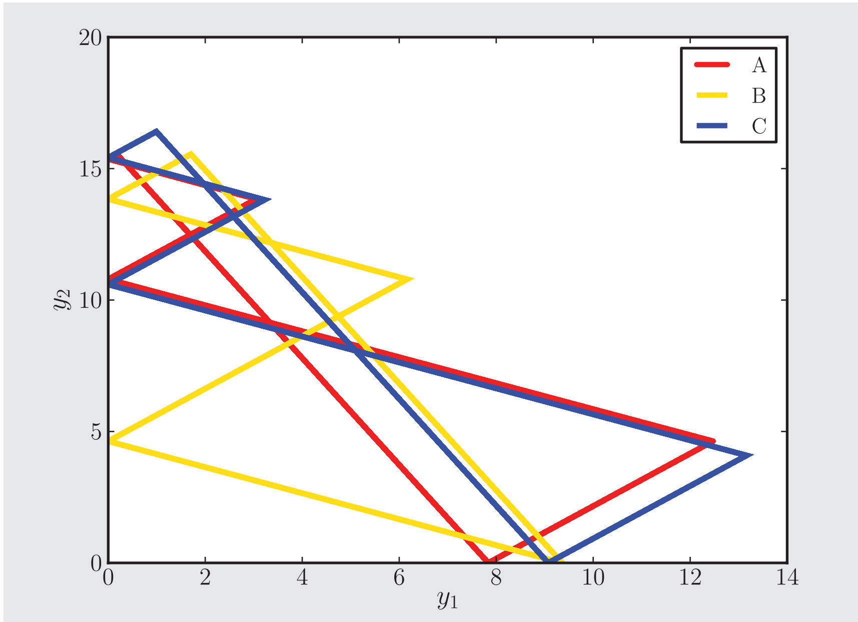


Figure 4-13: Example of a system with multiple limit cycles. The parameters of the system correspond to Instance 6 of Dataset 2 (see Table D.4) with $\rho = 0.99$, $c\mu = 1$, $c_i = 1$, and $S = 10$. The policy used in this case was LOP.

While in theory the definitions of J and I require us to consider *all* initial conditions, as an approximation we simulated each system instance with 6 different initial conditions: for $\sigma_0 = 1, 2, 3$, one set of initial conditions was very close to $\boldsymbol{Z}^{\mathrm{U}}$, namely, $y_i(t_0) = 0.1$, while the other set was far from $\boldsymbol{Z}^{\mathrm{U}}$, with $y_i(t_0) = 500$ for $i = 1, 2, 3$. These initial conditions assume a value for the non-dimensional parameters $c\mu$ and $h\mu$ of 1, and that $c_i = h_i = 1$ for $i = 1, 2, 3$. The maximum cost over these 6 different runs was reported in each case.

4.6.2 Results and Analysis

Cost J

We begin by comparing the J-cost performance of the HZP with respect to CLB, LOP, the optimal $\boldsymbol{f}_{10}$-cyclic schedule, and the lower bound J_{LB}. The results are shown in Fig. 4-14, where each marker corresponds to a different system instance. We include both datasets and two workload values for each, $\rho = 0.9$ and $\rho = 0.99$ (note that the scaling factor $1 - \rho$ in $\tilde{J}$ allows us to condense neatly the results on the same graph). On each plot, the horizontal axis represents the value of $\tilde{J}$ for the policy in question, while the vertical axis shows the percent cost difference obtained by the HZP. The extreme cases on each plot are labelled with a number and the actual system parameters corresponding to these numbers can be found in Section D.2. All points that lie on the negative vertical axis correspond to cases in which the HZP outperformed the policy compared.

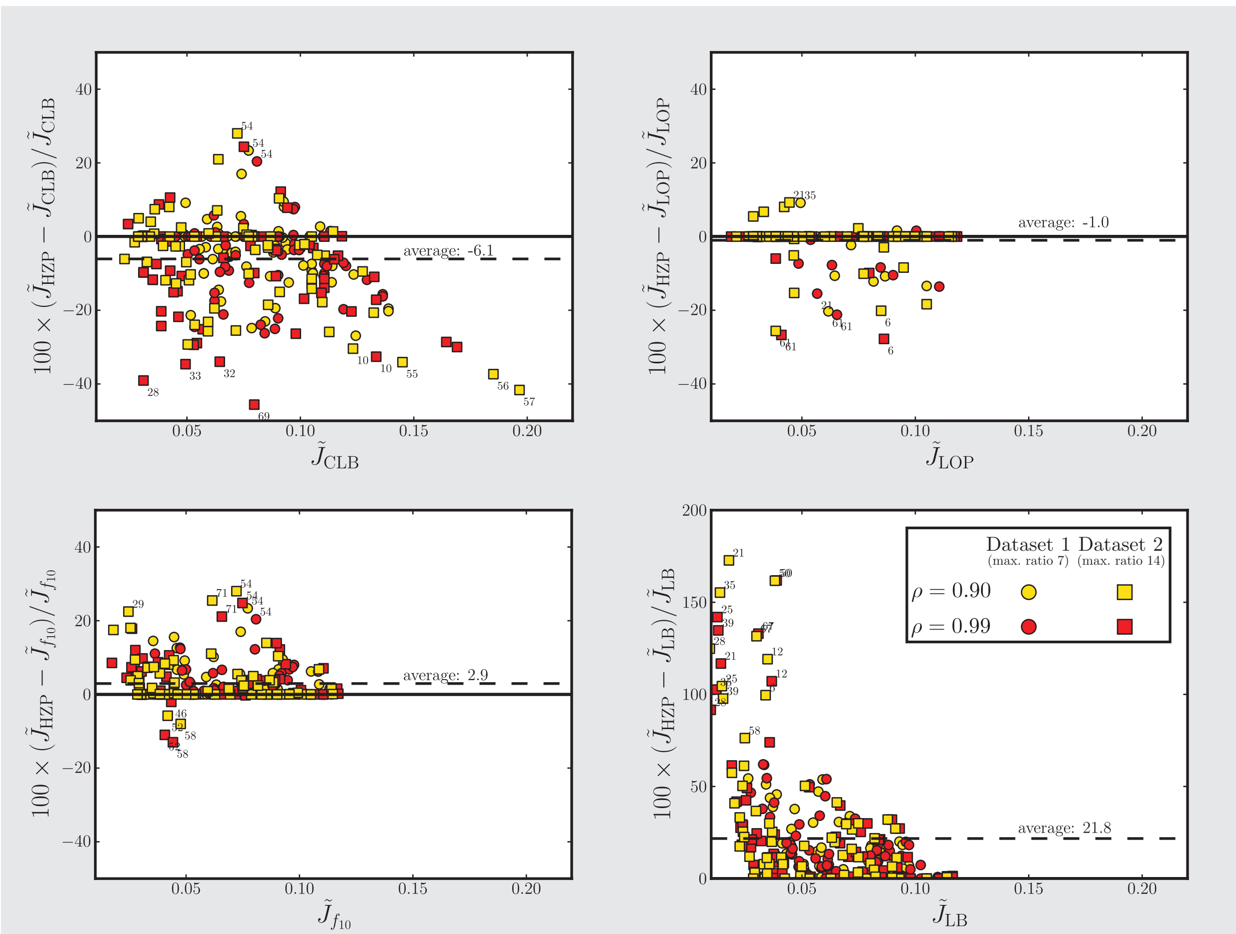


Figure 4-14: Comparisons of the non-dimensional cost $\tilde{J}$ of the HZP with respect to CLB, LOP, $\boldsymbol{f}_{10}$, and the lower bound.

The results of the top two plots in the figure suggest that, in terms of J-OP, the HZP performs overall better than CLB and LOP. As discussed earlier, given that HZP and LOP have very similar setup zones inside the hedging zone, we attribute most of the performance improvements with respect to LOP to the use of the $c\mu$ rule. (It turns out that all of the points labelled in the plot in which HZP outperformed LOP corresponded to cases in which one part type's $c\mu$ coefficient was much larger than the rest. See Section 6.1.2).

Since the optimal $\boldsymbol{f}_{10}$-schedule was found through an exhaustive search over all sequences of length 10 or less, it is reasonable to expect that the cost of this schedule will be close to optimal. Thus, given that the HZP had on average less than 3% extra cost than $\boldsymbol{f}_{10}$ (averaged over all instances), we may conclude that the the HZP generates on average very good solutions to J-OP. It is also interesting to note that in a couple of instances, the HZP even outperformed the optimal $\boldsymbol{f}_{10}$-cyclic sequence. This means that, in those cases, the HZP converged into a lower-cost sequence that contained more than 10 positions.

Finally, the plot in the lower right of Fig. 4-14 compares the performance of the HZP with respect to the lower bound, showing that the difference between the two is in some cases very large. We conclude from this plot that the lower bound can be quite loose; this follows because the HZP never deviated by more than 30% from the cost of $\boldsymbol{f}_{10}$, which, as mentioned, is close to optimal. Thus, although the comparison with J_{LB} suggests that on average the HZP was within 22% of the optimal cost for J-OP, this percentage should be much closer to the 3% suggested by the comparison with $\boldsymbol{f}_{10}$. An important implication of the looseness of J_{LB} is that, in cases where the ideal schedule is too far off from the optimal schedule, the parameter selection procedure based on y_i^* may lead to poor performance.

Cost I

Figure 4-15 shows the comparisons in terms of cost I for $\rho = 0.90$ and 0.99, and a service level of $\theta = 0.95$. Looking at the two upper plots in the figure, we can see that the HZP outperforms CLB in most cases, and either matches or outperforms LOP in almost all. This supports our claim that a good CC Class policy in terms of J-OP is also a good CC Class policy for I-OP.

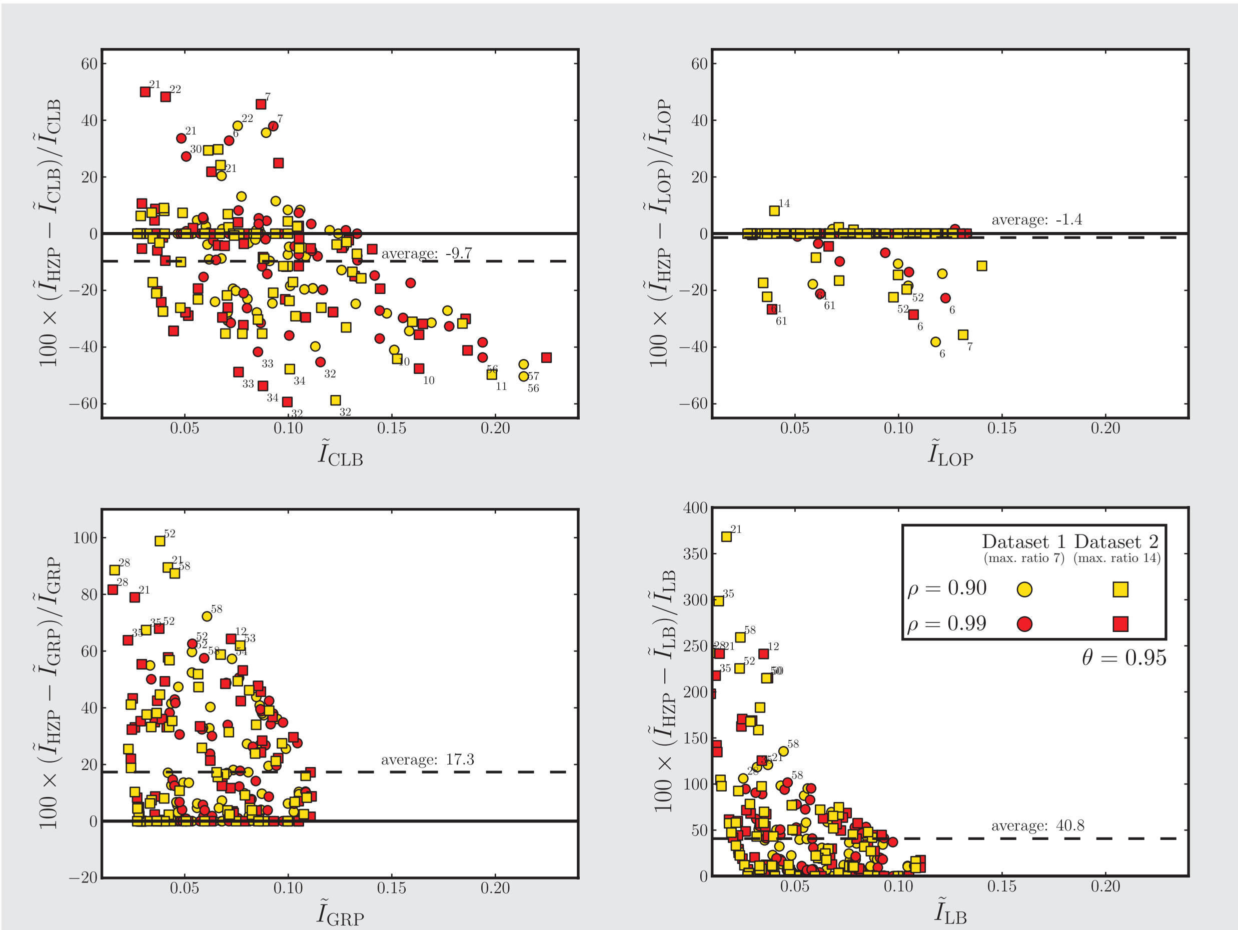


Figure 4-15: Comparisons of the non-dimensional cost $\tilde{I}$ of the HZP with respect to CLB, LOP, GRP, and the lower bound. Service rate was $\theta = 0.95$.

The lower left plot compares the HZP with GRP. Since GRP converges to the close-to-optimal schedule given by $\boldsymbol{f}_{10}$, the comparisons show that the HZP (as well as the other CC Class policies) tend to have poor performance in terms of I-OP. This behavior is not entirely surprising. As we discussed in Sections 4.3.1 and 4.3.2, the Extended Zero-Switch Rule is very close to optimal for I-OP. However, by definition, any CC Class policy will generate trajectories that satisfy the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule, which means that it is likely that the production of some items will begin with inventory still available. Therefore, the structure of CC Class trajectories is better suited for J-OP than I-OP, and the simulation results confirm this fact.

The previous arguments show that a price is paid for the simplicity of having a fixed surplus target in the CC Class. However, two important points must be brought into consideration. First, we have been able to find very good or optimal target sequences for GRP through the exhaustive-search procedure. For larger systems, the quality of the sequences will depend on the heuristic employed, and thus GRP may not perform as well. More importantly, as stressed throughout this thesis, our goal is not to optimize J or I in a purely deterministic setting—which is unrealistic—but on a setting in which the future is not perfectly predictable. We will see in the next chapter that, once we incorporate randomness into the model, the HZP becomes competitive for solving I-OP. (As a side comment, note that if we were truly interested in solving the deterministic ESLP, the simulation results for J-OP suggest an interesting approach for obtaining good open-loop solutions. This approach consists of running the HZP until it reaches steady-state, and then applying the transformation of Fig. 4-9. If the J costs of the original trajectory are good, then the transformed solution will also have good I costs.)

We conclude by pointing out two interesting facts about GRP. As seen in Fig. 4-15, in several cases the costs of HZP and GRP coincided. This implies that, for those systems, GRP generated CC Class trajectories. (Gallego 1994 has identified conditions in which this occurs.) Another interesting observation arises when we compare the I costs of GRP with those of the optimal $\boldsymbol{f}_{10}$ sequence, as shown in Fig. 4-16. We stated earlier that GRP converges to the predetermined schedule in the absence of disruptions, but the figure shows that the costs are not always exactly the same. The reason for this discrepancy comes from the fact that the formulation for obtaining $\boldsymbol{f}_{10}$ forces the solution to follow the EZSR. However, when this rule is not optimal, the solution will not necessarily satisfy the optimal service level relation (4.3). On the other hand, when we implemented GRP, the steady-state surplus trajectory was always adjusted so as to match the optimal service level, and this led to a slightly lower cost on those cases. Nevertheless, it is reassuring to see that the difference in costs is very small (less than 2% in the worst case), which suggests that there is not a significant cost penalty for following the EZSR in I-OP, or the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule in J-OP.

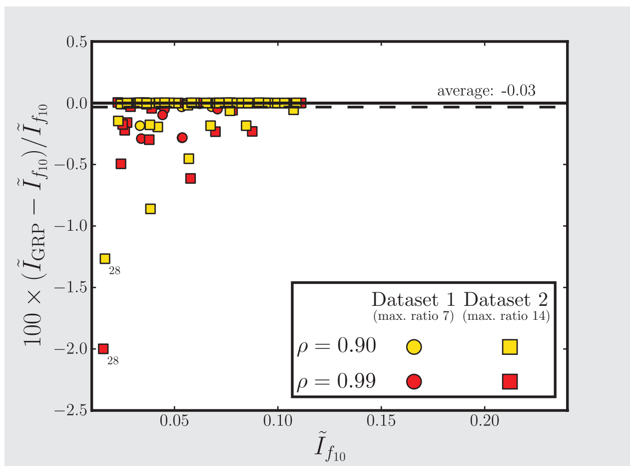


Figure 4-16: Comparison of I costs for GRP and the $\boldsymbol{f}_{10}$ optimal sequence. The small differences correspond to cases in which the EZSR was not optimal.

Recovery Costs

Figures 4-17 and 4-18 show the performance of the policies in terms of recovery costs, for the initial conditions $\boldsymbol{x}_0 = -(500, 500, 500)^{\mathrm{T}}$, $\sigma_0 = 1$ (with $c\mu = h\mu = 1$ and $c_i = h_i = 1$, $i = 1, 2, 3$) and $\boldsymbol{Z}^{\mathrm{U}}$ adjusted for a long-term service level of $\theta = 0.95$. The plots of $C_{\tilde{J}}$ and $C_{\tilde{I}}$ for the CC policies were almost exactly the same, and thus we omit the latter.

The results suggest that the $c\mu$ rule is a good recovery rule for CC Class policies, outperforming LOP and CLB in almost all cases. On the other hand, GRP has the best recovery costs

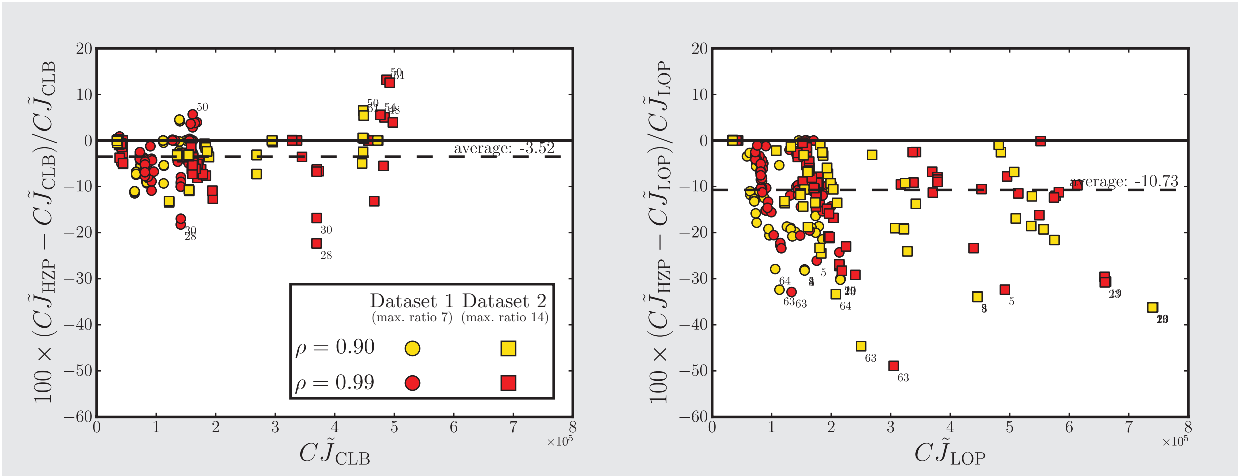


Figure 4-17: Comparison of the recovery costs $C_{\tilde{J}}$ for CLB, LOP, and HZP.

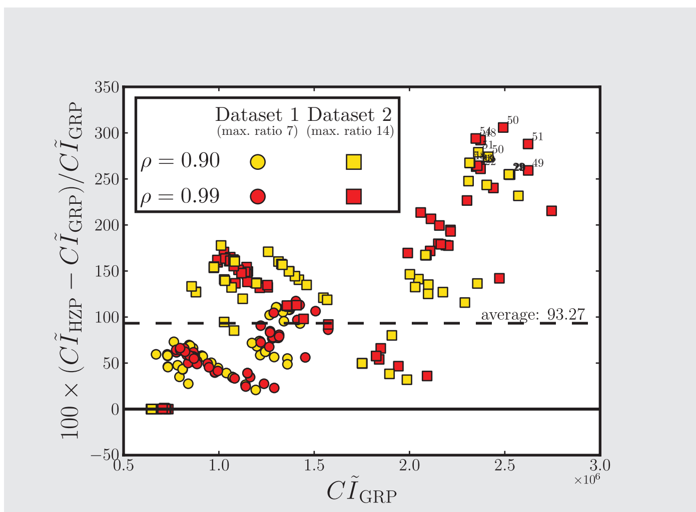


Figure 4-18: Comparison of the recovery costs $C_{\tilde{I}}$ of HZP and GRP. (The corresponding plots for the CC Class policies are very similar to those in Fig. 4-17.)

and performs significantly better than all CC Class policies with respect to $C_{\tilde{I}}$. This shows that, if a good sequence is known, the optimal recovery costs following that sequence will also be good, at least when there are no further disruptions or when these occur relatively infrequently. We will use these observations to interpret our stochastic simulation results of the next chapter.

4.7 Summary

This chapter has covered the performance analysis of our deterministic system, setting the stage for the stochastic analyses to follow. We began by defining two measures of long-term schedule performance: average surplus-deviation cost J and average inventory and backlog costs I. We also defined the recovery costs metric, which is a non steady-state cost that relates to how well a system can recover from a disruption.

We presented the derivation of two lower bounds on our cost measures, and showed how these bounds can be very helpful for selecting good values of the parameters of our policies. We also discussed the relationship between our model and the classical Economic Lot Scheduling Problem, and described in detail Gallego's Recovery Policy, a closed-loop policy that is based on solutions to this problem.

The main contribution of this chapter consisted of the design and implementation of an extensive set of simulation experiments that compared the performance of CLB, LOP, HZP, and GRP in terms of each cost. By restricting the system's parameter space, we were able to explore many cases in a systematic way, drawing more meaningful conclusions about the policies. In particular, we saw that the HZP tends to outperform the other CC Class policies in most cases, both in terms of J and I. These results also agreed with our statement that a good CC policy for J tends to be also a good CC policy for I. On the other hand, given that GRP was able to take advantage of the near-optimal sequence $\boldsymbol{f}_{10}$, this policy showed the best deterministic behavior in terms of cost I.

The deterministic results presented here, although interesting in their own right, will provide us with valuable intuition and help us interpret better the results of the next chapter, in which we introduce random failures into our model.

Chapter 5

Failure-Prone System

In this chapter, we depart from the deterministic world and consider the case in which our machine is subject to random failure and repair times. As we revisit the issues of stability and performance, it will become apparent that our previous deterministic results and analyses constitute a strong foothold for dealing with the more complex formulation.

We begin by formally describing the failure-prone model and adapting the scheduling policies to this model. We then define the *corresponding deterministic model*, which captures roughly the average behavior of the stochastic system by accounting for the long-term fraction of time that the machine is operational. We then show that our stability conditions, when applied to the corresponding deterministic model, guarantee the stability of the stochastic system. On the final section we study the performance of the policies through a series of stochastic simulation experiments similar to those of Chapter 4. The results presented here allow us to gauge the merits of each policy under an environment that is more realistic of the actual conditions in the factory floor.

5.1 Stochastic Model and Policies Formulation

5.1.1 Model Description

Out of the many possible sources of randomness in a manufacturing system, in many settings machine failures can have the most significant effects on the performance of the plant (Law and Kelton 1991, p. 705). For this reason, while we expect our policies to also work well under other disruptions, we focus here exclusively on the issue of machine breakdowns. This means that, other than the effects that these failures can have on the system, the demand, production, and setup processes in our model remain the same as before (i.e., deterministic and with continuous material). It follows from our justifications of Section 2.1.2, and from the hierarchical approach to scheduling of Gershwin (1989), that such a model would be appropriate for time scales in which failures do not occur so frequently that they can be averaged out of the model, or so infrequently that the system appears perfectly reliable over the period of study.

The operational state of the machine is given by the binary state process $\{\,\alpha(t)\ ;\ t \geq 0\,\}$, where $\alpha(t) = 1$ indicates that the machine is able to produce at time t, while $\alpha(t) = 0$ indicates that the machine is down and under repair. To formalize the probability model for $\alpha(t)$, we define the counting processes $\Psi_{\mathrm{F}}(t)$ and $\Psi_{\mathrm{R}}(t)$, which correspond to the total number of failures after t units of machine *production* time and the total number of repairs completed after t units of machine *repair* time, respectively. Let $T_i(t)$ denote the total time allocated for *production* of item i up to t and $R_i(t)$ the total time that the machine has been under repair with setup i. We assume that the machine cannot fail during a changeover and that the probability of failure does not depend on the production rate of the machine (the latter assumption simplifies the mathematics; see, e.g., Gershwin 2002, p. 277). Moreover, failure and repair times are independent. At any time t, the state of the machine will then be given by

$$1 - \alpha(t) = \Psi_{\mathrm{F}}\big(\textstyle\sum_{i=1}^{N} T_i(t)\big) - \Psi_{\mathrm{R}}\big(\textstyle\sum_{i=1}^{N} R_i(t)\big). \tag{5.1}$$

We assume that the machine does not lose its setup after a failure and that changeovers are not allowed during the repair process. That is, if a repair starts at time t_1 and concludes at time t_2, then $\sigma(t_1) = \sigma(t_2)$.

To further simplify our model, we will focus exclusively on the case where the machine uptimes and downtimes are i.i.d. exponentially-distributed random variables and thus $\Psi_{\mathrm{F}}(t)$ and $\Psi_{\mathrm{R}}(t)$ are Poisson counting processes. This assumption is commonly used for modeling unreliable machines (see, e.g., Gershwin (2002)), since the memoryless property of the uptime/downtimes simplifies the analysis considerably. Furthermore, when modeling complex machinery where a failure could be caused by a variety of factors—each with a low probability of occurring—the exponential distribution for machine uptime may be an accurate representation (see the discussion in Gershwin 2002, p. 38). Nevertheless, we expect that the results in this chapter—particularly the performance comparisons—will be robust to other reliability models.

The machine efficiency e is defined as the ratio

$$e = \frac{\mathrm{MTTF}}{\mathrm{MTTF} + \mathrm{MTTR}}, \tag{5.2}$$

where MTTF is the mean time to fail and MTTR is the mean time to repair (note that MTTF is calculated over the machine's *production* time). From the strong law of renewal processes (Gallager 1996, p. 60) we have that, almost surely

$$\lim_{t\to\infty} \frac{\Psi_{\mathrm{F}}(t)}{t} = \frac{1}{\mathrm{MTTF}}$$

and

$$\lim_{t\to\infty}\frac{\Psi_{\mathrm{R}}(t)}{t}=\frac{1}{\mathrm{MTTR}}.$$

Also, under the assumption of i.i.d. uptime and downtimes, we can verify that, with probability one,

$$\lim_{t\to\infty}\frac{\sum_{i=1}^{N}T_i(t)}{\sum_{i=1}^{N}(T_i(t)+R_i(t))}=e.$$

In our stability results of the next section, we will rely on the stronger statement that says that, for all part types $i\in\mathcal{Q}$ and with probability one,

$$\lim_{t\to\infty}\frac{T_i(t)}{T_i(t)+R_i(t)}=e \tag{5.3}$$

The above equation is motivated in Fig. 5-1, where we show a typical realization of the machine operational state $\alpha(t)$. Note that, according to our assumptions, the setup indicator variable $\sigma(t)$ can only change during periods when $\alpha(t)=1$ and $\alpha(t)$ must always be 1 during setup times (shaded gray in the figure). Moreover, due to the memoryless property of the exponential distribution, once a setup change is complete, the time to the next failure is independent of the age of the machine (i.e., the elapsed non-setup time since the last repair). Therefore, if we look at $\alpha(t)$ exclusively over periods when $\sigma(t)=i$, the process will still have exponentially-distributed uptimes and downtimes.

The rest of our modeling assumptions remain unchanged. That is, the cumulative demand process $D_i(t)$ is continuous and satisfies $D_i(t)=d_it$, where d_i is the constant demand rate. The cumulative production process $P_i(t)$ is also continuous and determined by the time integral of the instantaneous production rate $u_i(t)$. At all times t, the production rate must satisfy

$$0\le u_i(t)\le\mu_i\alpha(t)[\![\sigma(t)=i]\!], \tag{5.4}$$

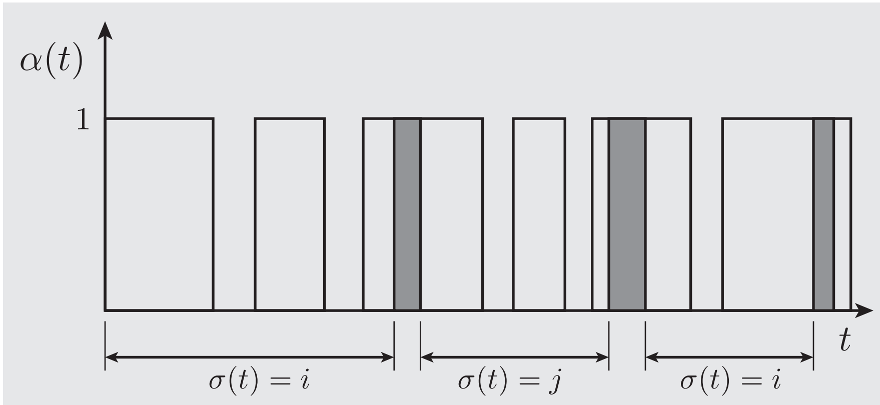


Figure 5-1: Depiction of a typical realization of the machine operational state process $\alpha(t)$ (the gray blocks correspond to setup times). Once a setup change is complete, the time to the next failure is independent of the age of the machine and remains exponentially-distributed.

which states that the machine can only produce type i while it is up and when its current setup is $\sigma(t) = i$. The system's surplus is denoted by $x_i(t)$ and corresponds to the difference between the cumulative production and the cumulative demand at time t. The deterministic setup time and cost for each i-to-j changeover is denoted by S_{ij} and K_{ij}, respectively.

A model satisfying these assumptions is called a *stochastic system* $\hat{\Sigma}$, which we formally define below. (As before, when referring to the closed-loop version of this system operated under some specified policy, we will use the notation $\hat{\Sigma}_\pi$.)

Definition 5.1.1 (Stochastic System $\hat{\Sigma}$) *A stochastic system $\hat{\Sigma}$ is a model instance with parameters $(\boldsymbol{\mu}, \boldsymbol{d}, \boldsymbol{S}, \boldsymbol{K}, MTTF, MTTR)$, where $\boldsymbol{\mu}$ is an $N \times 1$ vector of maximum production rates, $\boldsymbol{d}$ is an $N \times 1$ vector of demand rates, $\boldsymbol{S}$ is an $N \times N$ matrix (or $N \times 1$ vector for the sequence-independent case) of setup times, and $\boldsymbol{K}$ is an $N \times N$ matrix (or $N \times 1$ vector for the sequence-independent case) of setup costs. The system has i.i.d, exponentially-distributed uptime and downtimes, with a mean time to fail over non-setup time of $MTTF$ and mean time to repair $MTTR$. Its dynamics satisfy* (2.1), (2.2), (2.4), (5.1), *and* (5.4).

5.1.2 Corresponding Deterministic System

The fact that, in the long term, the fraction of production time over non-setup time is equal to the machine efficiency e, suggests that the average behavior of the failure-prone system may be captured, at least roughly, by a deterministic model with production rate $e\boldsymbol{\mu}$. This idea leads to the definition of the *corresponding deterministic system*, consisting of a system Σ (defined in Chapter 2) in which the effect of failures is represented by the effective production rate.

Definition 5.1.2 (Corresponding Deterministic System) *For a given stochastic system $\hat{\Sigma}$ with $(\boldsymbol{\mu}, \boldsymbol{d}, \boldsymbol{S}, \boldsymbol{K}, MTTF, MTTR)$, its corresponding deterministic system is the system Σ with $(e\boldsymbol{\mu}, \boldsymbol{d}, \boldsymbol{S}, \boldsymbol{K})$, where e denotes the machine efficiency given by* (5.2)*.*

An example comparing the behavior of a stochastic system $\hat{\Sigma}_\pi$ and its corresponding system Σ_π is shown in Fig. 5-2. Note that the deterministic system provides an estimate of the time it takes the system to reach steady state from an initial condition $\boldsymbol{x}(0)$, as well as the evolution of the system in surplus space. This suggests that our analyses of the previous chapters, which were based on Σ_π, can be leveraged for understanding and scheduling the stochastic model $\hat{\Sigma}_\pi$. However, it must be mentioned that the degree of similarity between the two systems may also be influenced by the scheduling policy. For example, if a policy makes abrupt changes in its schedule after every failure, then these changes will not be captured by the deterministic model and thus it is likely that the correspondence between the two models will be poor. As we discuss next, this is not the case in the policies that we consider in this thesis, and thus the corresponding deterministic model turns out to be a very useful concept.

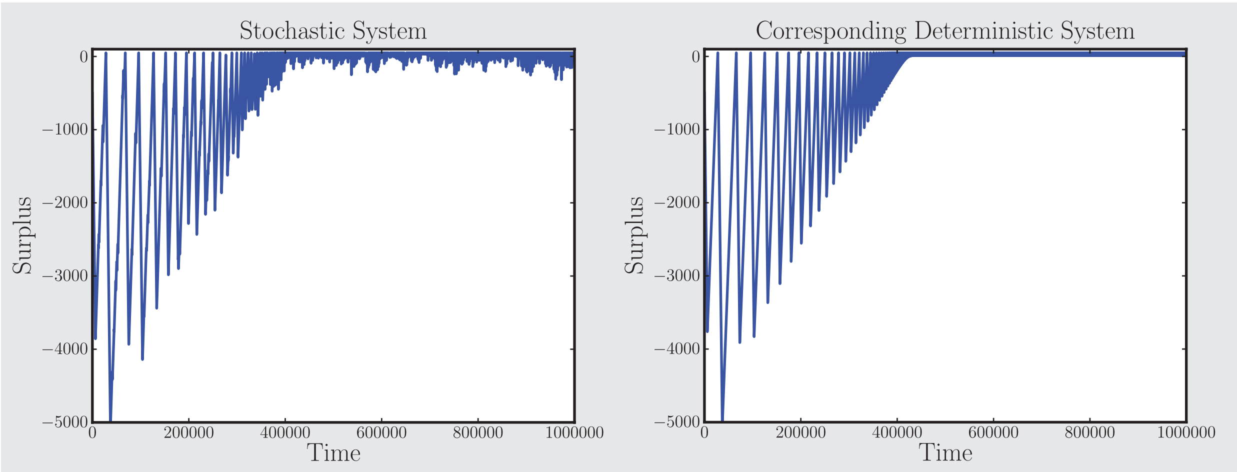


Figure 5-2: Comparison of the surplus versus time trajectories for a stochastic system $\hat{\Sigma}_\pi$ and its corresponding deterministic system Σ_π, under the HZP. (The parameters of the problem correspond to Instance 20 of Dataset 1, Table D.3, with $\rho/e = 0.97$, $S = 10$, MTTF $= 1000$, and MTTF $= 80$.)

5.1.3 Policies Specification in the Failure-Prone System

Perhaps the simplest way to adapt our policies to the failure-prone model is by not altering the changeover decision epochs. Recall that, in the CC Class, setup decisions are only made when the surplus of the current part type reaches its base stock level, a property that we referred to as the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule. While it may be desirable to consider changing setups at other times as well (e.g., after the completion of a repair, particularly if it was a long one), this introduces complexity into the statement, analysis, and parameter selection of the policy. Therefore, consistent with our thesis objective of finding well-behaved and easy-to-implement policies, we decide to follow the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule explicitly, even under the presence of failures.

We can then see that, with the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule enforced, the steps describing each of the CC policies in Chapter 2 apply directly to the stochastic model, as long as we realize that the machine will only be able to produce while it is operational (i.e., while $\alpha(t) = 1$). Thus, in Policies 2.1 through 2.7, we need to add to the control actions "sprint" and "cruise" the condition "whenever the machine is up" to make them applicable to the stochastic system $\hat{\Sigma}$. Also, in order to enforce the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule, whenever the machine recovers from a failure it is understood that we will immediately go back to Step 1 of the policy.

For the case of GRP, our stochastic implementation always follows the target sequence $\boldsymbol{f}$ strictly, which means that we also maintain a close relationship with the corresponding deterministic system. Thus, for each position n in the sequence and each k-th cycle, if $t^n + v_k^n > 0$ the system will change over to part type f^n and allocate $t^n + v_k^n$ time units for production of the part. Note that the wording here is important, since the allocated time $t^n + v_k^n$ also includes possible repair time. Therefore, to be precise, Step 3 in GRP (see Policy 4.1 in Chapter 4) should read

> *For* $n = 1, 2, \ldots, M$, *if* $t^n + v_k^n > 0$, *switch to part type* f^n *and produce it at full capacity whenever the machine is up and until the model time* t *advances by* $t^n + v_k^n$ *time units.*

We can see that, in the CC Class policies, the goal on each run consists of clearing the surplus deviation of the current part type, and the time it takes to do this is random due to possible failures that might occur along the way. In GRP, a fixed amount of time is allocated at the beginning of every control cycle to the production of each item in the sequence, but the actual amount of material produced will be random due to possible lost time during machine repairs. In any case, we can expect that the corresponding closed-loop deterministic system, which considers the average or effective production rate of the machine $e\boldsymbol{\mu}$, will provide a good representation of the average behavior of the stochastic system. For this reason, *the parameters in all policies are selected using the corresponding deterministic model* and the methods discussed in Chapter 4.

5.2 Stochastic Stability of the HZP

There has been a great deal of research in the area of stability of stochastic queues and networks of queues. Initially, this research focused on obtaining the stationary distributions of the systems explicitly, but this is generally hard to do (Bramson 2008). The seminal work by Kumar and Seidman (1990), where it was showed that even in systems with no setup times and adequate capacity there can be instabilities due to material cyclic flows, motivated the search for an alternate approach for ensuring stability. This approach is based on the analysis of deterministic fluid limits.

It has been shown that, under some assumptions about the arrival and service time distributions of the queueing network, the stability of a fluid limit implies the stability of the underlying stochastic process (see Dai 1995, Chen 1995, Stolyar 1995, the recent survey by Bramson 2008, as well as the more accessible introduction to this material by Dai 1999). The stability of the deterministic fluid limit, although still difficult to characterize exactly (see Gamarnik 2002 and our numerical results of Fig. 3-4), can in turn be demonstrated through the use of Lyapunov functions. For example, Kumar and Meyn (1995) developed a stability condition based on a quadratic Lyapunov function. Evaluation of their condition reduced to solving a linear or nonlinear programming problem, but no general analytical solution was obtained. Down and Meyn (1997) used piecewise linear functions that are also constructed through the solution of a linear program. Bertsimas et al. (1996) developed exact conditions for a two-station multi-class system using linear programming duality, and they showed that piecewise linear, convex Lyapunov functions have the power of checking stability exactly for networks with two stations.

Fewer works have studied the stability of queuing networks with setup times and under dynamic-sequence policies. Lou et al. (1992) showed that, when a long-term average capacity constraint is satisfied, the work-in-process of the clear-the-largest-work (CLW) policies (see Section 2.4.1) is recurrent. The authors were only able to verify the boundedness of the expected work-in-process under a strengthened capacity condition. Recently, Dai and Jennings (2004) applied some of the results of the literature on standard queuing networks to develop stability conditions for networks with setup times. This important work was compared with our results in Section 3.3.2.

We will now develop a rigorous verification that the stability conditions of Chapter 3, when applied to the corresponding deterministic system, imply the stability of $\hat{\Sigma}_\pi$ (at least with respect to a specific notion of stochastic stability). While we will only consider the non-cruising HZP in systems with sequence-independent setups and unique priorities, the results should be straightforward to extend to the more general case, as well as to the other CC Class policies discussed in Chapter 2.

5.2.1 Notion of Stability

As in the analysis of deterministic systems, there exist several notions of stability for characterizing stochastic systems. We will only focus on the concept of *rate stability* (also called pathwise stability) which requires that the limiting average production rate of the system matches the demand rate with probability one (Chen 1995, Dai and Jennings 2004).

Definition 5.2.1 (Rate Stable) *A system is defined to be rate stable if, for all $i \in \mathcal{Q}$,*

$$\lim_{t\to\infty} \frac{P_i(t)}{t} = d_i \qquad a.\,s.$$

The very least we expect from a scheduling policy is rate stability since otherwise we would not be able to keep up with demand. However, several other definitions of stochastic stability exist (see, e.g., Dai 1995 and Bielecki and Kumar 1988) and could be considered for ensuring other desirable properties, such as boundedness of the surplus' moments. In any case, the derivations presented here should serve as a first step towards addressing these other notions of stability, which require a more thorough analysis that was beyond the scope of this research.

5.2.2 Overview of Approach

We outline here the general approach that will be followed for verifying the stability of the HZP. We will start by defining a *fluid limit solution* $\breve{\mathbb{X}}(t)$, which consists of a solution to a specific set of deterministic dynamic equations. These equations are not exactly the same equations describing the dynamics of the corresponding deterministic model Σ_π, but they share similar characteristics. One key feature about the equations for the fluid limit solution is that they are constructed in such a way that $\breve{\mathbb{X}}(t)$ will always be stable as long as $\rho < e$.

The next step of the approach consists of considering the closed-loop, HZP-controlled stochastic system $\hat{\Sigma}_\pi$. We state a set of equations that are always satisfied by this model, for all realizations ω of the stochastic process. A solution to this set of equations is denoted by $\hat{\mathbb{X}}(t, \omega)$.

Finally, we bring into the picture the corresponding deterministic system Σ_π (recall that this is the model of Chapter 2 with μ replaced by $e\mu$). We show that when Σ_π satisfies Relaxed Stability Condition 3.3.10, the scaled process $\hat{\mathbb{X}}(rt, \omega)/r$ converges to a fluid limit solution $\breve{\mathbb{X}}(t)$ as r grows. Since $\breve{\mathbb{X}}(t)$ is always stable, this implies then that if Σ_π satisfies our stability condition, the stochastic system $\hat{\Sigma}_\pi$ will be (rate) stable. Thus, the problem of assessing the stochastic stability of $\hat{\Sigma}_\pi$ reduces to the verification of the deterministic stability of Σ_π, which we already know how to do.

Figure 5-3 summarizes the approach and the relationships between the different concepts introduced. In the following sections we develop the above arguments rigorously.

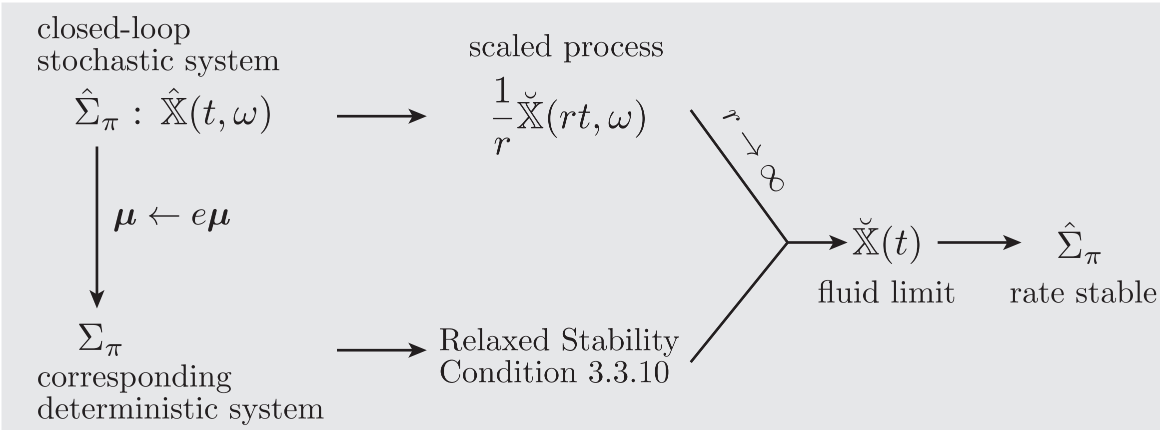


Figure 5-3: Overview of the approach used to prove rate stability of $\hat{\Sigma}_\pi$. When the corresponding deterministic system satisfies the stability condition, the scaled stochastic process converges to a fluid limit that is always stable. This in turn implies that $\hat{\Sigma}_\pi$ is rate stable.

5.2.3 Fluid Limit Model

It is well known in the queueing theory literature that under an appropriate scaling of state and time, a stochastic system may converge into a deterministic *fluid limit model* (Dai 1999). As mentioned, such a fluid limit resembles our deterministic model of Chapter 2, but their mathematical descriptions are not the same. In particular, a stable system Σ_π in our deterministic formulation will always settle down into a bounded limit cycle or attractor, while in the fluid limit the stable system reaches the zero-surplus level. That is, all bounded fluctuations collapse at $\boldsymbol{x}(t) = \mathbf{0}$ in the limit, and we say that the system "drains". Another important characteristic about the fluid limit model is that its solutions are not unique; in fact, with this model, we only seek to obtain a sufficiently accurate picture of the evolution of the system in order to assess its stability. Thus, in some sense, this approach is similar to the Lyapunov theory approach of Chapter 3, where we saw that it is possible to assess the stability of the system by only examining some features of its trajectories (e.g., that runs are V-decreasing), as opposed to considering the complete dynamic evolution of the system.

To verify the stability of $\hat{\Sigma}_\pi$, we will rely on a fluid limit model that captures some unique characteristics about the HZP. For a system operated under the non-cruising HZP and with items arranged so that $\mathsf{P}(1) > \mathsf{P}(2) > \cdots > \mathsf{P}(N)$, we will denote *any* solution to this fluid limit model as

$$\breve{\mathbb{X}}(t) = \big(\breve{\boldsymbol{y}}(t), \breve{\boldsymbol{T}}(t), \breve{\boldsymbol{P}}(t), \breve{\boldsymbol{S}}(t)\big) \geq \mathbf{0},$$

where $\breve{\boldsymbol{y}}(t)$ is an N-dimensional vector of surplus deviations, $\breve{\boldsymbol{T}}(t)$ contains the production allocation times (i.e., the sprinting times) for each item, $\breve{\boldsymbol{P}}(t)$ is the vector of cumulative production, and $\breve{\boldsymbol{S}}(t)$ contains the cumulative setup times into each item. Any fluid limit solution $\breve{\mathbb{X}}(t)$ satisfies (whenever the time derivatives exist)

$$\breve{\boldsymbol{y}}(t) = \breve{\boldsymbol{y}}(0) + \boldsymbol{d}t - \breve{\boldsymbol{P}}(t) \tag{5.5}$$

$$\sum_{i=1}^{N} \Big(\breve{T}_i(t) + \breve{S}_i(t)\Big) = t \tag{5.6}$$

$$\breve{P}_i(t) = e\mu_i \breve{T}_i(t), \text{ for } i \in \mathcal{Q} \tag{5.7}$$

$$\begin{aligned} &\text{If } \dot{\breve{T}}_i(t_1) > 0, \text{ then } \dot{\breve{T}}_i(t) = 1 \text{ on the interval } (t_1, t_2], \\ &\qquad\qquad \text{where } t_2 \text{ is such that } \breve{y}_i(t) > 0 \text{ for } t < t_2 \text{ and } \breve{y}_i(t_2) = 0, \text{ for } i \in \mathcal{Q}. \end{aligned} \tag{5.8}$$

$$\begin{aligned} &\text{If } \dot{\breve{T}}_N(t) = 0, \text{ then } \sum_{i=1}^{N-1} \phi_i \dot{\breve{y}}_i(t) < 0 \text{ for some } \phi_i \text{ such that} \\ &\qquad\qquad \phi_i > 0 \text{ and } \phi_i - \tau_i/e \textstyle\sum_{j=1}^{N-1} \phi_j d_j > 0,\ i = 1, 2, \ldots, N-1. \end{aligned} \tag{5.9}$$

$$\text{If } \breve{y}_N(t) = 0 \text{ and } \sum_{i=1}^{N-1} \breve{y}_i(t) > 0, \text{ then } \dot{\breve{T}}_N(t) \text{ jumps to 0.} \tag{5.10}$$

Equation (5.5) follows directly from the definition of the surplus deviation and is essentially a flow balance relation. (Notice that we could have also stated our model in terms of surplus, but we decided to use surplus deviations here to keep our derivations more in line with the queueing networks literature.) In (5.6), we state that the system can only be either producing (in this case, sprinting) or changing setups (i.e., no failures are included in the fluid limit model). Equation (5.7) states that item i is always produced at rate $e\mu_i$ during the allocated time for that item. Equation (5.8) ensures that each production run is clearing. Note that if at some time t_1 part i is currently under production, then $\dot{\breve{T}}_i(t_1) > 0$. Therefore, the equation states that the system must be allocating all available capacity to type i (i.e., $\dot{\breve{T}}_i(t) = 1$) until its surplus deviation is cleared, which occurs at a later time t_2. Equation (5.9) states that whenever part type N is *not* being produced, then the weighted sum of the derivatives of the first $N-1$ deviations must be decreasing. This weighted sum corresponds to a Lyapunov function that satisfies (3.11) for the reduced, $(N-1)$-part-type system (with τ_j replaced by τ_j/e), and thus the equation states that there must exist such a function with negative derivative over all intervals where type N is not produced. Finally, (5.10) states that production allocation of the lowest-priority part type N must stop if its deviation has been cleared and there are higher-priority items with a positive deviation.

The following lemma shows that any solution $\breve{\mathbb{X}}(t)$ to the above equations must be stable.

Lemma 5.2.2 (Stability of $\breve{\mathbb{X}}$) *If $\rho < e$, then any solution $\breve{\mathbb{X}}(t)$ must drain (reach $\breve{\boldsymbol{y}}(t) = \mathbf{0}$) in finite time.*

Proof: *To prove the lemma, we will first show that any solution $\breve{\mathbb{X}}(t)$ must be work-conserving or non-idling (Chen and Yao 2001, p. 127), which means that the production allocation process increases at the maximum possible rate (i.e., rate of 1) whenever $\breve{\boldsymbol{y}}(t) \neq \mathbf{0}$. Mathematically,*

$$\sum_{i=1}^{N} \dot{\breve{T}}_i(t) = 1 \quad \textit{if} \quad \sum_{i=1}^{N} \breve{y}_i(t) > 0 \ ,$$

whenever the time derivatives exist.

Suppose $\dot{\breve{T}}_N(t) = 0$ at some time t. Then, (5.8) *and* (5.9) *require that $\dot{\breve{T}}_j(t) = 1$ for some $j < N$ with $\breve{y}_j(t) > 0$. (This is true because $\sum_{i=1}^{N-1} \phi_i \dot{\breve{y}}_i(t)$ must be negative at t and, since $\breve{\boldsymbol{y}}(t) \geq \mathbf{0}$, such a condition is only possible if some type j has positive and decreasing surplus deviation.)*

On the other hand, if $\dot{\breve{T}}_N(t) > 0$ and $\breve{y}_N(t) > 0$, (5.8) *requires that $\dot{\breve{T}}_N(t) = 1$ until the deviation of type N is cleared, and* (5.10) *ensures that once $\breve{y}_N(t) = 0$ the solution will allocate production to clearing some other item with positive deviation.*

We see then that any fluid limit solution is work-conserving. Therefore, using the same arguments as in the proof of Appendix B.3, we find that $\sum_i d_i/(e\mu_i) < 1$ is sufficient for ensuring that the fluid limit drains in finite time, which is equivalent to $\rho < e$. □

The proof of Lemma 5.2.2 has shown that the equations governing the fluid limit model are such that any solution $\breve{\mathbb{X}}(t)$ is perfectly flexible (i.e., it does not waste time due to setups) while it is draining. Of course, in the stochastic model $\hat{\Sigma}_\pi$ that we care about, it is not possible to produce multiple products without wasting time changing setups. However, we will argue in the next section that under an appropriate limit the stochastic model in fact appears to be flexible while it is clearing a large surplus deviation.

5.2.4 Convergence of the Stochastic Model to the Fluid Limit

Consider any realization of the (HZP-controlled) stochastic model $\hat{\Sigma}_\pi$ when the initial condition $\boldsymbol{y}(0)$ is far from zero. Due to the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule, the initial runs in the system will take a long time to complete and, therefore, changeovers during these long runs will have a negligible effect on the dynamics of the system. Furthermore, when looked over a long scale, the fluctuations due to failures will become less visible except for the fact that the maximum production rate of the system will appear to be reduced by a factor of e. We see then that the trajectory starts resembling a fluid limit solution $\breve{\mathbb{X}}(t)$ as we consider larger initial conditions $\boldsymbol{y}(0)$.

To illustrate this convergence, Figure 5-4 depicts a trajectory of the stochastic system over a long scale, obtained through simulation. The plot on the left hand side shows how, initially, the accumulated setup time is almost flat at 0 and then starts increasing (almost linearly) when *all* deviations are close to zero. The plot on the right shows the work-in-process of the first $N-1$ part types (i.e., the sum of $\tau_i y_i(t)$ for $i = 1, 2, \ldots, N-1$). This quantity is an $(N-1)$-dimensional Lyapunov function, and we can see that it is decreasing during periods when item N is not produced, as required by (5.9).

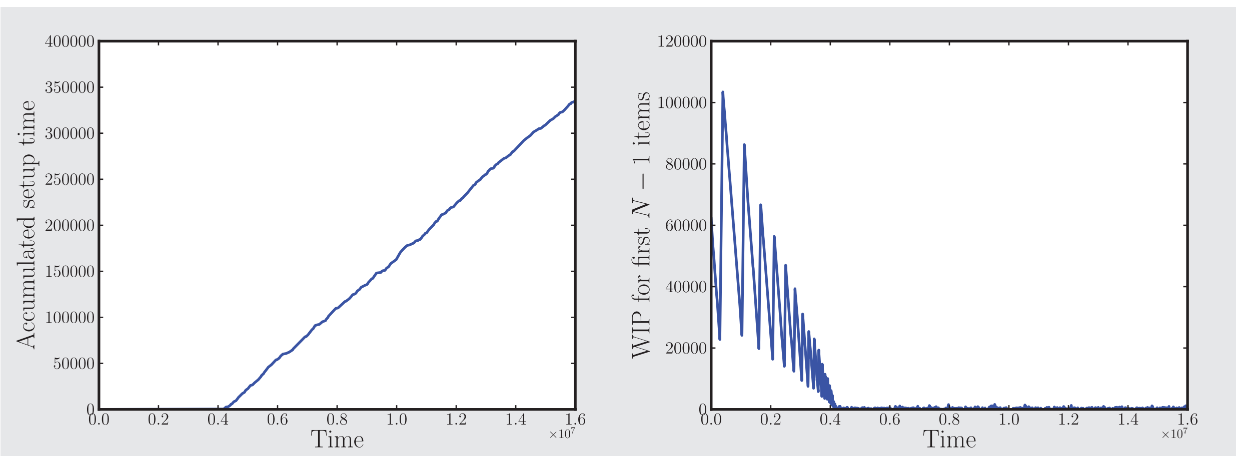


Figure 5-4: Evolution of the stochastic system $\hat{\Sigma}_\pi$ from a large initial condition under the HZP. The plot on the left shows the accumulated setup time, while the plot on the right shows the WIP for the first $N-1$ products. Notice how the plots start looking like a valid solution $\breve{\mathbb{X}}(t)$. (The parameters of the problem correspond to Instance 20 of Dataset 1, Table D.3, with $\rho/e = 0.97$, $S = 10$, MTTF $= 1000$, and MTTF $= 80$.)

We now formalize the convergence of the stochastic model to a fluid limit solution and use this convergence to establish the rate stability of $\hat{\Sigma}_\pi$. Towards this end, we will consider a *subset* of the (stochastic) dynamic equations of $\hat{\Sigma}_\pi$, and take the limit of an appropriate scaling of the solutions to these equations.

Let $\hat{\mathbb{X}}(t) = \big(\boldsymbol{y}(t), \boldsymbol{T}(t), \boldsymbol{P}(t), \boldsymbol{S}(t), \boldsymbol{R}(t)\big) \geq \mathbf{0}$ denote any solution to

$$\boldsymbol{y}(t) = \boldsymbol{y}(0) + \boldsymbol{d}t - \boldsymbol{P}(t) \tag{5.11}$$

$$\sum_{i=1}^{N} (T_i(t) + S_i(t) + R_i(t)) = t \tag{5.12}$$

$$P_i(t) = \mu_i T_i(t), \text{ for } i \in \mathcal{Q} \tag{5.13}$$

If for $t_1 \geq 0$ and *any* $t_2 > t_1$ we have $T_i(t_2) + R_i(t_2) - T_i(t_1) - R_i(t_1) > 0$, then

$$T_i(t_3) + R_i(t_3) - T_i(t_1) - R_i(t_1) = t_3 - t_1 \text{ for } t_3 > t_1, \tag{5.14}$$

where t_3 is such that $y_i(t) > 0$ in (t_1, t_3) and $y_i(t_3) = 0$, for $i \in \mathcal{Q}$.

If $y_i(t) = 0$ for some i and $\boldsymbol{y}(t) \not\leq \boldsymbol{\Delta Z}$, then $\exists\ \delta > 0$ such that

$$T_j(t + S_j + \delta) + R_j(t + S_j + \delta) - T_j(t + S_j) - R_j(t + S_j) = \delta \tag{5.15}$$

for a j with $y_j(t) > \Delta Z_j$ and $y_k(t) \leq \Delta Z_k$ for all $k < j$.

For any $t \geq 0$ and $\delta > 0$,

$$\sum_{i=1}^{N} \big(T_i(t + S_i + \delta) + R_i(t + S_i + \delta) - T_i(t) - R_i(t)\big) > 0. \tag{5.16}$$

Equation (5.11) is a material conservation statement that establishes the dynamics of the surplus deviations. In (5.12) we state that, at any instant, the stochastic system can only be either producing, changing setups, or repairing the machine, and thus the sum of the cumulative times for each activity must add up to the current time t. Equation (5.13) says that the cumulative production of item i must increase at rate μ_i during the production-allocation times (recall that we are considering the non-cruising HZP). The expression in (5.14) ensures that all runs are clearing and that the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule is satisfied. To see this, note that if at some time t_1 it is true that the sum $T_i(t) + R_i(t)$ increases over *any* interval of the form $[t_1, t_2]$, then the machine must necessarily have setup i at time t_1. Given that no changeover will take place until the deviation of type i is cleared, it then follows that

$T_i(t) + R_i(t)$ will increase at rate 1 up to t_3, which is the next time after t_1 that the deviation of i equals 0. Equation (5.15) states the HZP rule for selecting changeovers when the system is *outside of the hedging zone* $\mathcal{Z}_N$. If at time t the surplus deviation of type i is cleared, and the system is not in $\mathcal{Z}_N$ (i.e., $\boldsymbol{y}(t) \not\leq \boldsymbol{\Delta Z}$), then the system will immediately change over into the highest-priority part type j whose deviation exceeds ΔZ_j and start producing it after S_j time units. Thus, $T_j(t) + R_j(t)$ will start increasing after the changeover is complete. Finally, (5.16) avoids pathological solutions in which the sum of accumulated setup times $S_i(t)$ increases at rate 1 for all t and nothing gets produced. That is, this equation ensures that the sum of production allocation and repair times does not remain equal to 0 indefinitely.

It is important to recognize that any realization of the stochastic process $\hat{\Sigma}_\pi$ will give a solution $\hat{\mathbb{X}}(t)$. Thus, to make the distinction more evident, we will sometimes denote by $\hat{\mathbb{X}}(t, \omega)$ the realization corresponding to sample function ω, where ω is drawn from the probability space defined by our model.

We now scale $\hat{\mathbb{X}}(t, \omega)$ so that it starts looking like the plots in Fig. 5-4, where we progressively compress more and more events over a fixed period of time and ignore small fluctuations on the trajectories (i.e., we scale both time and state). For any $r > 0$, we define the scaled process $\hat{\mathbb{X}}^r(t, \omega)$ as

$$\hat{\mathbb{X}}^r(t, \omega) = \left(\frac{\boldsymbol{y}(rt)}{r}, \frac{\boldsymbol{T}(rt)}{r}, \frac{\boldsymbol{P}(rt)}{r}, \frac{\boldsymbol{S}(rt)}{r}, \frac{\boldsymbol{R}(rt)}{r} \right). \tag{5.17}$$

Now, take any sequence $\{ r_n \,;\, n \geq 0 \}$ such that $r_n \to \infty$. As proved in Section C.1 of Appendix C, for any such sequence there exists a subsequence $\left\{ r_{n_q} \,;\, q \geq 0 \right\}$ for which $\hat{\mathbb{X}}^{r_{n_q}}(t, \omega)$ converges to some continuous limit,[1] which we denote by $\overline{\boldsymbol{y}}(t)$, $\overline{\boldsymbol{T}}(t)$, $\overline{\boldsymbol{P}}(t)$, $\overline{\boldsymbol{S}}(t)$,

[1] To be precise, the convergence here is uniformly on compact sets (u. o. c). A function $f_n(t)$ is said to converge to $f(t)$ u.o.c. if, for any $s \geq 0$,

$$\lim_{n \to \infty} \sup_{t \in [0,s]} |f_n(t) - f(t)| = 0.$$

and $\overline{\boldsymbol{R}}(t)$. We will show that as long as the hedging zone is selected appropriately, a valid solution to the fluid model $\breve{\mathbb{X}}(t)$ is

$$\breve{\mathbb{X}}(t) = \left(\overline{\boldsymbol{y}}(t), \overline{\boldsymbol{T}}(t) + \overline{\boldsymbol{R}}(t), \overline{\boldsymbol{P}}(t), \overline{\boldsymbol{S}}(t)\right),$$

where we note that the sum of the limits $\overline{\boldsymbol{T}}(t)$ and $\overline{\boldsymbol{R}}(t)$ is set equal to the production allocation time $\breve{\boldsymbol{T}}(t)$ in the fluid limit solution $\breve{\mathbb{X}}(t)$. Thus, by aggregating the stochastic model's production and repair time and reducing the machine's production rate from μ_i to $e\mu_i$, we obtain a valid solution to the fluid limit model.

This result, which is stated formally in the next theorem, leads to the important conclusion that the stability of the stochastic model $\hat{\Sigma}_\pi$ is implied by the stability condition derived on Chapter 3 for the corresponding deterministic system Σ_π.

Theorem 5.2.3 (Rate Stability) *Let $\hat{\Sigma}_\pi$ denote a stochastic system and Σ_π its corresponding deterministic system. If Σ_π satisfies Relaxed Stability Condition 3.3.10, then $\hat{\Sigma}_\pi$ will be rate stable.*

Proof: *We first need to show that in fact any subsequence $\hat{\mathbb{X}}^{r_{n_q}}(t,\omega)$ converges to a valid solution $\breve{\mathbb{X}}(t)$ after combining the limiting production allocation and repair processes. Although this may be inferred from the similarity between the equations for $\breve{\mathbb{X}}(t)$ and $\hat{\mathbb{X}}(t)$ and Fig. 5-4, the rigorous proof is deferred to Appendix C.*

Once established that for almost all ω the scaled system gives us a valid fluid limit solution $\breve{\mathbb{X}}(t)$, we use Lemma 5.2.2, which guarantees that this solution will drain in finite time. This means that, since any fixed initial condition $\boldsymbol{y}(0)$ converges to $\breve{\boldsymbol{y}}(0) = \mathbf{0}$ after scaling it, the fluid limit remains drained and from (5.5) we will have that $\breve{\boldsymbol{P}}(t) = \boldsymbol{d}t$ for all $t \geq 0$. Taking

$t = 1$, this implies that for all $i \in \mathcal{Q}$

$$\breve{P}_i(1) = \lim_{r\to\infty} \frac{P_i(r)}{r} = d_i,$$

and thus $\hat{\Sigma}_\pi$ is rate stable. $\square$

5.3 Performance Analysis

In this section, we study the performance of the policies under the failure-prone model. Our experimental design is very similar to that described in Chapter 4, and we use the conclusions from that chapter to interpret the results of our stochastic experiments. The results presented here constitute an important contribution from this research, as they systematically compare the performance of different scheduling policies in failure-prone systems with setups, under both make-to-stock and make-to-order settings.

5.3.1 Experimental Setup

We will consider our two familiar costs, J and I, as measures of performance. For the stochastic model, these costs are formally defined as

$$J(\boldsymbol{y}_0, \sigma_0) = \lim_{T\to\infty} \frac{1}{T}\,\mathbf{E}\left[\int_0^T \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}(t)\,\mathrm{d}t + \sum_{i,j\in\mathcal{Q}} Q_{ij}(T)K_{ij} \;\middle|\; \boldsymbol{y}_0, \sigma_0, \alpha(0) = 1\right] \tag{5.18}$$

and

$$I(\boldsymbol{x}_0, \sigma_0) = \lim_{T\to\infty} \frac{1}{T}\,\mathbf{E}\left[\int_0^T \big(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{x}^{+}(t) + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}^{-}(t)\big)\,\mathrm{d}t + \sum_{i,j\in\mathcal{Q}} Q_{ij}(T)K_{ij} \;\middle|\; \boldsymbol{x}_0, \sigma_0, \alpha(0) = 1\right], \tag{5.19}$$

where we assume that, for the policies under consideration, the limits always exist and are independent of the initial conditions.

Our stochastic experiments focused on the same systems of Chapter 4, which had only three part types, sequence-independent setups times, a target service rate of 95% for all items, and no cruising. Motivated by our deterministic experiments, where there was no significant difference in our conclusions for the cases of $\rho = 0.9$ and $\rho = 0.99$, we decided to focus exclusively on the case $\rho/e = 0.9$. Only Dataset 1 was simulated (see Table D.3). The machine efficiency was fixed to $e = 0.91$ in all cases and we considered non-dimensional values for the mean time to fail of $\mathrm{MTTF}/S = 2, 8, 14$, and 20, where $S = S_1 + S_2 + S_3 = 10$. By holding the machine efficiency fixed and taking the limit as MTTF goes to 0, we obtain a deterministic system with maximum production rate $e\boldsymbol{\mu}$ and utilization of 0.9. This corresponds to one of the cases considered in our deterministic experiments of Chapter 4, and thus we include those results here as well for comparison (labelled as "MTTF $\rightarrow$ 0").

For each system instance and policy combination, the estimators of J and I were obtained through 30 independent replications, and the reported (absolute) 95% confidence intervals were calculated assuming a normal distribution. All systems started with the initial condition $\boldsymbol{y}_0 = (\,500, 500, 500\,)^{\mathrm{T}}$, $\sigma_0 = 1$ (as before, $c_i = h_i = 1$ and $c\mu = h\mu = 1$) and the averages were computed during the simulation time interval $[2.5 \times 10^6, 3.0 \times 10^6]$, with $S = 10$. In the worst case, this simulation interval encompassed about 10,000 and 2,000 failures during the transient and averaging periods, respectively, and allowed us to achieve adequate accuracies in the results (as estimated through the confidence intervals). Also, since our main concern is the difference in costs between policies, for each replication we used common random numbers across all policies, which helped reduce the variability of the estimators.

5.3.2 Results and Analysis

Cost J

The simulation results for cost J are presented in Fig. 5-5, where we see that our conclusions of the previous chapter regarding the relative performance of HZP versus CLB and LOP still hold (c.f. Fig. 4-14). In particular, the HZP tends to outperform the other CC policies most of the time, and has a better average performance (over all instances).

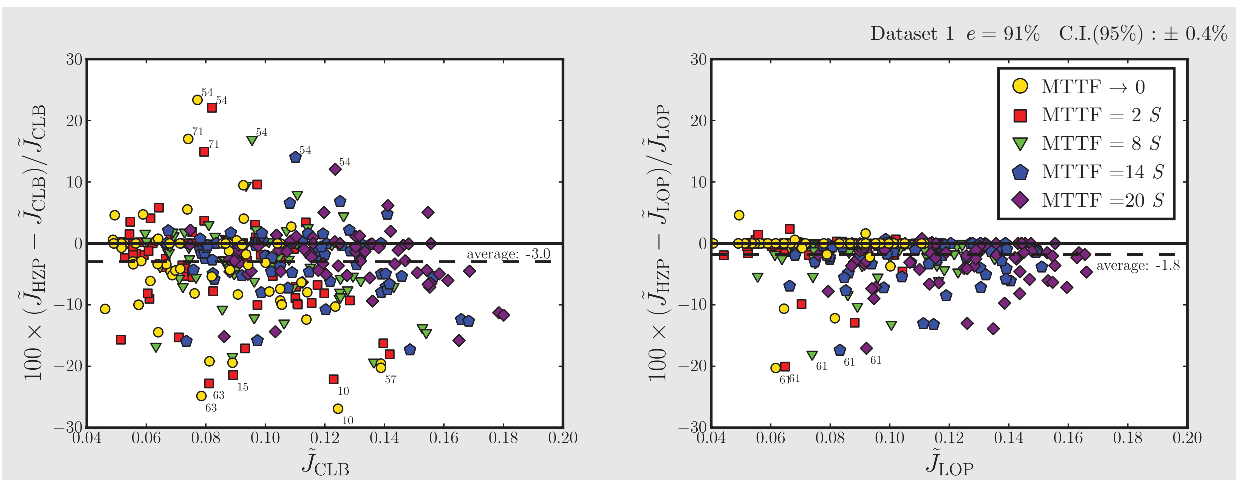


Figure 5-5: Comparisons of J costs of the HZP with respect to CLB and LOP at different MTTF values. Negative values correspond to a lower cost for the HZP. All instances were taken from Dataset 1 with $\rho/e = 0.90$ and $e = 0.91$.

It is also interesting to consider the effect that the frequency of failures has on the performance comparisons. Figure 5-6 shows the percent difference in J-costs for HZP with respect to the other policies as a function of MTTF. The plotted values correspond to the averages over all system instances, and negative values correspond to a better performance of the HZP. We can see that the average performance of the HZP and CLB turns slightly more even as failures become less frequent and repairs take longer, while the HZP gives an increasingly better performance with respect to LOP. The latter result can be explained from the recovery costs C_J comparisons of Chapter 4 (see Fig. 4-17). As breakdowns take longer to repair, the system will spend more time recovering from large surplus deviations, and we know from our deterministic experiments that the HZP performs a more efficient recovery than LOP.

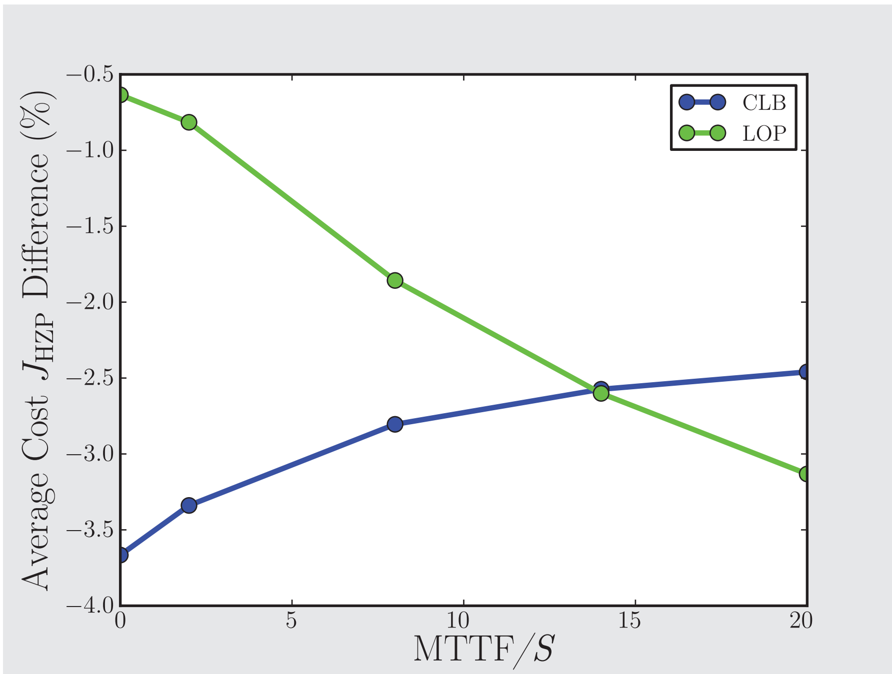


Figure 5-6: Relative J-costs difference for the HZP with respect to CLB and LOP, averaged over all system instances and as a function of MTTF, normalized by $S = S_1 + S_2 + S_3$. Negative values correspond to a lower cost for the HZP.

Cost I

Figure 5-7 shows the performance comparisons in terms of inventory and backlog costs, and Figure 5-8 plots the relative I-cost difference averaged over all system instances, for each value of MTTF. We observe the same trends when comparing the HZP with CLB and LOP. Namely, the former tends to outperform the other CC policies in most cases and, as failures become less frequent, the HZP and CLB have more even performance, while the performance of LOP worsens.

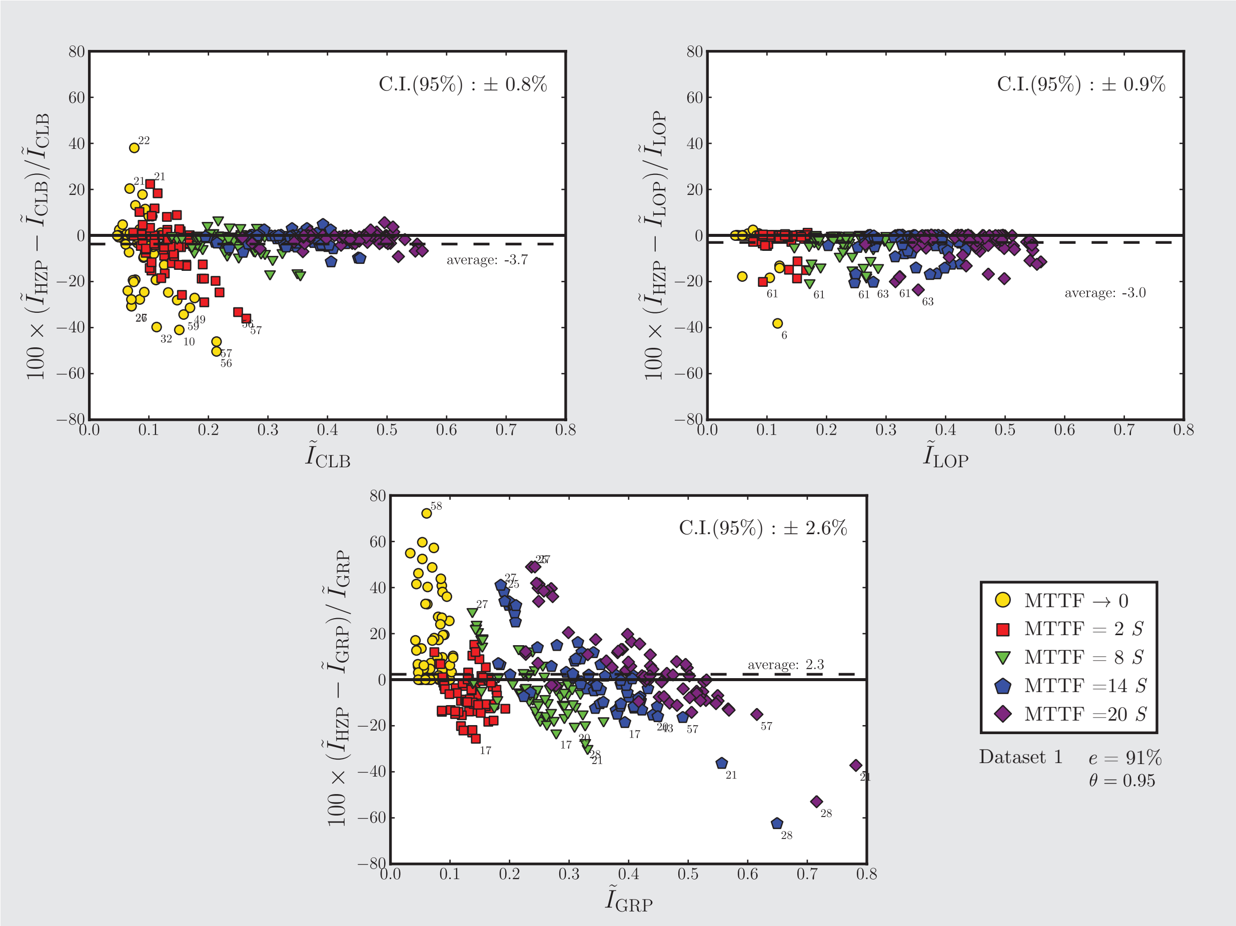


Figure 5-7: Comparisons of I costs of the HZP with respect to CLB, LOP, and GRP at different MTTF values. Negative values correspond to a lower cost for the HZP. All instances were taken from Dataset 1 with $\rho/e = 0.90$ and $e = 0.91$.

The comparisons with GRP provide some interesting new insights. Recall that, for the deterministic systems of the previous chapter, GRP is almost optimal because it always converges to the near-optimal cyclic schedule (which was found through exhaustive search). However, for a fixed machine efficiency, as we start considering mean times to fail that are not so short that random failures can be averaged out of the model, we see that the HZP tends to outperform GRP. This supports our statement of Chapter 1, where we argued that a solution that is optimal or works well in a deterministic formulation might not always be adequate for a stochastic environment. In particular, when the target deterministic sequence is long, control updates will occur less frequently (recall that in our implementation of GRP

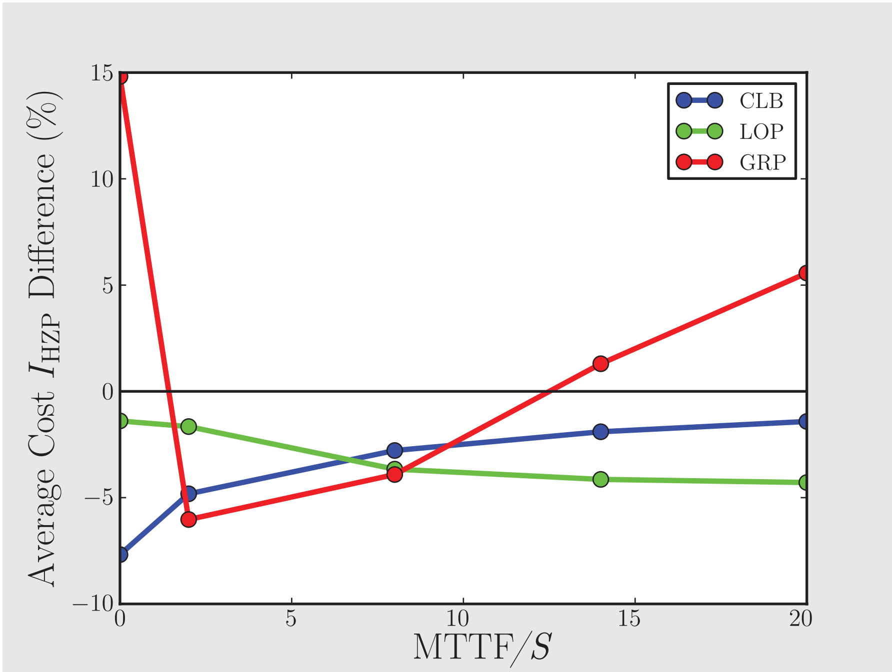


Figure 5-8: Relative I-costs difference for the HZP with respect to CLB, LOP, and GRP, averaged over all system instances and as a function of MTTF, normalized by $S = S_1 + S_2 + S_3$. Negative values correspond to a lower cost for the HZP.

the production times are only updated at the beginning of the sequence) and we therefore should expect a less responsive system with jerkier controls. As Gallego (1990) suggests, we could devise a more complex GRP implementation that updates the production times at the beginning of each position in the sequence, rather than just at the beginning of the whole sequence. However, the present implementation may be more representative of the actual practice in many shop floors, where a target schedule is "optimized" at the beginning of a fixed period (e.g., every Monday) and then the plant managers must do their best to follow this schedule under the uncertainty that surrounds the production environment. The results of this section show that it may not be worthwhile to spend significant amounts of computational resources obtaining these "optimal" schedules, since their cost benefits may become severely compromised due to the system's variability.

We also see from Figs. 5-7 and 5-8 that, as failures become more rare (large MTTF's), the

GRP starts regaining its advantages over the HZP. This is explained by the fact that the system now has more time to recover between failures. Thus, we are in a regime that is closer to the assumptions for which GRP behaves optimally (i.e., recovery from a single disruption and convergence to an optimal target schedule). While the results imply that the HZP loses its advantages over GRP in systems with infrequent failures, it is interesting to note that as we start considering MTTF's that are much greater than the setup times in the system (whose average is $S/3$), the hierarchical framework of Gershwin (1989), discussed in Section 1.2.2, could be adopted for scheduling the system. In this case, if we focus on the typical time scale at which failures take place, setups will occur so frequently that they can be taken into account in terms of their average frequency. Thus, the system could be scheduled first by solving a flow-control problem that accounts for lost capacity due to changeovers, and then the target rates resulting from the solution to this problem could be fed into an HZP controller in order to determine the exact changeover times. We expect that such a hierarchical approach will make the HZP more competitive with GRP for large values of MTTF/S, and thus should be investigated further.

Finally, we also observe that for MTTF's larger than $2S$, the distribution of points in the HZP-vs-GRP plot (Fig 5-7) seems to be quite even (i.e., with a mean close to 0) if we ignore a cluster of outliers in the upper left hand side (near the labelled instances 25 and 27). This suggests that it may be possible to select a better hedging zone (i.e., a better $\boldsymbol{\Delta Z}$) for systems that share similar characteristics as the ones in this cluster, leading to an improved performance of the HZP. This is a topic for future research that we discuss in the following chapter.

5.4 Summary

This chapter has considered the problem of scheduling under random failures by revisiting the previous chapters under this new formulation. We began by presenting our model of failures, which consisted of i.i.d. exponential machine uptimes and downtimes. We then adapted the scheduling policies for this stochastic model by enforcing the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule strictly, even after long repairs. It was argued that with this restriction the stochastic system maintains a close relationship with a deterministic system in which the maximum production rate is replaced by the effective production rate, which accounts for the machine reliability. This relationship motivated us to use our parameter selection procedures based on the deterministic system for tuning the policies in the stochastic model.

We then provided a proof of the stochastic stability of our model, supported by the tools and methodology from the queuing networks literature. We showed that under an appropriate scaling, our stochastic model converges to a fluid limit whenever the corresponding deterministic model satisfies the Relaxed Stability Condition of Chapter 3. Since by construction this fluid limit is always stable, we thus verified that stability of the deterministic system implies stability of our stochastic model.

Finally, we complemented our numerical experiments of Chapter 4 by considering machine failures. The results showed that the HZP tends to outperform the other policies within the CC Class, and we identified cases in which the HZP performs better than GRP.

We now have completed the presentation of the main contributions from this research. Clearly, there is still much room to increase the generality of our model and the applicability of our policies, and in the next chapter we discuss some interesting directions for future work along these lines.

Chapter 6

Extensions and Future Research

This chapter explores several directions of study that were beyond the scope of this research but that constitute interesting extensions for future work. In particular, we address the limitations in our experimental evaluation of the policies, discussing the cases of systems with $N > 3$, setup costs, and cruising. Through an example problem, we provide evidence that our main conclusions from the previous chapters might still apply to these more complex cases. We also discuss the case in which the items might have different service rate requirements, proposing a method for prioritizing part types in the HZP. Then, the case of sequence-dependent setups is discussed and illustrated through an example, where we use a modified version of the HZP that gives more flexibility to the policy.

Another topic we discuss in this chapter is the case in which the model's randomness originates not from breakdowns but from shortages in the supply of raw material. We state this model in some detail and discuss its connection with the problem of scheduling multiple stages in a manufacturing line. We propose a suitable modification to our policy that is easy to implement and that may share some of the benefits of the HZP that were observed for failure-prone systems. Finally, we conclude by indicating other interesting system models to consider in the future.

6.1 Sequence-Independent Setups

6.1.1 Large Systems with Setup Costs

As evidenced in Chapters 4 and 5, the performance of any scheduling heuristic tends to be heavily dependent on system data. Therefore, rather than studying a few large systems, our approach in the previous chapters consisted of exploring systematically a narrow subset of the system parameter space that consisted of cases with three-part-types and no setup costs. The results of this exploration constitute a starting point for addressing more general cases, such as machines that produce more than three items and that, in addition to setup times, are also subject to setup costs. It would also be desirable to consider the performance of the policies in situations where cruising is ideal, which occurs in systems with relatively low utilizations and/or significant setup costs.

While more research is needed to study the cases mentioned above, we expect that many of the conclusions from the previous chapters will still hold. For example, the fact that the HZP has lower recovery costs than LOP should prove advantageous under these more complex systems as well, particularly when recovering from a long repair. To illustrate this fact, we present results for a 10-part type system with setup times and costs, and for which cruising is ideal (i.e., the lower bound solution cruised). This example system, which was based on real data, was first studied by Bomberger (1966), and constitutes a benchmark problem that has been considered by many authors in the ELSP literature. The parameters for the problem are summarized in Table D.2 of Section D.2, and the performance comparisons for the HZP and LOP are shown in Table 6.1 (negative values indicate that the HZP had a *lower* cost than LOP).

Table 6.1: Relative difference in cost I for HZP w.r.t. LOP for Bomberger's 10 part type problem ($e = 0.91$, $\theta = 0.99$).

Det.	**MTTF 5S**	**MTTF 21S**	**MTTF 37S**	**MTTF 53S**
0.68%	-6.5%	-14.1%	-15.3%	-15.7%

For the stochastic simulations, 95% C.I. was within $\pm 0.5\%$.

We can see from the simulation comparisons that, although LOP outperforms HZP in the deterministic case by a small percentage, the performance benefits of the HZP over LOP increase as failures become less frequent and the machine takes on average longer to repair.

It is also interesting to compare the costs attained by the policies in the deterministic case with ELSP solutions to Bomberger's problem. The results are summarized in Table 6.2, where the last row corresponds to, as far as we know, the best cost for this problem found in the literature, as reported in Table 2 by Davis and Davis 1990. (Regarding the numbers in the last row of Table 6.2 we recall that, as discussed in Chapter 4, any ELSP solution works equally well for both J and I. However, the published result for the ELSP heuristic was obtained for a service level of 1.0, while the other I-costs in the table assume a service level of 0.99. Thus, for $\theta = 0.99$, the ELSP heuristic will have a slightly lower I-cost than the one indicated in the table.)

Table 6.2: Deterministic System Costs in \$/year for Bomberger's 10 part type problem ($\theta = 0.99$).

Policy	J	I
HZP	7,888	9,658
LOP	7,862	9,592
ELSP heur.	7,697	7,697[a]

[a]This cost was obtained for $\theta = 1.0$.

We can see that, in terms of cost J, both the HZP and LOP are very competitive with respect to the ELSP heuristics in this problem, while in terms of inventory and backlog costs the trajectories have costs that are at least 25% higher than the best published solutions. This is consistent with our results and analyses of Chapter 4, and is attributed to the different structure of the trajectories (i.e., the scheduling policies follow the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule and cruise only when $x_i = Z_i^{\mathrm{U}}$ for the current setup i, while the ELSP solutions follow the Zero-Switch Rule and, although they do not cruise at 0 as in the ideal case, they idle when necessary). On the other hand, the ELSP solution consisted of an $\boldsymbol{f}$-cycle of size $M = 40$. In the stochastic system, such a long sequence will be troublesome to follow using GRP since the control updates will occur very infrequently (also, the computations for GRP needed to obtain the optimal $\boldsymbol{f}$-cyclic schedule and matrix $\boldsymbol{G}$ become harder). Thus, while the HZP and the other CC Class policies readily handle arbitrarily-large systems, fixed-sequence policies become more complex and difficult to implement.

6.1.2 Learning from the Results for $N = 3$ Systems

In our simulation results of the previous chapters, we did not examine any possible relationships between each instance's parameters and its performance. Further research is needed in order to identify if such relationships exist, as well as to determine their generalization or prediction power when considering new instances that were not in the original simulation experiments.

As an example, Figs. 6-1 and 6-2 show the I-costs for MTTF $= 8S$ that were presented in Chapter 5, where we have marked each simulation point in a way that makes it possible to identify each system instance. The circular markers are split into three sections, corresponding to the three triads of parameters related to the utilizations, the $h\mu$ coefficients, and the setup times (see Fig. 4-11 and explanations therein). The markers' sections are colored according to the location of the triad in its triangular parameter space. Thus, for example, a marker with the colors red, green, and blue indicates that each of the triads was located close to a different corner in its corresponding triangle. This means that, for that system instance, one part type had the largest utilization, another part type had the largest $h\mu$ coefficient, and the remaining part type had the largest setup time.

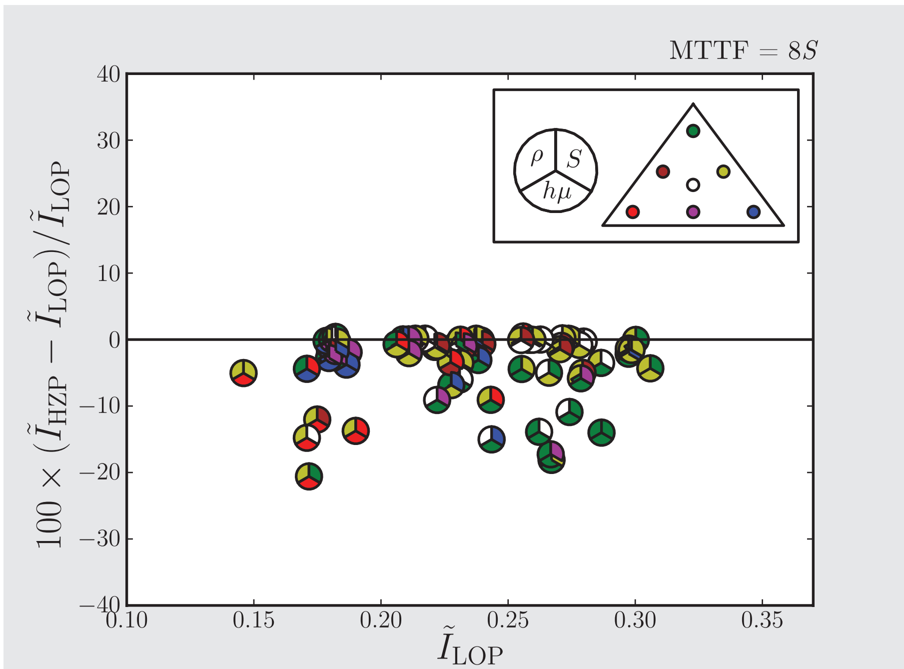


Figure 6-1: I-costs comparison for the HZP vs LOP. Each system instance is identified through the coloring scheme explained in the plot. The results shown are for MTTF $= 8S$.

One pattern we can readily identify from Fig. 6-1 is that, in most of the systems in which one of the $h\mu$ coefficients was much larger than the other two (which corresponds to the $h\mu$ section in the marker being colored either red, green, or blue), HZP outperformed LOP. This is not surprising, given that the HZP gives more attention to the items with large $h\mu$ (or, since b_i/h_i was assumed constant for all i, to $b\mu$) coefficients. We can also see from Fig. 6-2 that the upper left hand side cluster of systems in which GRP outperforms the HZP (which we discussed in Section 5.3.2) seems to consist of systems in which the part type with the largest utilization also had a small $h\mu$ coefficient compared to the other two items.

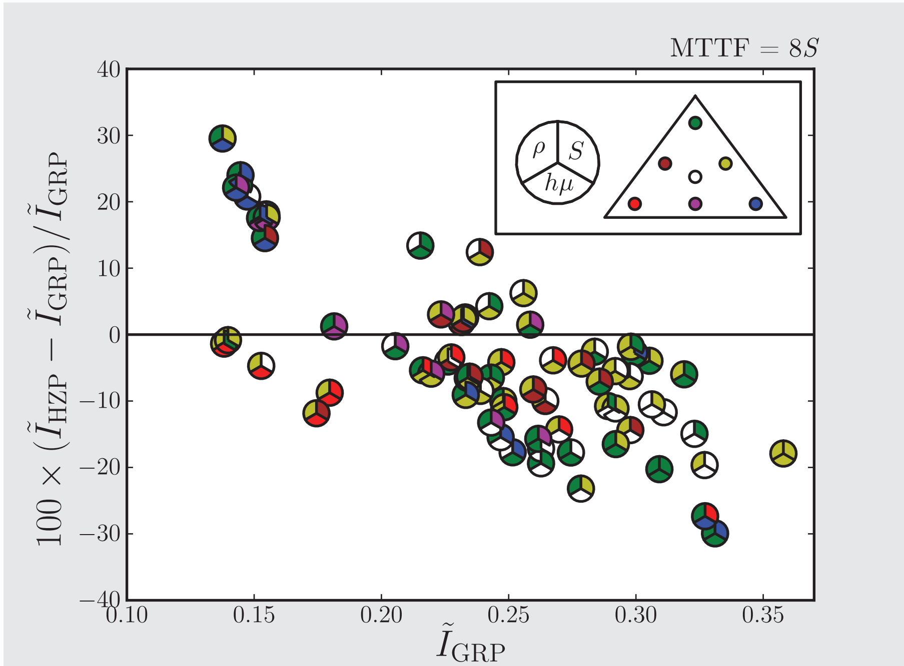


Figure 6-2: I-costs comparison for the HZP vs GRP. Each system instance is identified through the coloring scheme explained in Fig. 6-2. The results shown are for MTTF $= 8S$.

This suggests that the HZP is not devoting enough attention to that high-utilization item because of its low $h\mu$ coefficient.

More work is needed to identify other relationships of this type and to study their robustness. In particular, it would be interesting to apply machine learning algorithms (e.g. support-vector machine classifiers) to the data and study their generalization properties with new system instances (i.e., instances that are not in the learning set of simulations). If the idea proves successful for three-part-type systems, it would also be desirable to consider ways to extend it to systems with more part types. For example, it may be that only a few summarizing values of the system's parameters (e.g. the mean, range, and standard deviation of each set of related parameters ρ, $h\mu$, and S) are necessary to classify instances. Such a classifier would then allow us to select *a priori* (i.e., without having to simulate the system) the best policy to use or the best choice of policy parameters, based only on the demand, production rates, setup times, etc. The classifier could also be helpful for developing better heuristics and for designing simulation experiments over larger system parameter spaces.

6.1.3 Systems with Item-Dependent Target Service Levels

In our simulation experiments, we assumed that the ratio b_i/h_i was the same for all i (or, equivalently, that all part types had the same target service level). As discussed in Section 4.4.2, this simplification implies that the $h\mu$ and $b\mu$ rules give the same rank ordering of part types. On the other hand, when the desired service levels of the items are not all equal, the most adequate rank ordering should be determined.

One approach in this case is motivated from the lower bounds J_{LB} and I_{LB} derived in Chapter 4. A comparison of the two bounds suggests that the prioritization rule

$$\mathsf{P}(i) = \frac{h_i b_i}{h_i + b_i}\mu_i = h_i \theta_i^* \mu_i \tag{6.1}$$

could provide an adequate rank ordering, especially if there are two items that have similar $h_i\mu_i$ coefficients but one has a larger target service rate θ_i^*. Furthermore, this prioritization reduces to the $h\mu$-Rule (or, equivalently, the $b\mu$-Rule) when b_i/h_i is the same for all i.

Another reasonable heuristic for prioritizing part types is derived as follows. Let X_i be a random variable corresponding to the system's surplus in steady state. That is, X_i is distributed according to the steady-state probability of the system, so that

$$\mathbf{P}\left\{X_i = \tilde{x}_i\right\} = \lim_{t\to\infty} \mathbf{P}\left\{x_i(t) = \tilde{x}_i\right\}.$$

Under the necessary ergodicity assumptions, and assuming no setup costs, it follows that I

can be written as

$$I = \sum_{i=1}^{N} \left(h_i \mathbf{E}\left[\, X_i^+ \,\right] + b_i \mathbf{E}\left[\, X_i^- \,\right] \right),$$

where $X_i^+ = \max(0, X_i)$ and $X_i^- = \max(0, -X_i)$.

Now, let $Y_i = Z_i^{\mathrm{U}} - X_i$ represent the steady-state surplus deviation and assume, for the sake of our derivation, *that this variable is exponentially-distributed*. Note then that $\mathbf{P}\left\{Y_i = Z_i^{\mathrm{U}}\right\} = 0$ and thus

$$\mathbf{E}\left[\, Y_i \,\right] = \mathbf{E}\left[\, Y_i \;\middle|\; Y_i < Z_i^{\mathrm{U}} \,\right] \mathbf{P}\left\{Y_i < Z_i^{\mathrm{U}}\right\} + \mathbf{E}\left[\, Y_i \;\middle|\; Y_i > Z_i^{\mathrm{U}} \,\right] \mathbf{P}\left\{Y_i > Z_i^{\mathrm{U}}\right\}.$$

Since $\mathbf{E}\left[\, Y_i \;\middle|\; Y_i > Z_i^{\mathrm{U}} \,\right] = \mathbf{E}\left[\, Y_i \,\right] + Z_i^{\mathrm{U}}$ (this follows from the assumed memoryless property of Y_i), the above equation implies that

$$\begin{aligned}
\mathbf{E}\left[\, Y_i \;\middle|\; Y_i < Z_i^{\mathrm{U}} \,\right] &= \frac{\mathbf{E}\left[\, Y_i \,\right] - \mathbf{E}\left[\, Y_i \;\middle|\; Y_i > Z_i^{\mathrm{U}} \,\right] \mathbf{P}\left\{Y_i > Z_i^{\mathrm{U}}\right\}}{\mathbf{P}\left\{Y_i < Z_i^{\mathrm{U}}\right\}} \\
&= \frac{\mathbf{E}\left[\, Y_i \,\right] - (\mathbf{E}\left[\, Y_i \,\right] + Z_i^{\mathrm{U}}) \exp(-Z_i^{\mathrm{U}} / \mathbf{E}\left[\, Y_i \,\right])}{1 - \exp(-Z_i^{\mathrm{U}} / \mathbf{E}\left[\, Y_i \,\right])} \\
&= \mathbf{E}\left[\, Y_i \,\right] - \frac{Z_i^{\mathrm{U}} \exp(-Z_i^{\mathrm{U}} / \mathbf{E}\left[\, Y_i \,\right]}{1 - \exp(-Z_i^{\mathrm{U}} / \mathbf{E}\left[\, Y_i \,\right])}.
\end{aligned}$$

Using the above expression, we find that $\mathbf{E}\left[\, X_i^+ \,\right]$ is given by

$$\begin{aligned}
\mathbf{E}\left[\, X_i^+ \,\right] &= \mathbf{E}\left[\, X_i^+ \;\middle|\; X_i > 0 \,\right] \mathbf{P}\left\{X_i > 0\right\} + \mathbf{E}\left[\, X_i^+ \;\middle|\; X_i < 0 \,\right] \mathbf{P}\left\{X_i < 0\right\} \\
&= \mathbf{E}\left[\, X_i \;\middle|\; X_i > 0 \,\right] \mathbf{P}\left\{X_i > 0\right\} \\
&= \mathbf{E}\left[\, Z_i^{\mathrm{U}} - Y_i \;\middle|\; Y_i < Z_i^{\mathrm{U}} \,\right] \mathbf{P}\left\{Y_i < Z_i^{\mathrm{U}}\right\},
\end{aligned}$$

and, using our previous result, we get

$$\begin{aligned}\mathbf{E}\left[X_i^+\right] &= \left(Z_i^{\mathrm{U}} - \mathbf{E}[Y_i] + \frac{Z_i^{\mathrm{U}}\exp(-Z_i^{\mathrm{U}}/\mathbf{E}[Y_i])}{1-\exp(-Z_i^{\mathrm{U}}/\mathbf{E}[Y_i])}\right)(1-\exp(-Z_i^{\mathrm{U}}/\mathbf{E}[Y_i])) \\ &= Z_i^{\mathrm{U}} - \mathbf{E}[Y_i](1-\exp(-Z_i^{\mathrm{U}}/\mathbf{E}[Y_i])).\end{aligned}$$

Similarly, we obtain an expression for $\mathbf{E}\left[X_i^-\right]$ given by

$$\begin{aligned}\mathbf{E}\left[X_i^-\right] &= \mathbf{E}\left[X_i^- \mid X_i > 0\right]\mathbf{P}\{X_i > 0\} + \mathbf{E}\left[X_i^- \mid X_i < 0\right]\mathbf{P}\{X_i < 0\} \\ &= \mathbf{E}\left[-X_i \mid X_i < 0\right]\mathbf{P}\{X_i < 0\} \\ &= \mathbf{E}\left[Y_i - Z_i^{\mathrm{U}} \mid Y_i > Z_i^{\mathrm{U}}\right]\mathbf{P}\left\{Y_i > Z_i^{\mathrm{U}}\right\} \\ &= \mathbf{E}[Y_i]\exp(-Z_i^{\mathrm{U}}/\mathbf{E}[Y_i]).\end{aligned}$$

Therefore, the long-term average cost I is given by

$$I = \sum_{i=1}^{N}\left[h_i\left(Z_i^{\mathrm{U}} - \mathbf{E}[Y_i](1-\exp(-Z_i^{\mathrm{U}}/\mathbf{E}[Y_i]))\right) + b_i\mathbf{E}[Y_i]\exp(-Z_i^{\mathrm{U}}/\mathbf{E}[Y_i])\right].$$

We now use our expression for the optimal service level, (4.3), which states that

$$\mathbf{P}\{X_i > 0\} = \frac{b_i}{h_i + b_i}.$$

This allows us to relate the base stock level Z_i^{U} to $\mathbf{E}[Y_i]$ through the expression

$$\exp(-Z_i^{\mathrm{U}}/\mathbf{E}[Y_i]) = \frac{h_i}{h_i + b_i}.$$

Finally, substituting this result into the expression for I, we get that

$$I = \sum_{i=1}^{N}\left[h_i\left(\mathbf{E}[Y_i]\log\left(\frac{h_i + b_i}{h_i}\right) - \mathbf{E}[Y_i]\frac{b_i}{h_i + b_i}\right) + b_i\mathbf{E}[Y_i]\frac{h_i}{h_i + b_i}\right],$$

which simplifies to

$$I = \sum_{i=1}^{N} h_i \log\left(\frac{h_i + b_i}{h_i}\right) \mathbf{E}\left[\, Y_i \,\right]. \tag{6.2}$$

Notice that the above result solely depends on the long-term expected value of the surplus deviation, and thus it has the same form as the expression for cost J. Thus, this shows that *if the long-term surplus deviations are exponentially distributed*, then optimizing cost I is equivalent to optimizing cost J with

$$c_i = h_i \log\left(\frac{h_i + b_i}{h_i}\right). \tag{6.3}$$

Of course, in the actual system, the surplus deviations will generally not be exponentially distributed. Thus, in order to get an idea of the validity of this approximation, we determined the empirical distribution of Y_i for an example system, shown in Fig. 6-3.

To explain the shape of Fig. 6-3, it is important to realize that the distribution was taken over continuous time. Also, note that if the system had no failures and converged to a limit cycle in which each item gets produced exactly once per cycle, the long-term distribution of $y_i(t)$ for any i would be uniform with an edge at 0. The tail of the distribution in Fig. 6-3 is then showing a mixed behavior between the deterministic segments of the trajectory and the recoveries from random failures.

Interestingly enough, when we compare the empirical distribution with an exponential distribution that has the same sample mean, we find that the approximation is not terribly far off and may in fact be adequate for coming up with a good prioritization scheme. Therefore, while the quality of this approximation may depend on the system's parameters and the machine's failure model, we believe that the rule

$$\mathsf{P}(i) = h_i \log\left(\frac{h_i + b_i}{h_i}\right) \mu_i = h_i \log\left(\frac{1}{1 - \theta_i^*}\right) \mu_i \tag{6.4}$$

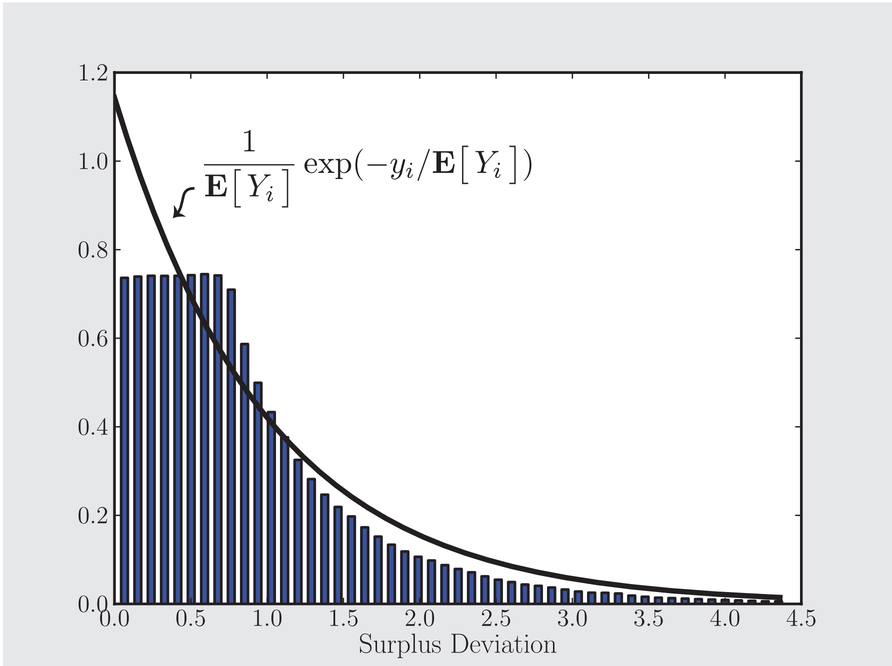


Figure 6-3: Empirical distribution of the surplus deviation for a system operated under the HZP. An exponential distribution with the sample mean is overlaid for comparison. (The parameters of the problem correspond to Instance 19 of Dataset 1 (see Table D.3), with $\rho/e = 0.9$, $S = 10$, MTTF = 80, and MTTR = 8.)

could also work well for rank ordering items. As with the previous scheme, this prioritization function reduces to the $h\mu$-Rule when θ_i^* is the same for all items and, otherwise, it weights very heavily on items whose target service rate is very close to 1. Furthermore, our derivations suggest that, for I-OP, rather than using the ideal deviations y_i^* computed from I_{LB} for selecting $\boldsymbol{\Delta Z}$ (see Section 4.4.2), we could use the ideal deviations resulting from the solution to J_{LB}, with cost coefficients given by (6.3).

6.2 Sequence-Dependent Setups

6.2.1 Generalization of the HZP

As we increase the level of complexity of the systems under study, the next step consists of considering sequence-dependent setups. Most of our analytical results from the previous chapters are directly applicable or easily extended to this case. However, it is likely that we will need a more general policy to deal with these systems. In particular, it seems natural to allow the hedging points to be sequence dependent so that, for each $i, j \in \mathcal{Q}$, we define the thresholds differences as $\Delta Z_{ij} = Z_i^{\mathrm{U}} - Z_{i,j}^{\mathrm{L}}$. (Note that this means that the hedging zone now depends on the current setup.) This extra generality allows us to encourage certain changeovers over others with more precision, but it comes at the price of having more parameters that need to be selected though a (hopefully) robust procedure.

6.2.2 Hierarchical Setups Example

To illustrate some of the difficulties that arise with sequence-dependent setups, we consider an example that was inspired by the work of Burman (1993). Burman studied systems that can produce multiple products and in which setup times are determined by different configuration characteristics (e.g., a raw material configuration, a punch die configuration, depth and length settings, etc.). The changeover times for the configurations are such that they can be arranged into a tree structure, in which changing a configuration that is high up in the tree takes much longer than changes of configurations that are further down in the tree. That is, suppose that a machine has K configuration characteristics $A_1, A_2, \ldots, A_K$ ordered in terms of the length of their changeover times. Let $p_{a_1 a_2 \ldots a_K}$ denote a part type with settings $a_1, a_2, \ldots, a_K$ for each of the characteristic and suppose that a changeover into some other part type $p_{a'_1 a'_2 \ldots a'_K}$ is required. In Burman's model, the changeover time depends only on the highest-level characteristic that is common to both parts. Thus, the time to change parts is given by some time s_n, where n is the smallest index for which a_n and a'_n differ. That is,

$$a_1 = a'_1, a_2 = a'_2, \ldots, a_{n-1} = a'_{n-1} \text{ and } a_n \neq a'_n.$$

Consider an example consisting of two characteristics, with $s_1 = 100$ and $s_2 = 10$. There are two possible settings for characteristic 1 and three for characteristic 2, giving a total of 6 different part types. The changeover times between part types are given by the matrix

$$\boldsymbol{S} = \begin{bmatrix} 0 & 10 & 10 & 100 & 100 & 100 \\ 10 & 0 & 10 & 100 & 100 & 100 \\ 10 & 10 & 0 & 100 & 100 & 100 \\ 100 & 100 & 100 & 0 & 10 & 10 \\ 100 & 100 & 100 & 10 & 0 & 10 \\ 100 & 100 & 100 & 10 & 10 & 0 \end{bmatrix}$$

Notice that we can group the part types into two clusters, one for parts 1,2, and 3 and another for parts 4,5, and 6. Setup changes within each cluster are short, while changes between different clusters take much longer. We assume that the demand rate for each product is equal to $d_i = 0.15$ and that the maximum production rate for each item is $\mu_i = 1$. There are no setup costs and the holding costs per item are $h_1 = h_2 = h_3 = 1$ and $h_4 = h_5 = h_6 = 4$. A tour that minimizes total setup time is $(1, 2, 3, 4, 5, 6, 1)$, and so we will use this as the target sequence $\boldsymbol{f}$ in GRP.

An attempt to follow the HZP Parameter Selection Procedure of Chapter 4 quickly shows the problems that may occur with sequence-dependent systems. Solving for J_{LB}, we get a solution in which $n^*_{ij} = 0$ for any i and j that belong to different clusters. That is, the lower bound's solution will likely be far from optimal because the frequencies are not feasible.

(Further research is needed to develop a lower bound that avoids this problem and that is still relatively easy to solve.) We can, however, force the sequence $\boldsymbol{f} = [1, 2, 3, 4, 5, 6]$ through the sequence-dependent thresholds differences ΔZ_{ij}. For $n = 1, 2, \ldots, 6$, we set $\Delta Z_{f^n j} \gg \Delta Z_{f^n f^{n+1}}$, for $j \neq f^{n+1}$. This means that, upon completing part type f^n, the system is very likely to switch to part type f^{n+1}, which has a small thresholds difference, unless there is some higher-priority item with an unusually large surplus deviation.

After a few trials, we find that $\Delta Z_{ij} = 500$ for the inter-cluster thresholds and $\Delta Z_{ij} = 100$ for the intra-cluster thresholds makes the system converge to $\boldsymbol{f} = [1, 2, 3, 4, 5, 6]$ in the deterministic case. The results comparing the HZP and GRP (for the same target $\boldsymbol{f}$) are summarized in Table 6.3, where we see that the HZP has a slight advantage over GRP in the failure-prone system.

Table 6.3: Relative difference in cost I for HZP w.r.t. GRP, for the hierarchical setups example ($e = 0.91$, $\theta = 0.95$).

Det.	**MTTF 20**	**MTTF 80**	**MTTF 140**	**MTTF 200**
0.0	-1.3%	-3.4%	-4.8%	-6.0%

For the stochastic simulations, 95% C.I. was within ±0.5%.

6.3 Raw Material Shortage Model

Other lines of research should consider different sources of randomness. One important extension consists of the case in which raw material upstream of the machine is not always available, and thus changeover decisions need to take this into account. To address this problem, we propose a model in this section, as well as an intuitive adaptation of the HZP for dealing with it. We then suggest how this model fits into the general problem of scheduling multiple-stage manufacturing systems.

6.3.1 Model Description

Consider a perfectly-reliable machine in which there is a raw-material buffer with level $w_i(t)$, for $i \in \mathcal{Q}$. We can produce product i as long as there is raw material available and the machine has the correct setup. Therefore, we have the constraint on the production rate

$$0 \le u_i(t) \le \mu_i [\![\sigma(t) = i]\!] [\![w_i(t) > 0]\!]. \tag{6.5}$$

The raw-material buffer level $w_i(t)$ is given by the difference between the cumulative raw-material arrival $W_i(t)$ and the cumulative production $P_i(t)$ at time t. That is,

$$w_i(t) = W_i(t) - P_i(t),$$

and, in order for the system to be stable we require that

$$\lim_{t \to \infty} \frac{W_i(t)}{t} = d_i,$$

with probability one.

Notice that the cumulative production is constrained by the raw material process, so that at all times $P_i(t) \le W_i(t)$. Also, since $x_i(t) = P_i(t) - d_i t$, we have that $x_i(t) \le W_i(t) - d_i t$. If a policy with base stock Z_i^{U} is used, then for all t it must then be true that

$$x_i(t) \le \min(W_i(t) - d_i t, Z_i^{\mathrm{U}}). \tag{6.6}$$

This inequality shows that the model for $W_i(t)$ needs to be defined carefully in order for the long-term expected value of $x_i(t)$ to remain bounded. For example, if $W_i(t) - d_i t$ is modeled as a Brownian motion with zero drift (or if the model is such that it approaches this process in the limit), then its standard deviation will grow as $\sqrt{t}$. This growing variability means that the long-term average surplus cost will be infinite. (A suitable model that avoid this issue is as follows: Let orders arrive with i.i.d. inter-arrival times, and at each arrival time make the size of the order equal to the current difference $W_i(t) - d_i t$.)

6.3.2 Proposed Policy and Connection with Multi-Stage Systems

The cumulative raw-material arrival process $W_i(t)$ can be considered to be coming from some upstream machine, thus corresponding to that machine's cumulative production process. Let $x_i^{\mathrm{U}}(t)$ be defined by

$$x_i^{\mathrm{U}}(t) = W_i(t) - d_i t,$$

which we interpret as the surplus of the upstream machine that feeds material to the stage we are scheduling. From (6.6), we see then that

$$x_i(t) \leq \min(x_i^{\mathrm{U}}(t), Z_i^{\mathrm{U}}).$$

Now, defining a *time-varying* base stock level or upper hedging point $Z_i^{\mathrm{U}}(t)$ as

$$Z_i^{\mathrm{U}}(t) = \min(x_i^{\mathrm{U}}(t), Z_i^{\mathrm{U}}), \tag{6.7}$$

we see that at all times $x_i(t) \leq Z_i^{\mathrm{U}}(t)$.

The previous derivations suggest a way to apply the HZP directly to the problem of raw material shortages: simply follow the steps in Policy 2.4 with Z_i^{U} replaced by $Z_i^{\mathrm{U}}(t)$.[1] Given that the HZP has very good performance in terms of J-OP, the proposed policy should be

[1]The only part that we would need to modify relates to cruising. Since now the base stock level is not a fixed quantity, we cannot hold the surplus at that value by simply producing at the demand rate. Instead, we might need to "chase" $Z_i^{\mathrm{U}}(t)$ by adjusting $u_i(t)$ appropriately.

adequate for keeping the current stage's surplus for the costly items as close as possible to their base stock values. In a multi-stage system, this behavior could translate into faster flow of the expensive items through the system and thus lower WIP costs.

Future research is needed to test the performance of this modified version of the HZP and to compare it with other possible policies. A longer term goal would then consist of applying the policy to multi-stage manufacturing systems, where we would test the behavior that the local policy has on the global performance of the whole facility.

6.4 Other Models

As discussed in Chapter 1, a commonly studied problem in the real-time scheduling literature consists of the server model with discrete material, random arrival/service times, and random changeover times, either in a make-to-order setting (i.e., a polling model) or make-to-stock (i.e., the SELSP). We expect that our conclusions from the previous chapters will still be valid for this formulation, making the HZP a suitable policy for this model as well. However, simulation experiments are needed to confirm this conjecture. In particular, these experiments should replicate some of the test cases published in the literature for the different heuristics that have been proposed, such as those of Duenyas and van Oyen (1996), Federgruen and Katalan (1996), and Lan and Olsen (2006).

Another model of interest consists of the case of stages with parallel machines. In many systems, when a particular stage in the production process involves significant setups, it may be economical to purchase several identical machines and divide up the work among them. (Such was the case in the two factories that the author visited during the course of this research: a textile factory, where there were several looms that could produce the same variety of items and all incurred similar changeover times, and an auto parts manufacturer, which possessed several hot presses with similar characteristics and with very time-consuming setups due to the changing of dies.) An approach for dealing with this problem could consist of splitting the scheduling decisions into two parts: first, route work to the most appropriate machine (based perhaps on the current setup of the machine and the amount of upstream material on its buffers) and, second, use some local policy such as the HZP to schedule changeovers within each machine.

6.5 Summary

This chapter has discussed several interesting directions for further research, some of which we have already begun exploring. We discussed the need for extending the simulation results of Chapters 4 and 5 to the cases of systems with more than three part types, and also with setup costs and cruising. However, using a famous 10-part type example, we showed that many of our conclusions from the previous chapters may extend to these more complex systems. The idea of applying machine learning techniques to our experimental results was then discussed as a way to find possible relationships between the system parameters and the policies' performance. We also proposed two heuristics for selecting the parameters of the HZP when the ratio of inventory and backlog unit costs is not constant across all items.

The case of sequence-dependent setups was then addressed and we suggested an extension to the HZP that may give better results in these systems. We illustrated through an example the implementation of the extended policy and discussed some of the difficulties with selecting its parameters. Along another line of future research projects, we considered the case in which the machine is perfectly-reliable but raw material arrives randomly. A model for studying this problem and a suitable modification to the HZP were proposed. We then discussed how this model may serve as a building block for dealing with the complex problem of scheduling setups in multi-stage manufacturing systems. Finally, we concluded the chapter by mentioning other models that should be considered in the future, namely, polling systems, the Stochastic ELSP, and stages with parallel machines.

Chapter 7

Summary of Main Results and Contributions

We conclude by summarizing the main results and contributions derived from this research.

Policy Statement

- We proposed a new policy for scheduling systems with setups, the *Hedging Zone Policy* (HZP). This policy belongs to what we called the *Clearing Cruising (CC) Class*, which includes all produce-up-to or base stock policies.
- Following the results of previous researchers, we developed two versions of our policy: a non-cruising version, which always changes setups immediately after reaching the base stock level of the current item, and a cruising version, which may produce at the demand rate the current item for some time before changing setups.
- The structure of the HZP was motivated and compared with that of other policies (both in the CC Class and in more general classes) proposed in the literature for our problem.

Deterministic Stability

- We addressed the stability of the HZP in a deterministic setting. It was shown that the prioritized-structure of the changeover decisions can lead to a situation in which one or more lower-priority parts are ignored indefinitely, making the system unstable.

- We obtained sharp stability conditions for a three-part-type system. However, we illustrated through a numerical study that, for larger systems, obtaining both necessary and sufficient stability conditions is very complex.

- Using Lyapunov's direct method and a linear class of Lyapunov functions, we obtained a sufficient condition that guarantees stable production of all items. We showed that this condition can be reduced into a simple relation of the parameters of the policy, and we demonstrated its equivalence to a recently-derived condition by Dai and Jennings (2004) for stochastic queueing networks.

- We illustrated that the previous stability condition tends to be too conservative for our system. Through a careful analysis of the dynamics of the HZP, we then showed that a system that produces all part types infinitely often will always be stable. This allowed us to obtain a looser stability condition, and we illustrated through numerical experiments that the amount of relaxation can be significant.

Deterministic Performance

- We justified some of the properties of the HZP through the derivation of lower bounds on the costs of our problem. (These bounds have been well-known in the literature.) The results were also used to state a procedure for selecting the parameters of the HZP.

- We designed a set of experiments on three-part-type systems that allowed us to explore systematically different parameter combinations. Moreover, thanks to the low dimensionality of the systems considered, we were able to obtain an approximate optimal solution to the problem through an exhaustive search of production sequences.

- We presented results from our numerical simulations, which, as far as we know, constitute the first systematic study of the performance of different closed-loop policies in terms of make-to-order and make-to-stock costs, as well as in terms of the cost of recovering from a single, initial disruption.

- The experimental results showed that the HZP outperforms other policies within its class. They also confirmed that the policy performs very well with respect to the optimal cost in make-to-order problems. For (deterministic) make-to-stock problems, it was shown that the performance of the HZP (or any other policy in the CC Class) can be poor due to its base stock structure.

Failure-Prone Model

- We adapted the policies to a model where the machine can fail while producing. Relying on the concept of the *corresponding deterministic system*, we argued that this system can be used for selecting good parameters for the policies in the stochastic setting.

- We provided a rigorous proof that our sufficient stability conditions (applied to the corresponding deterministic model) imply the stochastic stability of the failure-prone machine. This proof was based on the convergence of the stochastic model to a fluid limit.

- We complemented our deterministic numerical experiments by considering different machine failure rates, for a fixed machine efficiency. This allowed us to gauge the overall performance of the heuristics as a function of the frequency of failures and setups in the system.

- The HZP was shown to outperform the other CC Class policies in the stochastic, in both make-to-order and make-to-stock formulations. Furthermore, we showed that when the frequency of setups and failures are comparable, the HZP outperforms a fixed-sequence policy (GRP) that tracks a near-optimal schedule for the corresponding deterministic system.

Extensions

- We discussed some of the limitations of our experiments that should be addressed in the future. Despite these limitations, we conjectured that our conclusions about the performance merits of the HZP will still be applicable for more complex systems. This conjecture was motivated by a performance comparison based on a widely-studied, 10-part-type example problem.

- We proposed the exploration of relationships between the parameters of the system and the performance of the policies. This idea was illustrated through two plots of our experimental results, where it was possible to observe some trends in the data (e.g., we identified that instances where the HZP outperformed one of the other CC Class policies corresponded to systems where one item had a very large $h\mu$ coefficient).

- The case of systems with different target service levels for each item was discussed. We proposed two heuristics for prioritizing parts in this case. Furthermore, we showed that when we assume that the stationary distribution of the surplus deviations is exponential, the make-to-stock and make-to-order problems are tightly related. A numerical experiment suggested that approximations based on this assumption may be adequate in some cases.

- Finally, we proposed two natural modifications to our policy for dealing with sequence-dependent systems and systems with raw-material shortages. The first one consists of using sequence-dependent hedging points, while the second one uses a time-dependent upper hedging point.

Appendix A

Supporting Proofs for Chapter 3

A.1 Proof of the Condition for Producing all Three Part Types

In this section, we prove that the inequalities given by (3.3) in Section 3.1 are necessary and sufficient for an $N = 3$ system to produce all part types.

We will restrict attention to the truncated system $\Sigma_\pi^\bullet$, which contains only the two highest-priority part types, types 1 and 2. It suffices to show then that the limit cycle of any trajectory of $\Sigma_\pi^\bullet$ is inside $\mathcal{Z}_2$, since that implies that this set will be visited infinitely often and thus that part type 3 will be produced infinitely many times by Σ_π.

Towards this end, let $\boldsymbol{x} = (\,x_1, x_2\,)^{\mathrm{T}}$ denote the surplus vector of $\Sigma_\pi^\bullet$ and suppose $\boldsymbol{x}_0 = \left(\,Z_1^{\mathrm{U}}, x_2(t_0)\,\right)^{\mathrm{T}}$, for some $x_2(t_0) \leq Z_2^{\mathrm{U}}$. Using (3.7) and (3.6), we have that the state at the end of the next run, $\boldsymbol{x}_1$, will be

$$\boldsymbol{x}_1 = \left(Z_1^{\mathrm{U}} - \frac{(S_{12} + (Z_2^{\mathrm{U}} - x_2(t_0))\tau_2}{1 - \rho_2} d_1,\, Z_2^{\mathrm{U}} \right)^{\mathrm{T}}.$$

Using (3.7) and (3.6) once more to obtain the next discrete-time state, we have that

$$\boldsymbol{x}_2 = \left(Z_1^{\mathrm{U}},\, Z_2^{\mathrm{U}} - \left[\frac{S_{21}}{1-\rho_1} + \left(\frac{S_{12} + (Z_2^{\mathrm{U}} - x_2(t_0))\tau_2}{1-\rho_2} d_1 \right) \frac{\tau_1}{1-\rho_1} \right] d_2 \right)^{\mathrm{T}},$$

where we notice that $x_2(t_2)$ depends linearly on $x_2(t_0)$ and is given by

$$x_2(t_2) = Z_2^{\mathrm{U}} - \left(\frac{S_{21} d_2}{1-\rho_1} + \frac{S_{12}\rho_1 d_2}{(1-\rho_1)(1-\rho_2)} + \frac{Z_2^{\mathrm{U}} - x_2(t_0)}{(1-\rho_1)(1-\rho_2)} \rho_1\rho_2 \right). \tag{A.1}$$

As depicted on the right sketch of Fig. 3-3, whenever $x_2(t_2) > x_2(t_0)$, the system is approaching $\boldsymbol{Z}^{\mathrm{U}}$ along the line $x_1 = Z_1^{\mathrm{U}}$. Using (A.1), this inequality can be written as

$$Z_2^{\mathrm{U}} - x_2(t_0) > \left(\frac{S_{21} d_2}{1-\rho_1} + \frac{S_{12}\rho_1 d_2}{(1-\rho_1)(1-\rho_2)} + \frac{Z_2^{\mathrm{U}} - x_2(t_0)}{(1-\rho_1)(1-\rho_2)} \rho_1\rho_2 \right).$$

Rearranging terms and simplifying, we get that the trajectory will be approaching $\boldsymbol{Z}^{\mathrm{U}}$ along $x_1 = Z_1^{\mathrm{U}}$ for all initial states $x_2(t_0)$ that satisfy

$$Z_2^{\mathrm{U}} - x_2(t_0) > \frac{S_{21}(1-\rho_2) + S_{12}\rho_1}{1-\rho_1-\rho_2} d_2. \tag{A.2}$$

Therefore, if (A.2) is true for $x_2(t_0) = Z_2^{\mathrm{L}}$, we are guaranteed that any initial trajectory will reach $\mathcal{Z}_2$ along the line $x_1 = Z_1^{\mathrm{U}}$, which corresponds exactly to the second inequality in (3.3). Furthermore, by interchanging labels in (A.2), we obtain the equivalent condition for reaching $\mathcal{Z}_2$ along the line $x_2 = Z_2^{\mathrm{U}}$.

We conclude then that, as long as *any* of the two inequalities given (3.3) is satisfied, we are guaranteed that the limit cycle of $\Sigma_\pi^\bullet$ will lie inside $\mathcal{Z}_2$ and that therefore part type 3 of Σ_π will be produced infinitely often. On the other hand, if none of the two inequalities is satisfied, the limit cycle of $\Sigma_\pi^\bullet$ will necessarily lie outside of $\mathcal{Z}_2$, because all states inside this set will be pushed outwards. Thus, the condition for production of all three part types is also necessary.

Appendix B

Supporting Proofs for Chapter 4

B.1 Gallego's Result for the Optimal Service Level Equation (4.3)

The derivation presented here is based on Gallego (1990). Suppose that the long-term trajectory of type i is periodic with period T. Then, the contribution of this part type to cost I will be given by

$$I_i = \frac{1}{T}\left[\int_0^T \left(h_i x_i(t) \, [\![x_i(t) > 0]\!] - b_i x_i(t) \, [\![x_i(t) < 0]\!]\right) \mathrm{d}t\right],$$

where $x_i(t)$ corresponds to the long-term trajectory and $[\![\cdot]\!] = 1$ whenever the condition inside the double brackets is met.

Consider now a small perturbation δ that is constant over time and that leads to a perturbed cost $I_i + \Delta I_i(\delta)$ given by

$$I_i + \Delta I_i(\delta) = \frac{1}{T}\left[\int_0^T \left(h_i(x_i(t)+\delta) \, [\![x_i(t)+\delta > 0]\!] - b_i(x_i(t)+\delta) \, [\![x_i(t)+\delta < 0]\!]\right) \mathrm{d}t\right].$$

Subtracting I_i from the previous equation (and being careful with the terms inside the $[\![\cdot]\!]$) we find that, for $\delta > 0$, the perturbation $\Delta I_i(\delta)$ will be given by

$$\begin{aligned}\Delta I_i(\delta) = \frac{1}{T}\int_0^T \Big(& h_i\delta\,[\![x_i(t) > 0]\!] + h_i(x_i(t)+\delta)\,[\![x_i \in (-\delta, 0]]\!] \\ & - b_i\delta\,[\![x_i(t) < -\delta]\!] + b_i x_i(t)\,[\![x_i \in [-\delta, 0)]\!]\Big)\mathrm{d}t.\end{aligned} \tag{B.1}$$

We now divide by δ (B.1) and take the limit of $\Delta I_i/\delta$ as $\delta \to 0$. Noting that the limit of $h_i x_i(t)[\![x_i(t) \in (-\delta, 0]]\!]/\delta$ is 0 for any t, we have that

$$\dot{I}_i^+(0) = \frac{1}{T}\left[\int_0^T \left(h_i\,[\![x_i(t) \geq 0]\!] - b_i\,[\![x_i(t) < 0]\!]\right)\mathrm{d}t\right].$$

Similarly, we can obtain a perturbation $\Delta I_i(\delta)$ for $\delta < 0$, divide by δ, and take the limit, obtaining

$$\dot{I}_i^-(0) = \frac{1}{T}\left[\int_0^T \left(h_i\,[\![x_i(t) > 0]\!] - b_i\,[\![x_i(t) \leq 0]\!]\right)\mathrm{d}t\right].$$

Notice that the above equations imply that if the set $\{\, t \mid x_i(t) = 0 \,\}$ has positive measure over the cycle, the derivative of $I_i(\delta)$ with respect to δ will be discontinuous at 0. This can also be seen intuitively; if $x_i(t)$ spends some finite fraction of time T_1 at 0, then raising or lowering the trajectory by δ will increase costs by an amount proportional to T_1.

Consider now the optimal trajectory $x_i^*(t)$ and assume that it is not flat at 0 (the case where this is not true is discussed Section B.2). Then, it follows that $\dot{I}_i^+(0) = \dot{I}_i^-(0) = \dot{I}_i(0)$ and, due to the optimality of $x_i^*(t)$, we must have $\dot{I}_i(0) = 0$. Using the fact that $1 = [\![x_i(t) > 0]\!] + [\![x_i(t) < 0]\!]$ for all t, we thus obtain

$$\frac{1}{T}\left[\int_0^T \left((h_i + b_i)\,[\![x_i^*(t) > 0]\!] - b_i\right)\mathrm{d}t\right] = 0. \tag{B.2}$$

Finally, since

$$\theta_i^* = \frac{1}{T}\left[\int_0^T [\![x_i^*(t) > 0]\!]\,\mathrm{d}t\right],$$

we have from (B.2)

$$(h_i + b_i)\theta_i^* - b_i = 0,$$

which gives the desired expression (4.2).

Given that $I_i(\delta)$ is convex in δ, the above result implies that if a trajectory $x_i(t)$ is nowhere flat, we can always reduce I by shifting it up or down until the service level matches θ_i^*.

B.2 Optimal Service Level with Cruising (4.2)

In Section B.1, we showed that if the optimal trajectory $x_i^*(t)$ is not flat at 0, then its service level must be equal to the ratio $b_i/(h_i+b_i)$. On the other hand, when the optimal trajectory is flat at 0, the derivative of I with respect to the magnitude of the shifting perturbation δ is not continuous, and thus there is no longer a closed expression for θ_i^*.

Recall that in our discussions of Section 4.2.3 we showed that an optimal trajectory should only cruise (if at all) at 0. Thus, if $x_i^*(t)$ is flat at 0, the perturbation that we considered in Section B.1 (i.e., a constant shift over time) will result in a new trajectory that cruises at a non-zero level, which is not optimal. We can easily correct this though, by bringing down (or up) the flat portions of $x_i^*(t) + \delta$ that were originally at 0, so that they are back at the zero-surplus level (this perturbation was also discussed in connection with Fig. 4-6). The effect of this new perturbation is that now the right and left derivatives of $I_i(\delta)$ satisfy

$$\dot{I}_i^+(0) = \frac{1}{T}\left[\int_0^T \big(h_i\,[\![x_i^*(t) \geq 0]\!] - b_i\,[\![x_i^*(t) < 0]\!] - h_i\,[\![x_i^*(t) = 0]\!]\big)\,\mathrm{d}t\right],$$

and

$$\dot{I}_i^-(0) = \frac{1}{T}\left[\int_0^T \left(h_i\,[\![x_i^*(t) > 0]\!] - b_i\,[\![x_i^*(t) \le 0]\!] + b_i\,[\![x_i^*(t) = 0]\!]\right) \mathrm{d}t\right].$$

In the expression for $\dot{I}_i^+(0)$, by bringing back down the flat segments of $x_i^*(t) + \delta$ that were originally at 0 (i.e., before the perturbation), we reduced the rate of increase of inventory costs with δ by h_i times the measure of these segments. In the expression for $\dot{I}_i^-(0)$, if $\delta < 0$, by bringing back up the flat segments of $x_i^*(t) + \delta$ we reduced the backlog cost rate by b_i times the measure of these segments.

Note that now the right and left derivatives of $I_i(\delta)$ coincide and, thus, the optimal trajectory satisfies $\dot{I}_i(0) = 0$. Using the fact that $1 = [\![x_i^*(t) > 0]\!] + [\![x_i^*(t) = 0]\!] + [\![x_i^*(t) < 0]\!]$ for all t, we obtain

$$\frac{1}{T}\left[\int_0^T \left((h_i + b_i)\,[\![x_i^*(t) > 0]\!] - b_i + b_i\,[\![x_i^*(t) = 0]\!]\right) \mathrm{d}t\right] = 0,$$

or

$$(h_i + b_i)\theta_i^* - b_i\left(1 - \frac{1}{T}\left[\int_0^T [\![x_i^*(t) = 0]\!]\,\mathrm{d}t\right]\right) = 0.$$

B.3 Proof of the Optimality of the $c\mu$ Rule

We provide here a partial proof that the $c\mu$ rule optimizes recovery costs for the case in which $S_{ij} = 0$ and $K_{ij} = 0$ for all i, j. The reader is referred to Sethi and Thompson (2000) for background on the optimal control theory concepts used in the proof.

Consider a cost functional of the form

$$G = \int_0^T -\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}(t)\,\mathrm{d}s, \tag{B.3}$$

and note that the dynamics of the surplus deviations satisfy

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{d} - \boldsymbol{u}(t).$$

At any time t, the production rate $\boldsymbol{u}(t)$ must be such that $\mathbf{0} \leq \boldsymbol{u}(t) \leq \boldsymbol{\mu}$. Given that setups are negligible, we allow $\boldsymbol{u}(t)$ to have more than one positive component at any given time. However, we do enforce the capacity constraint $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{u}(t) \leq 1$. These constraints on the control vector $\boldsymbol{u}(t)$, together with the state constraint $\boldsymbol{y}(t) \geq \mathbf{0}$, complete the specification of a finite horizon optimal control problem with pure state and control constraints.

Now, we note that the time it takes to clear an initial surplus deviation vector $\boldsymbol{y}(0)$ is policy-independent, provided that the policy utilizes all available capacity whenever $\boldsymbol{y}(t) \neq \mathbf{0}$; such a policy is called non-idling or work-conserving (see Chen and Yao 2001, p. 127). To see this, define the Lyapunov function $V(t) = \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}(t)$ and observe that

$$\dot{V}(t) = \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{d} - \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{u}(t) = \rho - \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{u}(t).$$

If the policy is work-conserving, then $\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{u}(t) = 1$ for all t such that $\boldsymbol{y}(t) > \mathbf{0}$. Thus, $\dot{V}(t) = \rho - 1$ during those times, and we will have that

$$V(t) = (\rho - 1)t + \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}(0).$$

Setting $V(T)$ equal to 0 and solving for T we get that

$$T = \frac{\boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{y}(0)}{1 - \rho}, \tag{B.4}$$

which shows that the system is able to clear any $\boldsymbol{y}(0)$ in finite time as long as $\rho < 1$. Furthermore, the long-term average cost J will be 0, which implies that the cost function G

defined in (B.3) will be equal to the negative recovery costs C_J.

In order to find the control that maximizes G (i.e., that minimizes C_J), we form the Hamiltonian

$$H = -\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\lambda}^{\mathrm{T}}(\boldsymbol{d} - \boldsymbol{u}),$$

where $\boldsymbol{\lambda}$ satisfies the adjoint equation

$$\dot{\boldsymbol{\lambda}}(t) = -\boldsymbol{c}$$

and $\boldsymbol{\lambda}(T) = \mathbf{0}$.

Solving the adjoint equation, we find that $\boldsymbol{\lambda}(t) = (T-t)\boldsymbol{c}$, implying that the optimal policy $\boldsymbol{u}^*(t)$ must satisfy for $t \leq T$

$$\boldsymbol{u}^*(t) = \operatorname{argmax}\left[-\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}^* - (T-t)\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{d} - \boldsymbol{u})\right] = \underset{\boldsymbol{c}}{\operatorname{argmax}}{}^{\mathrm{T}}\boldsymbol{u},$$

where the maximization is over all $\boldsymbol{u}(t)$ that meet the production rate and state constraints.

We see then that the control that maximizes H without violating the constraints will consist of producing the part type with the largest $c\mu$ coefficient first, until its surplus deviation is cleared. It will then hold that deviation at zero and utilize the rest of the capacity for clearing the deviation of the second largest $c\mu$ coefficient, and so on. That is, assuming that parts are ordered so that $c_1\mu_1 \geq c_2\mu_2 \geq \cdots \geq c_N\mu_N$, the optimal policy satisfies for all $t \in [0, T]$

$$u_i^*(t) = \begin{cases} d_1[\![y_1(t) = 0]\!] + \mu_1[\![y_1(t) > 0]\!] & \text{if } i = 1 \\ d_i\,[\![y_i(t) = 0]\!] + (1 - \sum_{j=1}^{i-1}\rho_j)\mu_i[\![y_i(t) > 0]\!] & \text{if } i > 1 \text{, and } y_j(t) = 0 \text{ for all } j < i \\ 0 & \text{otherwise.} \end{cases}$$

(Recall that $[\![\cdot]\!]$ is equal to 1 whenever the condition inside the double brackets; otherwise it equals 0.)

To complete the proof, we would need to verify that the above policy satisfies the rest of the necessary conditions for the optimal control problem (see Sethi and Thompson 2000, page 107). The optimality then follows from the linearity of the cost and constraint functions.

Note that if we wanted to implement $\boldsymbol{u}^*(t)$ with the restriction that at most one production rate component may be positive at all times, the policy will produce chattering (i.e., rapid fluctuations in the values of the control) after the surplus deviation of type 1 reaches 0. This highlights the difficulty of extending the $c\mu$ policy to systems with setups; trying to follow $\boldsymbol{u}^*(t)$ in a less than perfectly-flexible system leads to instability due to the high frequency of changeovers that the control generates.

Also note that, because of the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule, the HZP policy *does not* reduce to the control $\boldsymbol{u}^*(t)$ stated above as setup times become negligible and $\boldsymbol{\Delta Z} = \mathbf{0}$. In the HZP, a decision to produce item i during the interval $t + \delta t$ not only affects the accumulated cost during that time, but also implies a commitment to continue producing this item until $y_i(t) = 0$. This implies that the myopic selection of changeovers based on the $c\mu$ rule will likely not be always optimal under the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule. Nevertheless, the experimental results of Fig. 4-17 suggest that using the $c_i\mu_i$ indices for selecting changeovers is still a good heuristic for CC Class policies. Moreover, we can also see from our derivations that if $\boldsymbol{d} = \mathbf{0}$, the optimal policy will be clearing and will sequence the production of the items based on their $c_i\mu_i$ coefficients.

Appendix C

Supporting Proofs for Chapter 5

C.1 Proof of the Convergence of $\hat{\mathbb{X}}^{r_n}(t,\omega)$

The proof follows a standard argument in the theory of weak convergence (see Billingsley 1999, p. 80). Consider first the set of continuous functions on $[0,T]$ comprising all scaled allocation processes $T_i^r(t,\omega)$, for $r>0$ and with ω belonging to some probability space. Note that for any $0<s<t\leq T$ we have

$$0 \leq T_i(t,\omega) - T_i(s,\omega) \leq t-s,$$

which implies that the modulus of continuity $w_r(\delta)$ for any member of this set satisfies

$$w_r(\delta) = \sup_{|t-s|\leq\delta} |T_i^r(t,\omega) - T_i^r(s,\omega)| \leq \delta.$$

Thus, the set of functions $T_i^r(t,\omega)$ is uniformly *equicontinuous*. That is,

$$\lim_{\delta\to 0} \sup_{r>0} w_r(\delta) = 0. \tag{C.1}$$

Furthermore, since $T_i(0,\omega) = 0$, it follows that

$$\sup_{r>0} |T_i^r(0,\omega)| = 0. \tag{C.2}$$

By the *Arzelà-Ascoli Theorem*, it follows from (C.1) and (C.2) that the set of functions $T_i^r(t,\omega)$ is *relatively compact*. That is, for any sequence $\{\, T_i^{r_n}(t,\omega) \;;\; n \geq 0 \,\}$ there exists a subsequence $\left\{\, T_i^{r_{n_q}}(t,\omega) \;;\; q \geq 0 \,\right\}$ such that $T_i^{r_{n_q}}(t,\omega)$ converges uniformly to a continuous function in $[0, T]$.

Using the same argument, and assuming that, for all i,

$$\lim_{r\to\infty} \frac{|y_i(0,\omega)|}{r} < \infty,$$

we can see from (5.11)-(5.15) that the rest of the functions in the scaled model $\hat{\mathbb{X}}^{r_{n_q}}(t,\omega)$ will also converge to continuous limits uniformly on $[0, T]$.

C.2 Proof of Theorem 5.2.3

Let $\{\, r_n \;;\; n \geq 0 \,\}$ denote a convergent sequence of $\hat{\mathbb{X}}^{r_n}(t,\omega)$ for some sample[1] ω that satisfies (5.3) and denote the limit by $\overline{\mathbb{X}}(t) = (\overline{\boldsymbol{y}}(t), \overline{\boldsymbol{T}}(t), \overline{\boldsymbol{P}}(t), \overline{\boldsymbol{S}}(t), \overline{\boldsymbol{R}}(t))$. We now show that $\overline{\mathbb{X}}(t)$ can be transformed into a valid solution $\breve{\mathbb{X}}(t)$ whenever the Relaxed Stability Condition 3.3.10 is satisfied for the corresponding deterministic system. In particular, this solution is obtained by collapsing the production allocation and repair processes, that is,

$$\breve{\mathbb{X}}(t) = \left(\overline{\boldsymbol{y}}(t), \overline{\boldsymbol{T}}(t) + \overline{\boldsymbol{R}}(t), \overline{\boldsymbol{P}}(t), \overline{\boldsymbol{S}}(t)\right).$$

It follows directly from (5.11) and (5.12) that the solution satisfies (5.5) and (5.6). Further-

[1]For simplicity, we will omit ω in the subsequent expressions for $\hat{\mathbb{X}}(t)$.

more, by (5.3), we have that $\overline{T}_i(t) = e(\overline{T}_i(t) + \overline{R}_i(t))$ for all i. Therefore, $\overline{T}_i(t) = e\breve{T}_i(t)$ and thus (5.7) is also satisfied.

To prove (5.8), we first note that, because of uniform convergence, we have for any i and t

$$\begin{aligned}\dot{\breve{T}}_i(t) &= \lim_{h\to 0}\lim_{n\to\infty} \frac{T_i(r_n(t+h)) + R_i(r_n(t+h)) - T_i(r_n t) - R_i(r_n t)}{r_n h} \\ &= \lim_{n\to\infty}\lim_{h\to 0} \frac{T_i(r_n(t+h)) + R_i(r_n(t+h)) - T_i(r_n t) - R_i(r_n t)}{r_n h}.\end{aligned} \tag{C.3}$$

Therefore, if for some t_1 it is true that $\dot{\breve{T}}_i(t_1) > 0$, this implies that for any n sufficiently large we will have

$$T_i(r_n(t_1+h)) + R_i(r_n(t_1+h)) - T_i(r_n t_1) - R_i(r_n t_1) > 0 \quad \forall \quad h > 0.$$

By (5.14), this means that

$$T_i(r_n(t_1+h')) + R_i(r_n(t_1+h')) - \ T_i(r_n(t_1)) + R_i(r_n(t_1)) = h' r_n,$$

for any interval $(r_n t_1, r_n(t_1+h')]$ during which $y_i(t) > 0$. Therefore, we can see from (C.3) that this implies that $\dot{\breve{T}}_i(t)$ will be 1 during the interval starting after t_1 and ending at the instant when $\breve{y}_i(t)$ is cleared. This means that $\breve{T}_i(t) = \overline{T}_i(t) + \overline{R}_i(t)$ satisfies (5.8) for all i.

To show that (5.9) is satisfied, first note that (5.16) ensures that at least one run of some type i takes place. Also, note that since $\rho_i < \rho < e$ and because of the $\boldsymbol{Z}^{\mathrm{U}}$-Switch Rule, this run will get completed in finite time with probability 1. To see this, suppose the machine has completed the setup change into type i. During each subsequent uptime, the expected increase in the surplus will be $(\mu_i - d_i)$MTTF, while during the downtimes the expected decrease in the surplus will be d_iMTTR. Thus, the expected change in surplus over each uptime/downtime cycle will be μ_iMTTF $- d_i($MTTF + MTTR$)$. If $\rho_i < e$, this expected change is positive and the surplus during the production run evolves as a random walk with positive drift, which implies that it will reach Z_i^{U} with probability one.

Now let a Lyapunov function for the reduced, $(N-1)$-part-type system be given by

$$V^{\bullet}(t) = \sum_{i=1}^{N-1} \phi_i y_i(t),$$

and consider a period of time $[r_n t, r_n(t+h)]$ during which type N is not produced and $\boldsymbol{y}(t) \nleq \boldsymbol{\Delta Z}$. Suppose that during this period, there are Q_i *complete* runs of type $i < N$ and let t_i^q denote the start times of each of these runs, for $q = 1, 2, \ldots, Q_i$. These complete runs produce a change in $V_i^{\bullet}$ given by

$$\Delta V_i^{\bullet} = \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d}(\Delta S_i + \Delta R_i) - (\phi_i\mu_i - \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d})\left[\frac{\Delta S_i\rho_i + \Delta R_i\rho_i + \sum_{q=1}^{Q_i} y_i(t_i^q)\tau_i}{1-\rho_i}\right], \tag{C.4}$$

where $\boldsymbol{\phi} = (\,\phi_1, \phi_2, \ldots, \phi_{N-1}, 0\,)^{\mathrm{T}}$, and ΔS_i and ΔR_i denote the change in setup time and repair time with product i during the period of time comprising the Q_i runs (i.e., $\Delta S_i = S_i(r_n(t+h)) - S_i(r_n t)$ and $\Delta R_i = R_i(r_n(t+h)) - R_i(r_n t)$). Equation (C.4) follows from the fact that the Lyapunov function increases at rate $\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d}$ during the non-production periods of each run of i (note that ϕ_N is set to 0), which have a total duration of $\Delta S_i + \Delta R_i$. Furthermore, during the production periods of type i, the Lyapunov function decreases at rate $\phi_i\mu_i - \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{d}$, and the total production period of this item comprises the time it takes to clear the demand accumulated during setups and repairs, plus the time to clear the deviations $y(t_i^q)$ at the beginning of each of the Q_i runs.

Note that the term inside the brackets in (C.4) corresponds then to type i's production allocation time during the period in consideration, which we denote by ΔT_i. Furthermore, because of (5.3), we must have with probability one that

$$\Delta T_i = e(\Delta T_i + \Delta R_i) + o(r_n),$$

where $o(r_n)/r_n \to 0$ as $r_n \to \infty$. Therefore, equating the term in the brackets in (C.4) to ΔT_i and using the above equation to substitute ΔR_i, we get the expression

$$\Delta T_i = \frac{\Delta S_i \rho_i + \Delta T_i \frac{1-e}{e} \rho_i + \sum_{q=1}^{Q_i} y_i(t_i^q) \tau_i}{1 - \rho_i} + o(r_n),$$

from where we find that

$$\Delta T_i = \frac{\Delta S_i \rho_i + \sum_{q=1}^{Q_i} y_i(t_i^q) \tau_i}{1 - \rho_i / e} + o(r_n). \tag{C.5}$$

We now substitute (C.5) into (C.4) to get that

$$\begin{aligned}
\Delta V_i^\bullet = \boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{d} &\left[\Delta S_i + \frac{\Delta S_i \rho_i + \sum_{q=1}^{Q_i} y_i(t_i^q) \tau_i}{1 - \rho_i / e} \frac{1-e}{e} \right] \\
&- (\phi_i \mu_i - \boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{d}) \frac{\Delta S_i \rho_i + \sum_{q=1}^{Q_i} y_i(t_i^q) \tau_i}{1 - \rho_i / e} + o(r_n) \\
= \boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{d} \Delta S_i - \phi_i \mu_i &\left[\frac{\Delta S_i \rho_i + \sum_{q=1}^{Q_i} y_i(t_i^q) \tau_i}{1 - \rho_i / e} \right] + \boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{d} \left[\frac{\Delta S_i \rho_i + \sum_{q=1}^{Q_i} y_i(t_i^q) \tau_i}{e(1 - \rho_i / e)} \right] + o(r_n).
\end{aligned}$$

Let $\tilde{\rho}_i = \rho_i / e$. We can rearrange terms in the above equation into an expression that resembles (3.8), namely

$$\Delta V_i^\bullet = \frac{\boldsymbol{\phi}^T \boldsymbol{d}\, \tau_i / e - \phi_i}{1 - \tilde{\rho}_i} \sum_{q=1}^{Q_i} y_i(t_i^q) + \frac{\boldsymbol{\phi}^T \boldsymbol{d} - \phi_i d_i}{1 - \tilde{\rho}_i} \Delta S_i + o(r_n).$$

Now, recall that the coefficients ϕ_i are such that $\boldsymbol{\phi}^{\mathrm{T}} \boldsymbol{d} \tau_i / e - \phi_i < 0$, as required by (5.9) for all i. Furthermore, since $\boldsymbol{y}(t) \not\leq \boldsymbol{\Delta Z}$ during the period considered, we have by (5.15) that

$y_i(t_i^q) > \Delta Z_i$ for all q. Therefore,

$$\Delta V_i^\bullet \leq \frac{\boldsymbol{\phi}^T \boldsymbol{d}\,\tau_i/e - \phi_i}{1-\tilde{\rho}_i} Q_i \Delta Z_i + \frac{\boldsymbol{\phi}^T \boldsymbol{d} - \phi_i d_i}{1-\tilde{\rho}_i} Q_i S_i + o(r_n),$$

where we replaced all initial deviations $y_i(t_i^q)$ by ΔZ_i and the total setup time into i during the period, ΔS_i, by $Q_i S_i$. Ignoring the term $o(r_n)$, the right hand side in the expression above corresponds to Q_i times the change in Lyapunov function of a *deterministic* system with production rate $e\mu_i$, $N-1$ part types (since $\phi_N = 0$), and with runs that always start with the surplus at the lower hedging bound Z_i^{L}. Therefore, if the corresponding deterministic system satisfies Relaxed Stability Condition 3.3.10, $\Delta V_i^\bullet$ will be negative for all $i < N-1$ and for large enough r_n.

Now, recall that the interval $[r_n t, r_n(t+h)]$ was defined so that it corresponded to a period of time during which part type N is not produced. The change in the reduced system's Lyapunov function $V^\bullet(t)$ during this period will be given by the sum of $\Delta V_i^\bullet$ for $i = 1, 2, \ldots, N-1$, plus the change in $V^\bullet(t)$ during the two possibly incomplete runs at each endpoint of the interval. Since, as we showed earlier, all runs have finite duration w.p.1, the length of these two incomplete runs will be $o(r_n)$. Therefore, during this interval, we have $T_N(r_n(t+h)) + R_N(r_n(t+h)) - T_N(r_n t) - R_N(r_n t) = 0$ and

$$V^\bullet(r_n(t+h)) - V^\bullet(r_n t) \leq \sum_{i=1}^{N-1} \Delta V_i^\bullet + o(r_n),$$

which is negative for sufficiently large r_n. Thus, using a similar expression as (C.3), we see that the above result implies that $V^\bullet(t)$ will converge to some continuous function whose derivative is negative whenever $\dot{\breve{T}}_N(t) = 0$. This shows that (5.9) is satisfied.

We conclude by verifying that (5.10) is satisfied. Notice that if for some t_1 it is true that $\breve{y}_N(t_1) = 0$ and $\sum_{i=1}^{N-1} \breve{y}_i(t_1) > 0$, this implies then that there exists some finite A_N and m

such that $y_N(r_n t_1) < A_N$ and $\boldsymbol{y}(r_n t_1) \nleq \boldsymbol{\Delta Z}$, for all $n \geq m$. Equation (5.14) implies that if part type N was being produced at time $r_n t_1$, its production will continue until its deviation is cleared. Since $\rho_N < e$, the time for this to occur will be $o(r_n)$, and thus it follows from (5.15) that $T_j(t) + R_j(t)$ will start increasing for some $j < N$ at time $t = r_n t_1 + o(r_n) + S_j$. We conclude then that $\dot{\breve{T}}_j(t_1)$ will jump to 1 and consequently $\dot{\breve{T}}_N(t_1)$ jumps to 0 as required.

Appendix D

Experiments Implementation Details

D.1 Gallego's Recovery Policy

We provide in this section the equations needed to implement GRP. For the derivation of these equations, the reader is referred to the work by Gallego (1990). Without loss of generality, **we will assume unit demand rates** in this section; that is, $d_i = 1$ for all i. (Note that this can always be done by scaling the "material" units.)

Recall that $\boldsymbol{f}$ denotes a sequence of M positions that includes at least once every item $i = 1, \dots, N$ and that no item is repeated in two consecutive positions. In the definitions to follow, $\mathrm{diag}(\boldsymbol{\mu})$ denotes an $N \times N$ matrix with $\boldsymbol{\mu}$ in its diagonal and zeroes otherwise, and $[\boldsymbol{A}]_{ij}$ denotes the i, j entry of matrix $\boldsymbol{A}$. All unspecified entries of a matrix are taken as zero, and δ_{ij} denoes the Kronecker delta. The N-dimensional vector of ones is denoted by $\boldsymbol{e}$. Assume all matrices have the appropriate sizes.

$$
\begin{aligned}
[\boldsymbol{F}]_{ij} &= 1 \quad \text{iff } f^i = j, \text{ for } i = 1, \ldots, M, \text{ and } j = 1, \ldots, N \\
\boldsymbol{Q} &= \operatorname{diag}(\boldsymbol{\mu}) - \boldsymbol{e}\boldsymbol{e}^{\mathrm{T}} \\
[\boldsymbol{R}]_{ij} &= [\boldsymbol{F}\boldsymbol{Q}\boldsymbol{F}^{\mathrm{T}}]_{ij} \quad \text{if } j \leq i \\
[\boldsymbol{S}]_{ij} &= [\boldsymbol{F}\boldsymbol{Q}\boldsymbol{F}^{\mathrm{T}}]_{ij} \quad \text{if } j < i \\
[\boldsymbol{B}]_{ii} &= \frac{b_i \mu_i}{\mu_i - 1}, \quad \text{for } i = 1, \ldots, N \\
[\boldsymbol{H}]_{ii} &= \frac{h_i \mu_i}{\mu_i - 1}, \quad \text{for } i = 1, \ldots, N \\
[\hat{\boldsymbol{B}}]_{ij} &= \boldsymbol{F}\boldsymbol{B}\boldsymbol{F}^{\mathrm{T}} \delta_{ij} \\
[\hat{\boldsymbol{H}}]_{ij} &= \boldsymbol{F}\boldsymbol{H}\boldsymbol{F}^{\mathrm{T}} \delta_{ij} \\
\boldsymbol{C} &= \boldsymbol{F}^{\mathrm{T}}(\hat{\boldsymbol{B}} + \hat{\boldsymbol{H}})\boldsymbol{F} \\
\boldsymbol{D} &= -\boldsymbol{F}^{\mathrm{T}}(\hat{\boldsymbol{B}}\boldsymbol{S} + \hat{\boldsymbol{H}}\boldsymbol{R})\boldsymbol{F} \\
\boldsymbol{E} &= \boldsymbol{S}^{\mathrm{T}}\hat{\boldsymbol{B}}\boldsymbol{S} + \boldsymbol{R}^{\mathrm{T}}\hat{\boldsymbol{H}}\boldsymbol{R}.
\end{aligned}
$$

Let $\boldsymbol{M}$ denote the solution to the Algebraic Matrix Riccati Equation

$$
\boldsymbol{M} = \boldsymbol{M} + \boldsymbol{C} - (\boldsymbol{M}\boldsymbol{Q}\boldsymbol{F}^{\mathrm{T}} - \boldsymbol{D})(\boldsymbol{E} + \boldsymbol{F}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{Q}\boldsymbol{F}^{\mathrm{T}})^{-1}(\boldsymbol{M}\boldsymbol{Q}\boldsymbol{F}^{\mathrm{T}} - \boldsymbol{D})^{\mathrm{T}}. \tag{D.1}
$$

The above equation is solved numerically using the SLICOT SB02OD routine (Benner et al. 1997). With this solution, the matrix $\boldsymbol{G}$ in Policy 4.1 is then given by

$$
\boldsymbol{G} = (\boldsymbol{E} + \boldsymbol{F}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{Q}\boldsymbol{F}^{\mathrm{T}})^{-1}(\boldsymbol{M}\boldsymbol{Q}\boldsymbol{F}^{\mathrm{T}} - \boldsymbol{D})^{\mathrm{T}}. \tag{D.2}
$$

D.2 Datasets

Table D.1: Parameters for the system of Fig. 3-4b

	1	2	3	4	5
μ_i	1.0	1.0	1.0	1.0	1.0
d_i	0.2	0.1	0.12	0.16	0.13
S_i	30	140	40	100	10
$\boldsymbol{\Delta Z}_i$	-	-	70	70	70
$\mathsf{P}(i)$	5	4	3	2	1

Table D.2: Parameters for the system of Fig. 3-4c and Sec. 6.1.1, from Bomberger (1966).

	1	2	3	4	5	6	7	8	9	10
μ_i (items/day)	30000	8000	9500	7500	2000	6000	2400	1300	2000	15000
d_i (items/day)	400	400	800	1600	80	80	24	340	340	400
S_i (hours)	1	1	2	1	4	2	8	4	6	1
K_i (\$)	15	20	30	10	110	50	310	130	200	5
Piece cost (\$/item)	0.0065	0.1775	0.1275	0.1	2.785	0.2675	1.5	5.9	0.9	0.04
$\boldsymbol{\Delta Z}_i$	70785	15059	30617	26967	3927	8760	4967	5156	17662	18311
$\mathsf{P}(i)$	1	5	4	3	9	6	8	10	7	2

Multiply piece cost by \$0.1/\$-year to obtain c_i or h_i. A year consists of 240 days, with 8 work hours each.

Table D.3: Dataset 1

Instance	ρ_i/ρ	$c_i\mu_i/c\mu$ or $h_i\mu_i/h\mu$	S_i/S
1	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333
2	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333	0.1111, 0.1111, 0.7778
3	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333	0.1111, 0.4444, 0.4444
4	0.3333, 0.3333, 0.3333	0.1111, 0.1111, 0.7778	0.3333, 0.3333, 0.3333
5	0.3333, 0.3333, 0.3333	0.1111, 0.1111, 0.7778	0.1111, 0.1111, 0.7778
6	0.3333, 0.3333, 0.3333	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111
7	0.3333, 0.3333, 0.3333	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444
8	0.3333, 0.3333, 0.3333	0.1111, 0.1111, 0.7778	0.4444, 0.4444, 0.1111
9	0.3333, 0.3333, 0.3333	0.1111, 0.4444, 0.4444	0.3333, 0.3333, 0.3333
10	0.3333, 0.3333, 0.3333	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778
11	0.3333, 0.3333, 0.3333	0.1111, 0.4444, 0.4444	0.1111, 0.4444, 0.4444
12	0.3333, 0.3333, 0.3333	0.1111, 0.4444, 0.4444	0.7778, 0.1111, 0.1111
13	0.3333, 0.3333, 0.3333	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444
14	0.1111, 0.1111, 0.7778	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333
15	0.1111, 0.1111, 0.7778	0.3333, 0.3333, 0.3333	0.1111, 0.1111, 0.7778
16	0.1111, 0.1111, 0.7778	0.3333, 0.3333, 0.3333	0.1111, 0.7778, 0.1111
17	0.1111, 0.1111, 0.7778	0.3333, 0.3333, 0.3333	0.1111, 0.4444, 0.4444
18	0.1111, 0.1111, 0.7778	0.3333, 0.3333, 0.3333	0.4444, 0.4444, 0.1111
19	0.1111, 0.1111, 0.7778	0.1111, 0.1111, 0.7778	0.3333, 0.3333, 0.3333
20	0.1111, 0.1111, 0.7778	0.1111, 0.1111, 0.7778	0.1111, 0.1111, 0.7778
21	0.1111, 0.1111, 0.7778	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111
22	0.1111, 0.1111, 0.7778	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444
23	0.1111, 0.1111, 0.7778	0.1111, 0.1111, 0.7778	0.4444, 0.4444, 0.1111
24	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111	0.3333, 0.3333, 0.3333
25	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111	0.1111, 0.1111, 0.7778
26	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111	0.1111, 0.7778, 0.1111
27	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111	0.1111, 0.4444, 0.4444
28	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111	0.7778, 0.1111, 0.1111
29	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111	0.4444, 0.1111, 0.4444
30	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111	0.4444, 0.4444, 0.1111
31	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444	0.3333, 0.3333, 0.3333
32	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778
33	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444	0.1111, 0.7778, 0.1111
34	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444	0.1111, 0.4444, 0.4444
35	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444	0.7778, 0.1111, 0.1111

Instance	ρ_i/ρ	$c_i\mu_i/c\mu$ or $h_i\mu_i/h\mu$	S_i/S
36	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444
37	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444	0.4444, 0.4444, 0.1111
38	0.1111, 0.1111, 0.7778	0.4444, 0.4444, 0.1111	0.3333, 0.3333, 0.3333
39	0.1111, 0.1111, 0.7778	0.4444, 0.4444, 0.1111	0.1111, 0.1111, 0.7778
40	0.1111, 0.1111, 0.7778	0.4444, 0.4444, 0.1111	0.1111, 0.7778, 0.1111
41	0.1111, 0.1111, 0.7778	0.4444, 0.4444, 0.1111	0.1111, 0.4444, 0.4444
42	0.1111, 0.1111, 0.7778	0.4444, 0.4444, 0.1111	0.4444, 0.4444, 0.1111
43	0.1111, 0.4444, 0.4444	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333
44	0.1111, 0.4444, 0.4444	0.3333, 0.3333, 0.3333	0.1111, 0.1111, 0.7778
45	0.1111, 0.4444, 0.4444	0.3333, 0.3333, 0.3333	0.1111, 0.4444, 0.4444
46	0.1111, 0.4444, 0.4444	0.3333, 0.3333, 0.3333	0.7778, 0.1111, 0.1111
47	0.1111, 0.4444, 0.4444	0.3333, 0.3333, 0.3333	0.4444, 0.1111, 0.4444
48	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778	0.3333, 0.3333, 0.3333
49	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778	0.1111, 0.1111, 0.7778
50	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778	0.1111, 0.7778, 0.1111
51	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778	0.1111, 0.4444, 0.4444
52	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778	0.7778, 0.1111, 0.1111
53	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778	0.4444, 0.1111, 0.4444
54	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778	0.4444, 0.4444, 0.1111
55	0.1111, 0.4444, 0.4444	0.1111, 0.4444, 0.4444	0.3333, 0.3333, 0.3333
56	0.1111, 0.4444, 0.4444	0.1111, 0.4444, 0.4444	0.1111, 0.1111, 0.7778
57	0.1111, 0.4444, 0.4444	0.1111, 0.4444, 0.4444	0.1111, 0.4444, 0.4444
58	0.1111, 0.4444, 0.4444	0.1111, 0.4444, 0.4444	0.7778, 0.1111, 0.1111
59	0.1111, 0.4444, 0.4444	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444
60	0.1111, 0.4444, 0.4444	0.7778, 0.1111, 0.1111	0.3333, 0.3333, 0.3333
61	0.1111, 0.4444, 0.4444	0.7778, 0.1111, 0.1111	0.1111, 0.1111, 0.7778
62	0.1111, 0.4444, 0.4444	0.7778, 0.1111, 0.1111	0.1111, 0.4444, 0.4444
63	0.1111, 0.4444, 0.4444	0.7778, 0.1111, 0.1111	0.7778, 0.1111, 0.1111
64	0.1111, 0.4444, 0.4444	0.7778, 0.1111, 0.1111	0.4444, 0.1111, 0.4444
65	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444	0.3333, 0.3333, 0.3333
66	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444	0.1111, 0.1111, 0.7778
67	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444	0.1111, 0.7778, 0.1111
68	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444	0.1111, 0.4444, 0.4444
69	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444	0.7778, 0.1111, 0.1111
70	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444	0.4444, 0.1111, 0.4444
71	0.1111, 0.4444, 0.4444	0.4444, 0.1111, 0.4444	0.4444, 0.4444, 0.1111

Table D.4: Dataset 2

Instance	ρ_i/ρ	$c_i\mu_i/c\mu$ or $h_i\mu_i/h\mu$	S_i/S
1	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333
2	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333	0.0625, 0.0625, 0.8750
3	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333	0.0625, 0.4688, 0.4688
4	0.3333, 0.3333, 0.3333	0.0625, 0.0625, 0.8750	0.3333, 0.3333, 0.3333
5	0.3333, 0.3333, 0.3333	0.0625, 0.0625, 0.8750	0.0625, 0.0625, 0.8750
6	0.3333, 0.3333, 0.3333	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625
7	0.3333, 0.3333, 0.3333	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688
8	0.3333, 0.3333, 0.3333	0.0625, 0.0625, 0.8750	0.4688, 0.4688, 0.0625
9	0.3333, 0.3333, 0.3333	0.0625, 0.4688, 0.4688	0.3333, 0.3333, 0.3333
10	0.3333, 0.3333, 0.3333	0.0625, 0.4688, 0.4688	0.0625, 0.0625, 0.8750
11	0.3333, 0.3333, 0.3333	0.0625, 0.4688, 0.4688	0.0625, 0.4688, 0.4688
12	0.3333, 0.3333, 0.3333	0.0625, 0.4688, 0.4688	0.8750, 0.0625, 0.0625
13	0.3333, 0.3333, 0.3333	0.0625, 0.4688, 0.4688	0.4688, 0.0625, 0.4688
14	0.0625, 0.0625, 0.8750	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333
15	0.0625, 0.0625, 0.8750	0.3333, 0.3333, 0.3333	0.0625, 0.0625, 0.8750
16	0.0625, 0.0625, 0.8750	0.3333, 0.3333, 0.3333	0.0625, 0.8750, 0.0625
17	0.0625, 0.0625, 0.8750	0.3333, 0.3333, 0.3333	0.0625, 0.4688, 0.4688
18	0.0625, 0.0625, 0.8750	0.3333, 0.3333, 0.3333	0.4688, 0.4688, 0.0625
19	0.0625, 0.0625, 0.8750	0.0625, 0.0625, 0.8750	0.3333, 0.3333, 0.3333
20	0.0625, 0.0625, 0.8750	0.0625, 0.0625, 0.8750	0.0625, 0.0625, 0.8750
21	0.0625, 0.0625, 0.8750	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625
22	0.0625, 0.0625, 0.8750	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688
23	0.0625, 0.0625, 0.8750	0.0625, 0.0625, 0.8750	0.4688, 0.4688, 0.0625
24	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625	0.3333, 0.3333, 0.3333
25	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625	0.0625, 0.0625, 0.8750
26	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625	0.0625, 0.8750, 0.0625
27	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625	0.0625, 0.4688, 0.4688
28	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625	0.8750, 0.0625, 0.0625
29	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625	0.4688, 0.0625, 0.4688
30	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625	0.4688, 0.4688, 0.0625
31	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688	0.3333, 0.3333, 0.3333
32	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688	0.0625, 0.0625, 0.8750
33	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688	0.0625, 0.8750, 0.0625
34	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688	0.0625, 0.4688, 0.4688
35	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688	0.8750, 0.0625, 0.0625

Instance	ρ_i/ρ	$c_i\mu_i/c\mu$ or $h_i\mu_i/h\mu$	S_i/S
36	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688	0.4688, 0.0625, 0.4688
37	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688	0.4688, 0.4688, 0.0625
38	0.0625, 0.0625, 0.8750	0.4688, 0.4688, 0.0625	0.3333, 0.3333, 0.3333
39	0.0625, 0.0625, 0.8750	0.4688, 0.4688, 0.0625	0.0625, 0.0625, 0.8750
40	0.0625, 0.0625, 0.8750	0.4688, 0.4688, 0.0625	0.0625, 0.8750, 0.0625
41	0.0625, 0.0625, 0.8750	0.4688, 0.4688, 0.0625	0.0625, 0.4688, 0.4688
42	0.0625, 0.0625, 0.8750	0.4688, 0.4688, 0.0625	0.4688, 0.4688, 0.0625
43	0.0625, 0.4688, 0.4688	0.3333, 0.3333, 0.3333	0.3333, 0.3333, 0.3333
44	0.0625, 0.4688, 0.4688	0.3333, 0.3333, 0.3333	0.0625, 0.0625, 0.8750
45	0.0625, 0.4688, 0.4688	0.3333, 0.3333, 0.3333	0.0625, 0.4688, 0.4688
46	0.0625, 0.4688, 0.4688	0.3333, 0.3333, 0.3333	0.8750, 0.0625, 0.0625
47	0.0625, 0.4688, 0.4688	0.3333, 0.3333, 0.3333	0.4688, 0.0625, 0.4688
48	0.0625, 0.4688, 0.4687	0.0625, 0.0625, 0.8750	0.3333, 0.3333, 0.3333
49	0.0625, 0.4688, 0.4687	0.0625, 0.0625, 0.8750	0.0625, 0.0625, 0.8750
50	0.0625, 0.4688, 0.4687	0.0625, 0.0625, 0.8750	0.0625, 0.8750, 0.0625
51	0.0625, 0.4688, 0.4687	0.0625, 0.0625, 0.8750	0.0625, 0.4688, 0.4688
52	0.0625, 0.4688, 0.4687	0.0625, 0.0625, 0.8750	0.8750, 0.0625, 0.0625
53	0.0625, 0.4688, 0.4687	0.0625, 0.0625, 0.8750	0.4688, 0.0625, 0.4688
54	0.0625, 0.4688, 0.4687	0.0625, 0.0625, 0.8750	0.4688, 0.4688, 0.0625
55	0.0625, 0.4687, 0.4687	0.0625, 0.4688, 0.4688	0.3333, 0.3333, 0.3333
56	0.0625, 0.4687, 0.4687	0.0625, 0.4688, 0.4688	0.0625, 0.0625, 0.8750
57	0.0625, 0.4687, 0.4687	0.0625, 0.4688, 0.4688	0.0625, 0.4688, 0.4688
58	0.0625, 0.4687, 0.4687	0.0625, 0.4688, 0.4688	0.8750, 0.0625, 0.0625
59	0.0625, 0.4687, 0.4687	0.0625, 0.4688, 0.4688	0.4688, 0.0625, 0.4688
60	0.0625, 0.4688, 0.4688	0.8750, 0.0625, 0.0625	0.3333, 0.3333, 0.3333
61	0.0625, 0.4688, 0.4688	0.8750, 0.0625, 0.0625	0.0625, 0.0625, 0.8750
62	0.0625, 0.4688, 0.4688	0.8750, 0.0625, 0.0625	0.0625, 0.4688, 0.4688
63	0.0625, 0.4688, 0.4688	0.8750, 0.0625, 0.0625	0.8750, 0.0625, 0.0625
64	0.0625, 0.4688, 0.4688	0.8750, 0.0625, 0.0625	0.4688, 0.0625, 0.4688
65	0.0625, 0.4688, 0.4687	0.4688, 0.0625, 0.4688	0.3333, 0.3333, 0.3333
66	0.0625, 0.4688, 0.4687	0.4688, 0.0625, 0.4688	0.0625, 0.0625, 0.8750
67	0.0625, 0.4688, 0.4687	0.4688, 0.0625, 0.4688	0.0625, 0.8750, 0.0625
68	0.0625, 0.4688, 0.4687	0.4688, 0.0625, 0.4688	0.0625, 0.4688, 0.4688
69	0.0625, 0.4688, 0.4687	0.4688, 0.0625, 0.4688	0.8750, 0.0625, 0.0625
70	0.0625, 0.4688, 0.4687	0.4688, 0.0625, 0.4688	0.4688, 0.0625, 0.4688
71	0.0625, 0.4688, 0.4687	0.4688, 0.0625, 0.4688	0.4688, 0.4688, 0.0625

Bibliography

Adelman, D, C Barz. 2009. A unifying approximate dynamic programming model for the economic lot scheduling problem. *Working paper* .

Akella, R, P.R. Kumar. 1986. Optimal control of production rate in a failure prone manufacturing system. *IEEE Transactions on Automatic Control* **31**(2) 116–126.

Allahverdi, A, C.T. Ng, T.C.E. Cheng, M. Y. Kovalyov. 2008. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research* **187**(3) 985–1032.

Benner, P, V Mehrmann, V Sima, Sabine van Huffel, Andras Varga. 1997. *SLICOT-a subroutine library in systems and control theory*. Applied and Computational Control, Signals, and Circuits, Birkhäuser.

Bertsimas, D, D Gamarnik, J.N. Tsitsiklis. 1996. Stability conditions for multiclass fluid queueing networks. *Automatic Control, IEEE Transactions on* **41**(11) 1618–1631.

Bertsimas, Dimitris, Jose Nino-Mora. 1999. Optimization of multiclass queueing networks with changeover times via the achievable region approach: Part i, the single-station case. *Mathematics of Operations Research* **24**(2) 306–330.

Bielecki, T, P Kumar. 1988. Optimality of zero-inventory policies for unreliable manufacturing systems. *Operations Research* **36**(4) 532–541.

Billingsley, P. 1999. Convergence of probability measures. Wiley-Interscience.

Bomberger, Earl E. 1966. A dynamic programming approach to a lot size scheduling problem. *Management Science* **12**(11) 778–784.

Bramson, M. 2008. Stability of queueing networks. *Probability Surveys* **5** 169–345.

Burman, M. H. 1993. A real-time dispatch policy for a system subject to sequence-dependent, random setup times. Master's thesis, Massachusetts Institute of Technology.

Chase, C, P Ramadge. 1992. On real-time scheduling policies for flexible manufacturing systems. *IEEE Transactions on Automatic Control* **37**(4) 491–496.

Chen, H. 1995. Fluid approximations and stability of multiclass queueing networks: work-conserving disciplines. *The Annals of Applied Probability* **5**(3) 637–665.

Chen, H, David D Yao. 2001. Fundamentals of queueing networks. Springer Verlag, New York.

Conway, RW, WL Maxwell. 2003. Theory of scheduling. Dover Pubns.

Dai, J. 1995. On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *The Annals of Applied Probability* **5**(1) 49–77.

Dai, J, O Jennings. 2004. Stabilizing queueing networks with setups. *Mathematics of Operations Research* **29**(4) 891–922.

Dai, Jim. 1999. *Stability of fluid and stochastic processing networks*, vol. 15. MaPhySto Miscellanea Publication, Denmark.

Davis, Samuel G, Samuel G Davis. 1990. Scheduling economic lot size production runs. *Management Science* **36**(8) 985–998.

Dobson, G. 1987. The economic lot-scheduling problem: Achieving feasibility using time-varying lot sizes. *Operations Research* **35**(5) 764–771.

Dorf, Richard C, Robert H Bishop. 2005. *Modern Control Systems*. 10th ed. International Version, Pearson Prentice Hall.

Down, D, S Meyn. 1997. Piecewise linear test functions for stability and instability of queueing networks. *Queueing Systems* **27** 205–226.

Dudek, R, S Panwalkar, M Smith. 1992. The lessons of flowshop scheduling research. *Operations Research* **40**(1) 7–13.

Duenyas, I, M van Oyen. 1996. Heuristic scheduling of parallel heterogeneous queues with set-ups. *Management Science* **42**(6) 814–829.

Elhafsi, Mohsen, Sherman X Bai. 1997. Springerlink - journal of global optimization, volume 10, number 3. *Journal of Global Optimization* **10**(3) 283–303.

Elmaghraby, S. 1978. The economic lot scheduling problem (elsp): review and extensions. *Management Science* **24**(6) 587–598.

Federgruen, A, Z Katalan. 1996. The stochastic economic lot scheduling problem: cyclical base-stock policies with idle times. *Management Science* **42**(6) 783–796.

Fourer, Robert, David M Gay, Brian W Kernighan. 2003. *AMPL*. A Modeling Language for Mathematical Programming, Brooks/Cole.

Gallager, RG. 1996. Discrete stochastic processes. Kluwer Academic Publishers.

Gallego, G. 1990. Scheduling the production of several items with random demands in a single facility. *Management Science* **36**(12) 1579–1592.

Gallego, G. 1994. When is a base stock policy optimal in recovering disrupted cyclic schedules? *Naval Research Logistics* **41** 317–333.

Gallego, G, R Roundy. 1992. The economic lot scheduling problem with finite backorder costs. *Naval Research Logistics* .

Gallego, Guillermo, Dev Joneja, Guillermo Gallego, Dev Joneja. 1994. Economic lot scheduling problem with raw material considerations. *Operations Research* **42**(1) 92–101.

Gamarnik, David. 2002. On deciding stability of constrained homogeneous random walks and queueing systems. *Mathematics of Operations Research* **27**(2) 272–293.

Gershwin, S. 1989. Hierarchical flow control: a framework for scheduling and planning discrete events in manufacturing systems. *Proceedings of the IEEE* **77**(1) 195–209.

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

Gershwin, Stanley B. 2002. *Manufacturing Systems Engineering*. 2nd ed. Private Printing.

Gershwin, Stanley B, Ramakrishna Akella, Yong F. Choong. 1985. Short-term production scheduling of an automated manufacturing facility. *IBM Journal of Research and Development* **29**(4) 392–400.

Graves, S. 1980. The multi-product production cycling problem. *AIIE Transactions* **12**(3) 233–240.

Graves, S. 1981. A review of production scheduling. *Operations Research* **29**(4) 646–675.

Hopp, Wallace J, Mark L Spearman. 2008. Factory physics. Irwin/McGraw-Hill.

Hsu, W. 1983. On the general feasibility test of scheduling lot sizes for several products on one machine. *Management Science* **29**(1) 93–105.

Jones, Eric, Travis Oliphant, Pearu Peterson. 2001. Scipy.

Khmelnitsky, Eugene, Ernst Presman, Suresh P Sethi. 2009. Optimal production control of a failure-prone machine. *Annals of Operations Research* **182**(1) 67–86.

Kimemia, Joseph, Stanley B Gershwin. 1983. An algorithm for the computer control of a flexible manufacturing system. *IIE Transactions* **15**(4) 353–362.

Kumar, P, S Meyn. 1995. Stability of queueing networks and scheduling policies. *IEEE Transactions on Automatic Control* **40**(2) 251–260.

Kumar, P, T Seidman. 1990. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Transactions on Automatic Control* **35**(3).

Lan, W.M. 2000. Dynamic scheduling of multi-product systems: Bounds and heuristics. Ph.D. thesis, University of Michigan.

Lan, W.M., T. L. Olsen. 2006. Multiproduct systems with both setup times and costs: Fluid bounds and schedules. *Operations Research* **54**(3) 505–522.

Law, A, W Kelton. 1991. *Simulation modeling and analysis*. International edition ed. McGraw-Hill, Inc.

Levy, H., M. Sidi. 1990. Polling systems: applications, modeling, and optimization. *Communications, IEEE Transactions on* **38**(10) 1750–1760.

Little, John D C. 1961. A proof for the queuing formula: L= λW. *Operations Research* **9**(3) 383–387.

Liu, Z, P Nain, D Towsley. 1992. On optimal polling policies. *Queueing Systems* **11** 59–83.

Lou, S, S Sethi, G Sorger. 1991. Analysis of a class of real-time multiproduct lot schedulingpolicies. *IEEE Transactions on Automatic Control* **36**(2).

Lou, S, S Sethi, G Sorger. 1992. Stability of real-time lot scheduling policies for an unreliablemachine. *IEEE Transactions on Automatic Control* **37**(12) 1966–1970.

Luenberger, D. 1979. *Introduction to Dynamic Systems*. Theory, Models, and Applications, John Wiley & Sons Inc.

Maes, J, L Wassenhove. 1988. Multi-item single-level capacitated dynamic lot-sizing heuristics: A general review. *The Journal of the Operational Research Society* **39**(11) 991–1004.

Markowitz, D, M Reiman, L Wein. 2000. The stochastic economic lot scheduling problem: heavy traffic analysis of dynamic cyclic policies. *Operations Research* **48**(1) 136–154.

Markowitz, D, L Wein. 2001. Heavy traffic analysis of dynamic cyclic policies: A unified treatment of the single machine scheduling problem. *Operations Research* **49**(2) 246–270.

Maxwell, William L. 1964. The scheduling of economic lot sizes. *Naval Research Logistics* **11**(2) 89–124.

Meyn, SP. 2008. Control techniques for complex networks. Tata McGraw-Hill Education.

Michel, AN, L Hou, D Liu. 2008. *Stability of Dynamical Systems*. continuous, discontinuous, and discrete systems, Birkhäuser, Boston.

Moon, I, G Gallego, David Simchi-Levi. 1991. Controllable production rates in a family production context. *International Journal of Production Research* **29**(12) 2459–2470.

Moon, I, EA Silver. 2002. Hybrid genetic algorithm for the economic lot-scheduling problem. *International Journal of Production Research* **40**(4) 809–824.

Perez, A, P Zipkin. 1997. Dynamic scheduling rules for a multiproduct make-to-stock queue. *Operations Research* **45**(6) 919–930.

Perkins, J, PR Kumar. 1989. Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems. *IEEE Transactions on Automatic Control* **34**(2) 139–148.

Perkins, J, R Srikant. 1997. Scheduling multiple part-types in an unreliable single-machine manufacturing system. *IEEE Transactions on Automatic Control* **42**(3) 364–377.

Sethi, S, H Yan, H Zhang, Q Zhang. 2002. Optimal and hierarchical controls in dynamic stochastic manufacturing systems: A survey. *Manufacturing & Service Operations Management* **4**(2) 133–170.

Sethi, Suresh P, Gerald Luther Thompson. 2000. *Optimal Control Theory*. Second edition ed. applications to management science, Springer.

Sharifnia, A, M Caramanis, S Gershwin. 1991. Dynamic setup scheduling and flow control in manufacturing systems. *Discrete Event Dynamic Systems* **1** 149–175.

Simchi-Levi, David, Xin Chen, J Bramel. 2005. *The Logic of Logistics*. Theory, Algorithms, and Applications for Logistics and Supply Chain Management, Springer.

Sox, C, P Jackson, A Bowman, J Muckstadt. 1999. A review of the stochastic lot scheduling problem. *International Journal of Production Economics* **62**.

Srivatsan, N, Yves Dallery. 1998. Partial characterization of optimal hedging point policies in unreliable two-part-type manufacturing systems. *Operations Research* **46**(1) 36–45.

Stolyar, AL. 1995. On the stability of multiclass queueing networks: a relaxed sufficient condition via limiting fluid processes. *Markov Processes and Related Fields* **1**(4) 491–512.

Takagi, Hideaki. 1988. Queuing analysis of polling models. *ACM Computing Surveys (CSUR)* **20**(1) 5–28.

Tang, L. 2005. A simple recovery strategy for economic lot scheduling problem: A two-product case. *International Journal of Production Economics* **98** 97–107.

Trigeiro, W, L Thomas, J McClain. 1989. Capacitated lot sizing with setup times. *Management Science* **35**(3) 353–366.

Vollmann, Thomas, William Berry, D Clay Whybark, F Robert Jacobs. 2005. *Manufacturing Planning and Control for Supply Chain Management*. Fifth edition ed. McGraw-Hill Professional.

Xie, Zhiyu. 2008. Multiple-part-type production scheduling for high volume manufacturing (time-based approach). Master's thesis, Massachusetts Institute of Technology.