To Wave Or Not To Wave? Order Release Policies for Warehouses with an Automated Sorter[1]

Jérémie Gallien[2] and Théophane Weber[3]

August 13, 2009

Abstract

Wave-based release policies are prevalent in warehouses with an automated sorter, and take different forms depending on how much waves overlap and whether the sorter is split for operating purposes. Waveless release is emerging as an alternative policy adopted by an increasing number of firms. While that new policy presents several advantages relative to waves, it also involves the possibility of gridlock at the sorter. Using an extensive dataset of detailed flow information from the warehouse of a leading US online retailer, we first develop a model with validated predictive accuracy for a warehouse operating under a waveless release policy. We then use that model to compute operational guidelines for dynamically managing the main control lever of that policy, with the goal of maximizing throughput while keeping the risk of gridlock under a specified threshold. Secondly, we leverage that model and dataset to perform through simulation a performance comparison of wave-based and waveless policies in this context. The waveless policy yields larger or equal throughput than the best performing wave-based policy with a lower gridlock probability in all scenarios considered. Waveless release policies thus appear to merit serious consideration by practitioners. Facilities using a non-overlapping wave policy should also consider overlapping waves or a split sorter policy.

1 Introduction

Efficiently fulfilling a high volume of small orders chosen from a large number of SKUs is critical to many online retailers, direct mail-order firms, and retail distributors shipping to many stores on a frequent basis. The most critical infrastructure component in these distribution systems is often an automated split-case sorter. It enables a labor-efficient fulfillment process whereby a large set of orders can be disaggregated into individual item picking instructions distributed simultaneously in several zones within a split-case picking area (*batch and zone picking*), each having its dedicated team of workers (*pickers*). Picking an item typically involves scanning its bar-code and moving it from a storage rack to a small container (*tote*) carried by a rolling cart. When full, totes are offloaded onto a conveyor belt system transporting them to the sorter's induction stations; sometimes this transport system includes an intermediate circulating loop where selected totes may be temporarily

[1] The question of this title is from Gilmore (2006b).

[3] Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02142. E-mail: jgallien@mit.edu

[3] Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02142. E-mail: theo_w@mit.edu

held for the purpose of reducing the sorting time of orders. At the induction stations, workers empty incoming totes by placing their items onto individual tilting trays. These trays are circulated along a loop located above some accumulation chutes temporarily assigned to individual orders, and drop items into the appropriate chutes. Finally, when all the items of an order have been dropped to a chute, dedicated workers (*packers*) place them into a cardboard box which is then moved to a downstream shipping area[4].

The present paper focuses on the problem of coordinating the flow of work for the fulfillment process just described through order release control. Two main types of policies are currently used in practice to address this challenge: a very widespread and relatively old one known as *wave picking*, and a more recent and emerging one which is called in contrast *waveless picking* (we defer a detailed description of these policies until our practice survey section §3). While proponents of the latter claim it can increase throughput, equipment utilization and labor productivity relative to the former, there is currently no published rigorous study in support of these claims. In addition, waveless picking appears more challenging to control, and creates the possibility of a congestion-induced collapse known as *gridlock*.

The research work to be described here leverages our collaboration with a leading US online retailer, who had switched all of its warehouses with an automated sorter from wave-based to waveless picking before our interaction began, and provided most of our field observations and data. While our partner reports observing significant performance improvements with waveless picking, it did not initially establish formal guidelines for how managers should dynamically control this new policy, and experienced gridlock more frequently than desired. Indeed, there were no published guidelines on the control of waveless picking in either the trade or the academic literature then, and this still appears to be true at the time of writing. These observations motivate the two research objectives pursued in this paper: (Objective 1) *Develop a quantitative model to generate prescriptive control guidelines for waveless picking*; and (Objective 2) *Leverage this model to conduct a rigorous performance comparison between wave-based and waveless picking.*

After a discussion of the related literature and our contributions in §2, we provide in §3 a more detailed survey of the current practice for order release control in warehouses with an automated sorter. We then describe in §4 our work on the first objective stated above,

[4] Some sorters rely on different mechanisms to drop items into the accumulation chutes, also many include two or more vertical levels of accumulation chutes which can be accessed by the same trays (Saenz 2002).

present a quantitative model describing wave-based picking in §5, and discuss in §6 the simulation experiments we performed in order to achieve our second objective. Concluding remarks are provided in §7, and the Online Appendix to this paper contains supporting material, including auxiliary results and detailed algorithm statements. Mathematical variables in capital letters refer throughout to random quantities, while those in lower case refer to deterministic quantities. Also, notations with an upper (resp. lower) bar refer to the maximum (resp. minimum) value in an index set or interval, variables in bold refer to vectors or control policies, and new terminology being defined appears in italics.

2 Literature Review

We focus here on papers specifically motivated by warehouses with an automated sorter (either a split-case sorter as in §1 or a case sorter), and refer the reader to de Koster et al. (2007) for a recent survey of the extensive work on other types of warehouses.

A first set of papers examines whether an automated sorter constitutes a justified design option. Using queueing models, Choe et al. (1992) compare the cycle times associated with the three strategies of single order picking, batch picking and batch zone (wave) picking with an automated sorter. With simulation, Petersen (2000) investigates these policies plus sequential zone picking, considering not just cycle time but also labor requirements. Finally, Russell and Meller (2003) develop a deterministic cost model to decide whether manual or automated sorting should be used with wave picking.

A second group relies on simulation models to determine various warehouse dimensioning and control parameters. Bozer and Sharp (1985) explore the impact of the number of sorter chutes and their storage capacity as well as the use of recirculation and the concentration of items from each order within a wave. Bozer et al. (1988) also investigate the throughput implications of the wave profile (size, distribution of items per order), the chute assignment policy and the degree to which consecutive waves are allowed to overlap. Finally, Johnson and Lofgren (1994) report the use of model decomposition when designing a new warehouse. Also relevant is Johnson and Meller (2002), which presents an analytical model predicting the throughput of induction stations used in split-case sorting operations.

A last set explores more specific operational problems motivated by the use of automated sorters. Armstrong et al. (1979) and Le-Duc and de Koster (2005) formulate mixed integer optimization models to compute order batches, taking into account both discrepancies in

wave completion times across zones and limitations of sorter capacity. Owyong and Yih (2006) present a heuristic for modifying pick lists to reduce order accumulation time. Finally, Meller (1997) describes integer programming models for assigning orders to sorter chutes, assuming that the sequence of incoming items is known. Relaxing this last assumption, Johnson (1998) develops a stochastic model predicting the impact of various chute assignment strategies on expected wave sorting time.

The present paper makes the following five contributions to the existing literature:

1. We provide (in §3) the first academic survey of the various order release control methods used in practice for warehouses with an automated sorter, based on field research and a review of both trade and academic literatures. In particular, our paper is the first academic study to describe waveless picking (in §3.2);

2. We describe (in §4) the first quantitative model predicting warehouse flow dynamics under waveless picking. This model is noteworthy because (i) it has been validated empirically with field data (see Figure 1 and discussion in §4.3.2); (ii) it supports the first quantitative study of whether and how gridlock can be avoided, a topic discussed in the trade literature (e.g., Bradley 2007, Holste 2008); and (iii) more generally, it captures the relationship between the release of picking orders and sorter congestion, which has been consistently described as an important yet unexplored research topic (Johnson 1998, Petersen 2000, Owyong and Yih 2006);

3. We state (in §4.4) the first optimization formulation for the problem of dynamically controlling a waveless picking policy, which is meaningful because such control has been characterized as challenging in the trade literature (Demery 2007, Bradley 2007). In addition, we develop (in §4.5) a numerical algorithm for solving this problem, which leads us to describe (in §4.6) the first quantitative prescriptive guidelines for controlling a waveless picking policy;

4. We present (in §5) a model predicting sorter dynamics under wave picking. This model is noteworthy because (i) it generates the first quantitative description of the detailed dynamics of packing labor and sorter utilization under various wave picking policies (see Figure 4 and discussion in §6.2.2); and (ii) its output exhibits the same qualitative features that are discussed in the trade literature (e.g., Hinojosa 2006, Bradley 2007);

5. We perform the first rigorous performance comparison between wave-based and waveless

picking policies (see Table 1 and discussion in §6.2). This comparison is admittedly partial, because while our model supports an examination of throughput, sorter utilization, gridlock probability and packing labor, it does not capture the important dimension of picking labor, which is a primary cost driver in many warehouses. However, the exercise is still managerially relevant because many facilities see a relatively large fraction of their demand occur in a relatively short period of time (e.g., Christmas), during which throughput becomes a more important consideration than picking efficiency.

3 Survey of Practice for Order Release Control

3.1 Wave-Based Release Policies The traditional approach for coordinating the flow of work in warehouses with an automated sorter is aptly referred to as *wave picking*. In its simplest form, it consists of releasing large batches of orders (the waves) in a sequential manner, so that picking work for a given wave can only start when all the items from the previous wave have been already picked (Choe et al. 1992, Petersen 2000). Likewise, items of a given wave are only released into the sorter when all the orders from the previous wave have been already sorted and/or packed (Armstrong et al. 1979, Meller 1997). This approach presents several benefits: (i) Using large wave sizes increases the density of items to be picked and thus picking labor productivity, at least at the beginning and in the middle of the wave (see discussion below); (ii) Pick lists can be determined for all the workers simultaneously at specific points in time, and can be communicated using simple paper printouts; and (iii) Blocking effects at the sorter can be completely avoided by ensuring that the number of orders in each wave is less than or equal to the number of sorter chutes.

However, many sources also discuss several important drawbacks associated with this simplest form of wave picking: (i) Because the time for pickers to complete a wave is variable, some pickers may experience idle time at the end of waves (Choe et al. 1992, Petersen 2000, Gilmore 2006a, Gilmore 2006b and Bradley 2007); (ii) Large wave sizes generate a large buffer of inventory (cycle stock) between picking and sorting, which is costly because of the resulting accumulation conveyor and floor space requirements (Russell and Meller 2003, Bradley 2007). Due to the link with the number of chutes, wave sizes also drive the sorter purchase cost; (iii) Waves also add a cycle time component to order completion times, which can be problematic with time-sensitive customers; (iv) The sequential release of

non-overlapping waves into the sorter results in low capacity utilization, because the chutes corresponding to completed orders cannot be re-assigned until the end of the current wave. This issue is particularly critical during peak periods because sorters often constitute the throughput bottleneck (Apple 2006, Gilmore 2006a, Perkins 2008); and (v) The packing workload is concentrated in the second half of each wave, as most chutes only become ready to be packed then (Hinojosa 1996, 2006, Perry 2007).

To mitigate these problems, more sophisticated forms of wave picking have been developed. To reduce pickers' idling at the end of waves, some companies allow different waves to overlap in the picking area, either across zones (Armstrong et al. 1979) or within each zone (Owyong and Yih 2006). However, this practice creates the need for a pre-sorting operation to separate items from different waves before release into the sorter. To increase throughput as well as sorter and packing labor utilization, different waves are sometimes also allowed to overlap in the sorter. A first strategy consists of starting the release of each wave as soon as the previous one has reached a specified completion threshold such as 90% (Bozer et al. 1988) or 50% (Johnson and Lofgren 1994) of the orders. As pointed out in Johnson (1998) however, overlapping waves in the sorter presents control challenges, because of the blocking that occurs if all the sorter accumulation space becomes full (see discussion below).

A second strategy consists of splitting the available sorter space in two halves, with each half dedicated to a different wave so that packers can work on a completed wave in one half while the next wave is being accumulated into the other half (Ruben and Jacobs 1992, Russell and Meller 2003, Perry 2007, Perkins 2008). As a result, packers wait little or no time for the next wave to complete accumulation, and their utilization is much improved. In practice, this policy typically leverages a physical sorter design where chutes are laid out in two vertical levels, and packers work on a completed wave in one of the two levels (e.g., the lower level), while the next wave is being sorted in the other (e.g., upper) level. While many systems allow both packers and tilting trays to access both levels independently, others only allow sorting/accumulation in the upper level and packing in the lower level, with a dropping mechanism allowing to transfer complete orders from the upper level to the lower level when the latter is empty[5]. In those systems where only the chutes of the lower level are accessible to packers, they are sometimes referred to as *pack-out only chutes* in contrast with

[5] We are grateful to a referee for providing a description of this alternative design.

the accumulation chutes of the upper level. While this split sorter strategy seems relatively widespread, note that for all sorters except those with pack-out only chutes it divides the largest possible wave size by two[6], which may impact picking labor productivity (Perry 2007).

3.2 Waveless Release Policy Because of the challenges described above, a growing number of companies are using an alternative order release control policy referred to as *waveless picking* or *continuous flow picking* (Bradley 2007). While the different implementations of this new policy vary in details (see Hinojosa 2006, Trebilcock 2007, McMahon 2008 and Morris 2008), they all involve the same core principle, which is perhaps best explained through a comparison with traditional wave picking.

Wave picking conceptually involves a first queue of incoming customer orders and a second picking queue corresponding to all the orders covered by the current active picking assignments; whenever the second queue becomes empty, it is replenished at once by an entire batch of a given number of orders (the wave), which is transferred then as a whole from the first queue. In contrast, waveless picking involves the continuous transfer of individual orders from the first queue to the second one, based on a priority ranking of incoming customer orders typically based on target shipping dates.

The second queue (called a revolving batch or a virtual wave) still has a maximum capacity, which is an important control parameter that we will later refer to as the *revolving batch size*; when that maximum buffer size is reached, any new customer order may only enter the picking queue as another one exits, which occurs when the last one of its items is picked. Pick lists for individual pickers are determined and continually updated in real-time from the picking queue, using a partition of the warehouse storage area into continuous directed picking loops (the zones), and a dynamic partition of each picking loop between all the pickers assigned to that zone. Specifically, every worker's pick list consists at all times of all the items from orders in the picking queue that are located between his last recorded position and that of the next picker down his picking loop. In addition, this method involves a labor balancing mechanism which continuously evaluates for each zone the expected completion time of the current picking queue, and re-assigns pickers whenever imbalances of this quantity across zones exceed specified thresholds.

[6] With sorters where all the chutes can be used for both accumulation and packing, as an alternative to this split sorter strategy one could release sequential waves with a (doubled) size equal to the total number of chutes in both vertical levels. In contrast, the maximum wave size in a sorter with pack-out only chutes remains equal to the number of upper accumulation chutes.

Note that the waveless policy just described critically relies on expensive technology and software, specifically dependable real-time two-way wireless digital communications with every picker in the warehouse (typically provided by portable devices also including a bar-code reader), and real-time centralized database management. The benefit of this new method however is that while it involves batch and zone picking and may therefore achieve high picking productivity (with the density of pick assignments determined by the revolving batch size), it also appears to eliminate some of the inefficiencies associated with wave picking. In particular, no picker is ever starved for work at the end of a wave and, relative to facilities allowing picking waves to overlap, there is no need for a pre-sorting operation. In addition, the rate at which orders become available for packing is more steady, and completed orders in sorter chutes never need to wait before they can be assigned to a packer. Finally, any urgent incoming order can be assigned for picking almost instantly without waiting until the end of the current wave, and the average completion time of all orders is improved by the elimination of the cycle times before picking and between picking and sorting that are introduced by waves. Indeed, several trade journal articles and corporate white papers point out that waveless picking may generate substantial improvements in both throughput and labor costs relative to wave picking (Hinojosa 2006, Cooke 2007, Perry 2007), and several support this claim with observations from actual implementations (Bradley 2007, Morris 2008, McMahon 2008).

An essential caveat however is that waveless picking no longer involves the release into the sorter of separate batches of a fixed number of orders, which assures that its accumulation space is never exceeded. This new policy therefore creates the potential for severe blocking (Bradley 2007). Specifically, when all the chutes in the sorter are tied up (either by incomplete orders or by complete orders waiting for a packer), upstream congestion can start to build. As a result, the very items needed to complete orders tying up chutes and relieve this congestion may no longer reach the sorter because of... the same congestion. This phenomenon is known as *gridlock* (Johnson and Lofgren 1994), and because the corresponding recovery procedure is typically long and laborious, it can significantly reduce capacity and productivity (Holste 2008). In the words of Sam Sanders, a warehouse consultant quoted in Bradley (2007): "It's like a game of solitaire. If all the slots in the game are full, the game is over, and you lose. If you have 10,000 SKUs and 1,200 drop points, you can have a lot of

SKUs on the sorter with no place to drop into. If you want to work with continuous flow, you have to be cognizant of this."

In the rest of this paper, we develop and analyze quantitative models of warehouse flow dynamics under both waveless (§4) and wave-based (§5) release control.

4 Waveless Release Model and Analysis

This section begins with a formal definition of our predictive model in §4.1 and a related discussion in §4.2. We then present approximate dynamics and discuss their validation in §4.3, state the related optimization problem we consider in §4.4, present an associated solution algorithm in §4.5 and finally discuss in §4.6 the qualitative features of the waveless release policies computed through this approach.

4.1 Model Definition Our waveless release model is a three station serial queueing network with state-dependent service rates and a dynamically controllable input rate. Each circulating entity in this network represents a customer order. The release of new orders follows a Poisson process, with a controllable rate at time $\tau \geq 0$ noted $\lambda(\tau)$. This arrival process corresponds in the real system to the sequence of times at which the first item of any order is picked across the entire picking area. Control $\lambda(\tau)$, which thus corresponds to the current average order picking rate, is limited for capacity reasons by a fixed upper bound $\bar{\lambda}$. Also, because of database synchronization issues this average release rate may only be changed at discrete time points separated by a period δ (of the order of a few minutes). Consequently, the release policies $\boldsymbol{\lambda}$ considered amount to a discrete sequence of controls $(\lambda_t)_{t\in\mathbb{N}}$, where each discrete time period $t \in \mathbb{N}$ corresponds to the continuous time interval $[t\delta, (t+1)\delta)$, i.e., $\lambda(\tau) = \lambda_t$ for $\tau \in [t\delta, (t+1)\delta)$.

The first queueing station has an infinite number of servers, each with identically distributed service times following a state-dependent distribution noted A and representing the *time-to-chute*, or delay between the time when the first item of an order is picked and the time when the first item from that order reaches a sorter chute. The process representing the number of orders undergoing service in this first station is denoted $X(\tau)$, and provides a partial measure of the conveyor congestion upstream of the sorter. In the following, we will use the notation $X_t \triangleq X(t\delta)$.

The second station has an infinite number of servers, each with identically distributed

service times following a state-dependent distribution noted B and representing the *chute-dwell time* of every order, which is the delay between the arrivals of its first and last items to a chute. The number of orders undergoing service in this second station thus represents the number of incomplete chutes in the sorter, and follows a process denoted $Y(\tau)$. As before, we define $Y_t \triangleq Y(t\delta)$.

The third station represents the packing stage. It has a finite number of servers equal to the number w of packers assigned to the sorter, each with identically distributed service times representing the *pack-to-pack time* C, or cycle time experienced by a packer for each customer order (e.g., time spent walking to the next chute + time spent packing). The process representing the number of orders in this station (in queue and in service) is denoted $Z(\tau)$, which thus corresponds to the number of complete chutes in the sorter at any point in time (either complete and waiting for a packer, or being packed). Its values at the discrete time points $(t\delta)_{t\in\mathbb{N}}$ are also denoted $Z_t \triangleq Z(t\delta)$. Note that $Y(\tau) + Z(\tau)$ thus represents the total number of busy chutes at any time τ, so that the occurence of gridlock can be characterized as the event $Y(\tau) + Z(\tau) > n$ where n is the number of sorter chutes.

Finally, the state dependency of A and B captures the relationship between the actual time-to-chute and chute-dwell time of orders and the congestion upstream of the sorter, which is itself directly related to the processes $X(\tau)$ and $Y(\tau)$[7]. To capture this endogeneity, we consider a small number of congestion levels $g \in \{1, ..., \bar{g}\}$ corresponding to adjacent consecutive ranges $[d_g, d_{g+1})$ for conveyor system congestion, defined as the total number of items $I(\tau)$ on the conveyor system between the picking area and the sorter. With $\mathbb{E}[M]$ denoting the average number of items per customer order, we verified with field data that the expression $I(\tau) \approx \mathbb{E}[M](X(\tau) + Y(\tau)/2)$ provides an accurate estimate[8]. The last step is to specify fixed distributions $A(g)$ and $B(g)$ for all the congestion levels $g \in \{1, ..., \bar{g}\}$, respectively providing the service times of the first and second stations when the system state is such that $\mathbb{E}[M](X(\tau) + Y(\tau)/2) \in [d_g, d_{g+1})$. The Online Appendix illustrates how these distributions may be estimated using available field data, and contains empirical density plots of the service time distributions obtained with a specific data set.

4.2 Modeling Discussion

[7] The walking time of packers and therefore distribution C may also appear endogenous. However, field data shows that C is in fact fairly stationary.

[8] This expression corresponds to the approximation that the order statistics of the arrival times of the items of each order to their assigned chute are equally spaced in expectation

4.2.1 Scope and Control We observed in the field that the main daily flow control levers available as part of the waveless picking policy described in §3.2 include the size of the revolving batch, which can be adjusted a few times per hour, as well as the staffing levels for pickers and packers. Note that the revolving batch size directly affects the overall picking rate through the resulting density of items to be picked along the picking loops[9]. In practice, we have observed the use of simple but reliable data tables constructed empirically to determine the size of the revolving batch required to generate a specified overall average picking rate under various staffing levels for pickers. However, we saw that no formal guidelines were available for dynamically changing the target picking rate as a function of observed process conditions and packers' staffing level[10]. This motivates our choice of the target picking rate as the primary control, even if that control is to be effectively implemented through changes in the revolving batch size and pickers' staffing level. Also, our model considers the packers' staffing level as a fixed parameter, which seems justified because it is only changed a few times per day in practice.

As Russell and Meller (2003), we thus do not explicitly model the picking area layout, stowing policies and picker routing policies employed. Note however that we do consider their aggregate impact on downstream congestion, i.e., the overall rate at which pickers release items onto the conveyor system. Specifically, this impact is captured by the empirical data tables mentioned above, which link overall average picking rate with the revolving batch size and the pickers' staffing level. As Chew and Tang (1999) and Le-Duc and de Koster (2007) demonstrate, this relationship can also be derived analytically in some settings.

The main drawback of this approach is that our model does not allow to quantify the impact of waveless picking on picker productivity (see related discussion in §7). It presents however several important benefits: (i) our model remains tractable; (ii) this modular approach separates the problems of minimizing the labor cost of a target picking rate on one hand, and setting the picking rate to maximize warehouse throughput on the other hand – as a result we can better focus on the latter here; and (iii) our model is more widely applicable, because idiosynchratic features such as the layout of the picking area do not need to be

[9] Picking rate in this setting can be conceptually modeled by the throughput of a closed queueing network with a single station having as many servers as there are pickers, and where the number of circulating entities, which corresponds to the revolving batch size, affects the service rate.

[10] Instead, there were informal guidelines prescribing to try and stabilize the process around target numbers of complete and incomplete sorter chutes. These target numbers were not supported by analysis, appeared inconsistent across managers and facilities.

explicitly captured.

Finally, note that the arrival process in our model is random, which reflects that the actual system is only partially controllable. That is, specifying the revolving batch size for a given staffing level only implements an average picking rate, from which the current instantaneous picking rate may differ – this stems in practice from variability in the actual density of picks along the picking loops, pickers' individual productivity, the actual number of items per order, etc. The specific structure assumed for that randomness (Poisson) is motivated by both tractability considerations and the intuitive applicability of the Poisson superposition theorem to this setting[11].

4.2.2 Processing Our model of the order accumulation process is derived from Gallien and Wein (2001). Note that the conveyor transport system linking the picking area and the sorter is entirely captured by the congestion-dependent service time distributions $A(g)$ and $B(g)$. Similar to our modeling of the picking process, this approach relieves us from explicitly modeling idiosynchratic features such as the layout of the conveyor belt system and the possible use of an intermediate circulating loop (see §1). In the Online Appendix, we apply this approach to field data in order to illustrate that it can capture the key dynamics of a complex actual warehouse system.

In addition, we do not explicitly capture staffing decisions for induction stations (and more generally their capacity). This is justified by our field observations of induction stations with automated coordinated induction belts designed using realistic throughput models of the type described in Johnson and Meller (2002), so that only few operators are required to achieve a high induction capacity.

Finally, note that the number of servers of the second station is infinite and that the buffer size of the third station is unlimited, even though both parameters should be limited by the number of sorter chutes n. This is justified because (as will be seen in §4.4) we only consider release control policies $\boldsymbol{\lambda}$ ensuring with high probability that the gridlock event $\{Y(\tau) + Z(\tau) > n\}$ does not occur. Under such policies the seemingly salient model assumptions just stated are immaterial, because both events $\{Y(\tau) > n\}$ and $\{Z(\tau) > n\}$ are contained in the previous one.

[11] Many pickers simultaneously contribute to the overall picking rate, and the sequence of each worker's item picking times can be modeled as an independant renewal process. See e.g., Theorem 9.8.1 in Whitt (2002).

4.2.3 Input and Output Data The primary model data $(A, B, C, \bar{\lambda}, w)$ should be readily available in all warehouses using a modern warehouse control system (WCS), which involve bar-code scanners carried by pickers and packers as well as many sensors placed in many locations in the conveyor system, induction stations and sorter chutes. In addition, the primary model output $(X(\tau), Y(\tau), Z(\tau))$ is directly observable in practice, and this information corresponds in fact exactly to that actually observed by some managers when adjusting the revolving batch size. This model feature guarantees that the informational requirements of the release control policy developed are realistic, and also allows for an empirical model validation to be performed (see §4.3.2);

4.3 Approximate Dynamics

4.3.1 Derivation Our next step is to study the dynamics of the queueing model described in §4.1 as a function of the release control policy $\boldsymbol{\lambda}$. Unfortunately, an exact analysis appears challenging because (i) the short control time period δ precludes steady-state assumptions; and (ii) service time distributions for the first two stations are state-dependent. Instead, we develop here a more tractable approximate version of our queueing model and verify in §4.3.2 through a validation experiment that this version remains suitably realistic. We specifically consider the following approximations:

- *The service times $A(g)$, $B(g)$ and C of the three queueing stations follow exponential distributions with first moments given by actual data.* The empirical distributions of $B(g)$ and C constructed from data exhibit shapes that are similar to that of an exponential (see Online Appendix). However, the empirical distributions we constructed for $A(g)$ do not[12].
- *Orders move at most one station downstream during each time period $[t\delta; (t+1)\delta)$.* Because the actual expected service times $\mathbb{E}[A(g)]$ and $\mathbb{E}[B(g)]$ at the first and second stations are several times larger than the control period δ, the transitions that this assumption ignores have very low probability relative to all others.
- *The congestion level remains constant within each control period $[t\delta; (t+1)\delta)$.* We have found through simulation of the original system that consecutive changes of congestion level occuring less than δ time units apart were very rare.

[12] Weber (2005) derives more accurate system dynamics for this model using phase-type distributions and results on the $M_t/G/\infty$ queue from Eick et al. (1993). However, the optimization problem considered in §4.4 becomes much harder to solve under these dynamics.

– *The minimum of the numbers of packers w and closed chutes $Z(\tau)$ remains constant within each control period $[t\delta;(t+1)\delta)$.* We have likewise observed that policies performing well in simulations result in a relatively high capacity utilization for the third queue, yielding $\min(w, Z(\tau)) = w$ with high probability.

From elementary properties of Markovian queues, the above approximations result in the following discrete-time system dynamics (see Weber 2005 for this and other derivations of transient dynamics for this system)[13]:

$$\left\{\begin{array}{l} X_{t+1} = X_t + N_t^{\rightarrow X} - N_t^{X\rightarrow Y} \\ Y_{t+1} = Y_t + N_t^{X\rightarrow Y} - N_t^{Y\rightarrow Z} \\ Z_{t+1} = Z_t + N_t^{Y\rightarrow Z} - N_t^{Z\rightarrow} \\ g_t = \sum_{g=1}^{\bar{g}} g 1_{[d_g, d_{g+1})}(\mathbb{E}[M](X_t + \frac{Y_t}{2})) \end{array}\right. \quad \text{with} \quad \left\{\begin{array}{l} N_t^{\rightarrow X} \sim \text{Poisson}(\lambda_t \delta) \\ N_t^{X\rightarrow Y} \sim \text{Binom}(X_t, 1 - e^{-\frac{\delta}{\mathbb{E}[A(g_t)]}}) \\ N_t^{Y\rightarrow Z} \sim \text{Binom}(Y_t, 1 - e^{-\frac{\delta}{\mathbb{E}[B(g_t)]}}) \\ N_t^{Z\rightarrow} \sim \text{Poisson}(\frac{(w \wedge Z_t)\delta}{\mathbb{E}[C]}) \end{array}\right. , \tag{1}$$

where the four random variables $(N_t^{\rightarrow X}, N_t^{X\rightarrow Y}, N_t^{Y\rightarrow Z}, N_t^{Z\rightarrow})$ represent the number of customer orders that are respectively released into the first station, moved from the first to the second and the second to the third station, and processed out of the third station, between time periods t and $t+1$. An appealing feature is that simulating system (1) only involves generating four standard random variables in each time period, and can thus be performed very efficiently[14].

4.3.2 Validation We now validate the approximate queueing dynamics (1) with field data; our methodology is to compare the predicted model state under some given release rate and packer staffing history against that actually observed in the real system when subjected to the same input. Specifically, we rely on collected data series with one point per minute recording the actual state evolution $(x^*(\tau), y^*(\tau), z^*(\tau))$, actual control history $\lambda^*(\tau)$ and actual number of staffed packers $w^*(\tau)$ over a period of several days of peak demand. We use the control period δ (of the order of a few minutes) as our prediction lead-time, since the associated DP involves an expectation of the value function δ units of time into the future. We thus computed for every time τ the average release rate over the following period of length δ, $\tilde{\lambda}^*(\tau) = \frac{1}{\delta}\sum_{i=0}^{\delta-1}\lambda^*(\tau+i)$, and simulated the random variables $(X_{t+1}, Y_{t+1}, Z_{t+1})$ characterized by

[13] In particular, the last expression of (1) corresponds to the departure process over a period of length δ from a Markovian queue with w servers assumed to work continuously, except in the ramp-up periods.

[14] The binomial variables $N_t^{X\rightarrow Y}$ and $N_t^{Y\rightarrow Z}$ can be substituted with normal random variables having the same mean and variance, which from the De Moivre-Laplace theorem is asymptotically exact for the large values of X_t and Y_t that are typical of our setting.

(1) given $(X_t, Y_t, Z_t) = ((x^*(\tau), y^*(\tau), z^*(\tau))$, $\lambda_t = \tilde{\lambda}^*(\tau)$ and $w = w^*(\tau)$. We then compared them with the actual state for the corresponding period $(x^*(\tau + \delta), y^*(\tau + \delta), z^*(\tau + \delta))$.

Note that the historical data available only corresponds to a specific realization of our stochastic predictive model. For this reason, any discrepancy between the actual state $(x^*(\tau + \delta), y^*(\tau + \delta), z^*(\tau + \delta))$ and the estimated means of the random variables $(X_{t+1}, Y_{t+1}, Z_{t+1})$ just defined should be interpreted in light of the variability predicted by the model around those means. While our findings were consistent across all state variables, we only report here actual and predicted values for the number of busy chutes, which is particularly relevant because gridlock is modeled by the event $\{Y_t + Z_t > n\}$.

Figure 1 shows the mean $\mathrm{E}[Y_{t+1} + Z_{t+1}]$ and associated centered empirical range with length $6\sigma[Y_{t+1} + Z_{t+1}]$ estimated at each record time point (τ) over one representative day, along with the actual historical value $y^*(\tau + \delta) + z^*(\tau + \delta)$. Also highlighted in Figure 1 (with a gray background) are the time periods corresponding either to workers' breaks (from approximately 1:30 to 2:00, 5:30 to 6:00, 8:00, 10:15, 12:00 to 13:00, 15:15, 17:30 to 18:00, 20:15, 22:15, 23:30 onwards) or reduced activity due to shift change-over, equipment maintenance, breakdown or repair (around 0:15, 3:15, 11:30 to 12:00, 21:00).

Our main observation on the results shown in Figure 1 is that the time periods when the actual number of busy chutes falls outside of the empirical range predicted by our model coincide almost exactly with the workers' breaks and episodes of equipment maintenance/breakdown mentioned above. Furthermore, in all these periods the model significantly overestimates the number of busy chutes. This is because our model does not explicitly capture the induction stations. This modeling choice is justified during the regular (non-highlighted) working hours, as the induction stations have appropriate processing capacity then. However, induction staffing is drastically reduced during the periods highlighted in Figure 1, so that the actual flow of items into the sorter then is either considerably reduced (maintenance/breakdown) or stopped (work breaks). Indeed, note that the actual number of busy chutes remains constant during all work breaks listed above, which reflects that both actual flows into the sorter (induction) and out of it (packing) are stopped then. While the model correctly captures the packing rate reduction during the highlighted periods through its input data w, it ignores the corresponding decrease in induction rate, leading to the overestimation observed.

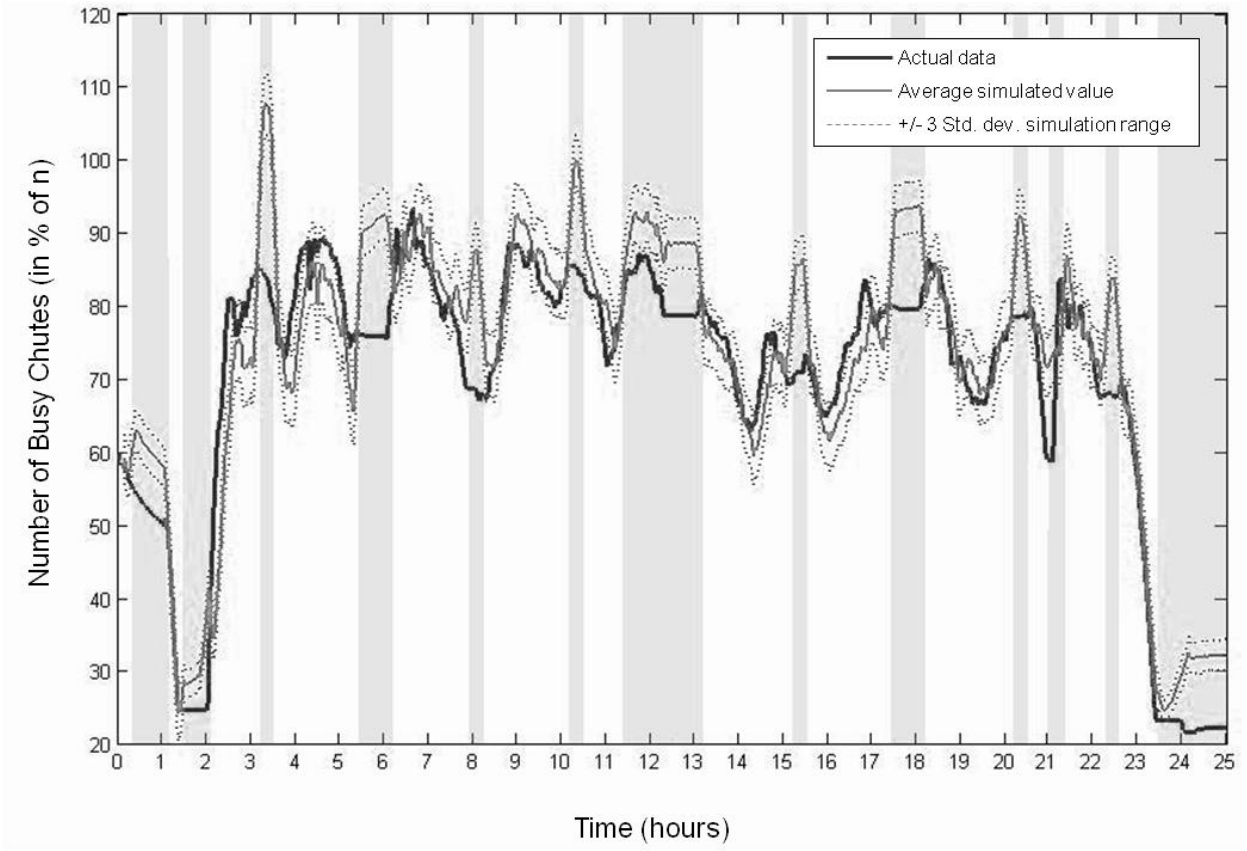


Figure 1: Predicted Distribution $Y_{t+1} + Z_{t+1}$ and Actual Value $y^*(\tau + \delta) + z^*(\tau + \delta)$ of the Number of Busy Chutes over a 24 Hour Period.

While the results observed during the highlighted periods of reduced activity are insightful, they do not pertain to validation since the goals of throughput maximization and gridlock avoidance are not relevant then. During regular work periods the actual number of busy chutes almost always lies within our model's predicted range. Consequently, the approximate dynamics tested appears sufficiently accurate given our purposes.

4.4 Optimization Problem Formulation We now state and discuss the formulation $CDP[\beta]$ which provides the framework of our optimization study:

$$\begin{array}{lll} CDP[\beta]: & \max\limits_{\boldsymbol{\lambda}} & \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t \lambda_t | X_0, Y_0, Z_0] \\ & \text{s.t.:} & \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\{Y_t^{\boldsymbol{\lambda}} + Z_t^{\boldsymbol{\lambda}} > n\}} | X_0, Y_0, Z_0] \leq \beta \\ & & \lambda_t \in [0, \bar{\lambda}] \text{ for all } t \in \mathbb{N}, \end{array} \tag{2}$$

where $\alpha \in (0,1)$ is a discount factor, $1_{\{.\}}$ is an event indicator function, β is the *risk budget* or parameter defining the level of risk tolerated for the event of gridlock (see discussion below), expectations are taken over the sample space of release and service time realizations,

and (X_0, Y_0, Z_0) is the initial state of the system. In (2), the maximum is taken over all stationary closed-loop and non-anticipative policies $\boldsymbol{\lambda}$, and the notations $Y_t^{\boldsymbol{\lambda}}$ and $Z_t^{\boldsymbol{\lambda}}$ reflect the dependence of the model output process (Y_t, Z_t) on the release control policy considered. Since no ambiguity arises here however, we will almost always omit that dependence from now on.

The objective function in (2) captures the goal of maximizing the throughput of the pick-to-ship process considered. Observe however that it is the (discounted) sum of release rates, which are proportional to the process input as opposed to process output – this is justified by the first constraint, which effectively prevents any unbounded accumulation of inventory in the system and is further discussed below. Note that the discount factor α introduces a preference for units shipped in earlier periods. The classical interpretation of such discount factor as one minus the probability of a process termination is appealing here: when running into gridlock, the real pick-to-ship process goes through a lengthy recovery procedure which is not captured by the queueing model described in §4.1. However, the primary reason for us to study here the discounted cost formulation (2) is that it is easier to solve numerically than the natural average cost formulation of the same problem. That latter formulation is formally linked to (2) through the following limiting statements (Blackwell 1962):

$$\begin{cases} \lim\limits_{\alpha\to 1-} (1-\alpha)\mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t \lambda_t | X_0, Y_0, Z_0] = \lim\limits_{k\to\infty} \frac{1}{k}\mathbb{E}[\sum_{t=0}^{k-1} \lambda_t] \\ \lim\limits_{\alpha\to 1-} (1-\alpha)\mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\{Y_t+Z_t>n\}} | X_0, Y_0, Z_0] = \lim\limits_{k\to\infty} \frac{1}{k}\mathbb{E}[\sum_{t=0}^{k-1} 1_{\left\{Y_t^{\boldsymbol{\lambda}}+Z_t^{\boldsymbol{\lambda}}>n\right\}}] = \lim\limits_{t\to\infty} \mathbb{P}(Y_t + Z_t > n) \end{cases} \tag{3}$$

The second and third equality statements in (3) imply that the first constraint in (2) is asymptotically equivalent as $\alpha \to 1$ (the relevant regime given the numerical value we use for α) to the more intuitive expression

$$\lim_{t\to\infty} \mathbb{P}(Y_t + Z_t > n) \leq (1-\alpha)\beta,$$

which specifies an upper bound on the steady-state probability that the system is in a state of gridlock. The unintuitive exact expression of that constraint in (2) (i.e., a discounted sum of indicator functions) is only motivated by technical dynamic programming considerations (see §4.5). From a modeling perspective, that constraint thus balances the throughput maximization objective in (2) with the need to avoid gridlock. Note that it is only a probabilistic statement: because the support of the inter-arrival and service time distributions we use are neither bounded from above or bounded away from zero, with any policy generating positive

release rates it is impossible to guarantee in a deterministic sense that gridlock will never occur.

Finally, observe that the statement of problem $CDP[\beta]$ depends on the initial state (X_0, Y_0, Z_0). However, we have used values of α very close to 1 in our experiments, and observed that the choice of the initial state had very little impact on the results, if any. This is explained in part by the limiting statements (3), where the r.h.s is independent of the initial state, as is typical of the objective of an average cost DP formulation.

4.5 Optimization Algorithm We have specifically formulated the dynamic program $CDP[\beta]$ in (2) so it would belong to a family of constrained Markov decision processes for which some theoretical results and approximate computational methods are available (see Altman 1999 for a review). In particular, we now outline a method for computing a solution to $CDP[\beta]$ by solving a sequence of related unconstrained DPs $UDP[\theta]$ obtained for any $\theta \geq 0$ as

$$\begin{array}{lll} UDP[\theta]: & \max\limits_{\boldsymbol{\lambda}} & \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t \left(\lambda_t - \theta.1_{\{Y_t+Z_t>n\}}\right) | (X_0, Y_0, Z_0) = (x, y, z)] \\ & \text{s.t.:} & \lambda_t \in [0, \bar{\lambda}] \text{ for all } t \in \mathbb{N}, \end{array} \tag{4}$$

where the underlying state dynamics are identical to those of the original problem $CDP[\beta]$. Problem $UDP[\theta]$ is thus a Lagrangian relaxation of $CDP[\beta]$ where the first constraint in (2) is now captured through the objective function and weighted by the multiplier θ, to be interpreted as an instant penalty for gridlock. Define now $\mathbf{j}^\theta(x, y, z)$ as the optimal cost-to-go function for $UDP[\theta]$, equal to $\mathbf{r}^\theta(x, y, z) - \theta.\mathbf{c}^\theta(x, y, z)$ with $\mathbf{r}^\theta(x, y, z) \triangleq \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t \lambda_t^\theta | (X_0, Y_0, Z_0) = (x, y, z)]$ and $\mathbf{c}^\theta(x, y, z) \triangleq \mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\left\{Y_t^{\boldsymbol{\lambda}^\theta} + Z_t^{\boldsymbol{\lambda}^\theta} > n\right\}} | (X_0, Y_0, Z_0) = (x, y, z)]$, where $\boldsymbol{\lambda}^\theta$ is an optimal policy for $UDP[\theta]$. The following results are obtained through straightforward adaptations to the discounted case of the proofs of Lemma 3.1, Theorem 4.3 and Theorem 4.4 from Beutler and Ross (1985) and Corollary 3.5 from Beutler and Ross (1986). In these statements, $\overline{\beta}$ denotes $\mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\left\{Y_t^{\bar{\lambda}} + Z_t^{\bar{\lambda}} > n\right\}} | X_0, Y_0, Z_0]$, i.e., the risk achieved by the constant release policy with rate $\bar{\lambda}$.

Lemma 1 *Assuming $\beta < \overline{\beta}$, there exists a stationary optimal policy $\boldsymbol{\lambda}$ for $CDP[\beta]$ that is deterministic in all states but one, and randomizes between at most two actions in that state. Moreover, $\boldsymbol{\lambda}$ achieves $\mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\{Y_t+Z_t>n\}} | (X_0, Y_0, Z_0) = (x, y, z)] = \beta$ and there exists $\theta^* \geq 0$ such that $\boldsymbol{\lambda}$ is optimal for $UDP[\theta^*]$.*

Lemma 2 *Suppose that for some $\theta \geq 0$ there exists a policy $\boldsymbol{\lambda}^\theta$ such that $\boldsymbol{\lambda}^\theta$ is optimal for $UDP[\theta]$ and achieves $\mathbb{E}[\sum_{t=0}^{+\infty} \alpha^t 1_{\{Y_t+Z_t>n\}} | (X_0, Y_0, Z_0) = (x, y, z)] = \beta$. Then $\boldsymbol{\lambda}^\theta$ is optimal for $CDP[\beta]$.*

Lemma 3 *For any initial state (x, y, z), $\mathbf{j}^\theta(x, y, z)$, $\mathbf{r}^\theta(x, y, z)$ and $\mathbf{c}^\theta(x, y, z)$ are all monotone non-increasing in θ.*

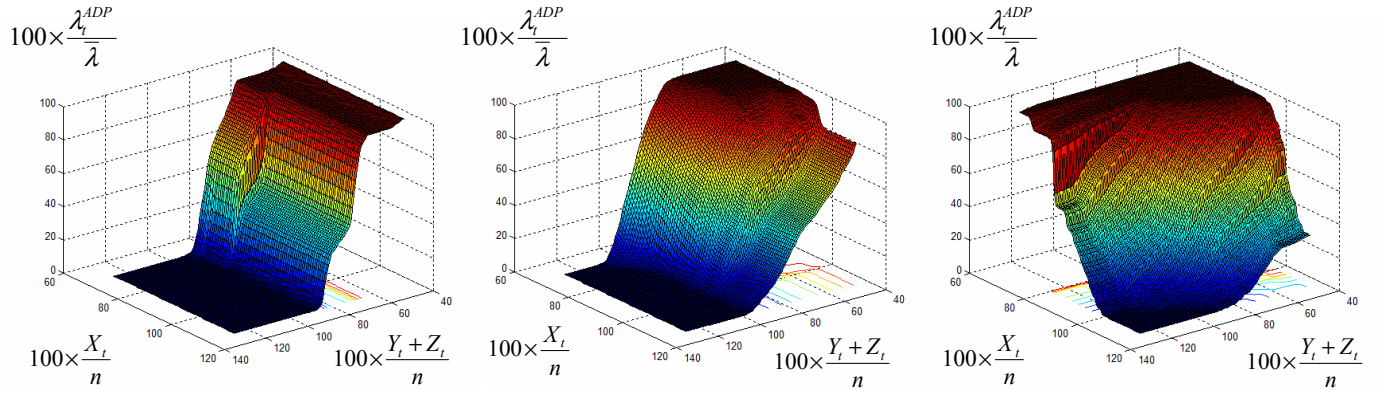
Our solution method consists of performing a line search over θ, where the optimal solution $\boldsymbol{\lambda}^\theta$ to $UDP[\theta]$ is computed at each iteration along with the corresponding cost-to-go functions $\mathbf{j}^\theta$, $\mathbf{r}^\theta$ and $\mathbf{c}^\theta$ using standard approximate DP methods, and the search proceeds until a value of θ achieving $\mathbf{c}^\theta(x, y, z) \approx \beta$ is found. Lemma 1 asserts that such θ exists; Lemma 2 suggests that once such θ is found, the resulting policy $\boldsymbol{\lambda}^\theta$ should be (near) optimal for $CDP[\beta]$; finally the monotonicity of $\mathbf{c}^\theta$ shown in Lemma 3 indicates that an efficient search can be used. The specific algorithm we have implemented is a dichotomic search over a specified interval $[\underline{\theta}, \bar{\theta}]$. In addition, we use an approximate policy iteration algorithm to solve each instance of $UDP[\theta]$, and initialize this algorithm at each iteration of the search with the best policy found in the previous iteration (see the Appendix for algorithmic details, related references and discussion).

In the remainder of this paper, we refer to the policy obtained from the algorithm just stated as ADP^β (the superscript β is omitted when no ambiguity arises) and denote its release rate function as $\lambda^{ADP}(x, y, z)$ or $\lambda_t^{ADP} \triangleq \lambda^{ADP}(X_t, Y_t, Z_t)$.

4.6 Policy Structure We now discuss the qualitative features of policy ADP^β. Because theoretical structural results have so far eluded us, the following discussion is based instead on a large number of consistent empirical observations.

The main observed features of policy ADP^β are illustrated by Figure 2, which includes plots of its normalized release rate λ_t^{ADP} as a function of the reduced and normalized state $(X_t/n, (Y_t + Z_t)/n)$[15] for three representative scenarios characterized by a limiting gridlock probability of 10^{-6} and a number of packers $w \in \{0.75p, p, 1.25p\}$, where p is the average number of packers used by our industrial partner during peak demand periods. A first obvious feature is that the release rate is a decreasing function of the state, in the sense that $\lambda^{ADP}(x', y', z') \leq \lambda^{ADP}(x, y, z)$ when $x' \geq x$, $y' \geq y$ and $z' \geq z$. In particular, the policy releases orders at the maximum rate allowed when the system is almost empty (in the upper

[15] More precisely the function plotted in the plane $(X_t, Y_t + Z_t)$ is $f(x, b) \triangleq \frac{1}{b+1} \sum_{j=0}^{b} \lambda^{ADP}(x, j, b - j)$.



(a) w = 0.75p (b) w = p (c) w = 1.25p

Figure 2: Release Rate Function of Policy ADP^β for the Scenarios $w \in \{0.75p, p, 1.25p\}$ and $\beta(1-\alpha) = 10^{-6}$.

middle corner of all plots in Figure 2), and stops releases altogether in the regions of the state-space corresponding to heavy congestion (as seen in the lower middle corner of the plots).

A second noteworthy feature is that the release rate function displays sudden drops, as seen around $X_t/n \approx 85\%$ in Figure 2 (a), $X_t/n \approx 95\%$ in Figure 2 (b) and $X_t/n \approx 100\%$ in Figure 2 (c). While this is not obvious from these figures alone because of the state reduction used, it can be verified that these drops correspond to transitions of the congestion level $\mathbb{E}[M](X_t + Y_t/2)$ between two consecutive ranges of values characterizing system dynamics (see §4.3.1). Indeed, simulation shows that these drops enable the policy to maintain the system in a desirable steady-state congestion level[16].

Finally, the structure of policy ADP^β is sensitive to the number of packers. Figure 2 (a) shows that with a small number of packers, the release rate function λ_t^{ADP} is almost independent of X_t. In such a regime good policies heavily utilize packing capacity, and any given change in the arrival rate of orders to the third packing station has a substantial impact on the number of green chutes Z_t. Consequently, the ADP^β policy reacts considerably more to changes in the state variable Z_t than to changes of X_t or Y_t, which is easily verified quantitatively (along with all similar statements in this discussion) by examining the relative values of the partial derivatives $\frac{\partial\lambda^{ADP}(x,y,z)}{\partial x}$, $\frac{\partial\lambda^{ADP}(x,y,z)}{\partial y}$ and $\frac{\partial\lambda^{ADP}(x,y,z)}{\partial z}$. In addition, ADP compensates then for even small deviations around an implicit target value for Z_t with drastic changes in its instantaneous release rate, as illustrated by the sudden drop of the

[16] The Online Appendix contains a more extensive discussion on why some congestion levels are preferable to others.

release rate surface seen in Figure 2 (a) around $(Y_t + Z_t)/n \approx 80\%$. In contrast, Figure 2 (c) illustrates that with a high number of packers, policy ADP^β reacts much more to changes in the number of orders in transit X_t than to changes in the number of busy chutes $Y_t + Z_t$, and it can be verified that λ_t^{ADP} is in fact almost independent of Z_t then. In this scenario with low packing utilization, completed chutes are immediately attended to by a packer and the third station experiences almost no queueing. Consequently, any temporary increase of Z_t around its operating steady-state average is absorbed by spare packing capacity, and likely corrected by the time any change in the order release rate can have any impact on the sorter (second and third queueing stations), as the expected time-to-chute $\mathbb{E}[A(g)]$ is long relative to the pack-to-pack time $\mathbb{E}[C]$ (see §4.1).

5 Wave Release Models

The present section discusses how the waveless release model described in §4 can be modified to support a performance comparison with wave-based policies. As observed by Johnson and Lofgren (1994) and others, these more traditional policies effectively decouple the warehouse areas of picking and sorting, since only batches of orders that have been completely picked (the waves) are typically released into the sorter. Our wave release models reflect this decoupling, in that they only consider the picking operation and the conveyor system leading to the sorting area through the assumption that complete waves of picked orders are always ready to be released into the sorter for induction, sorting and packing. This approach enables a meaningful comparison between wave-based and waveless release policies along the performance dimensions of throughput, gridlock probability, packer utilization and sorter utilization. However, it leaves aside the dimensions of order cycle time as well as storage requirements for the intermediary buffer between picking and sorting, which as emphasized in §1 constitute substantial negatives of wave picking. Finally, picker utilization is another important performance dimension that we are unable to explore, because our models do not explicitly capture the detailed layout and resulting picking tours in the picking area (see §4.2.1 and related discussion in §7).

In order to enable a meaningful comparison with waveless policies, we consider models capturing the most sophisticated wave release policies that are either described in the literature or that we have observed in practice (see §3 for background), as described in §5.1 and §5.2 below. Note that we thus leave aside policies combining sorter splitting and wave

overlapping, which are conceptually feasible but satisfy neither of those conditions.

5.1 Overlapping Waves Policy Under this policy (see §3), complete waves of items constituting a number of orders equal to the total number of sorter accumulation chutes (n) are successively released into the sorter. This is modeled using a three station serial queueing network sharing several features with the one stated in §4.1 and defined as follows. First, each wave is modeled as a sequence of items generated by simulating n independent draw from a distribution M of the number of items per order constructed from empirical data, and assuming that the items of each order are uniformely distributed within the wave (as in Hinojosa 1996 and Johnson 1998). Also, the first and last items of each order within the wave are tagged[17], for a reason that will soon become clear.

Note that in the waveless picking model of §4 induction capacity is only considered implicitly through the dependence of the time-to-chute on the congestion upstream of the sorter (see §4.2.2). While justified for waveless release however, that approach is not appropriate to model wave-based release. This is because induction stations are periodically faced then with the sudden release of large batches of orders, so that the congestion generated by their capacity limitations does becomes material over some time periods. The first station in our wave picking model thus represents the induction stations, and it processes individual items from each incoming wave with a service time distribution obtained from empirical data. The second station is an infinite server queue representing as before the incomplete chutes, however its service time for each order is now given endogenously by the time between the completion of its first and last items by the induction station (hence the tagging mentioned above). Finally, the third station represents the packing queue and is identical to that described in §4.1.

In the simplest form of wave picking, each wave is released into the sorter just when the last order of the previous one is packed. As in Bozer et al. (1988)) and Johnson and Lofgren (1994) however, we also consider the more general release policy whereby each new wave is released as a given percentage of the chutes in the sorter (denoted $\Omega \in (0, 100]$) become empty. This policy is easily simulated using the model defined above, and is referred to as W_Ω in the remainder of this paper. Note that the simple non-overlapping policy described earlier corresponds to the particular case W_{100}. Also, overlapping waves ($\Omega < 100$) give rise

[17] The waves do not include single item orders, which are typically processed through a separate process bypassing the sorter.

to the possibility of gridlock, which is still characterized by the event $Y(\tau) + Z(\tau) > n$, with $Y(\tau)$ and $Z(\tau)$ representing as before the number of orders in the second and third stations at time τ, respectively. This leads us to define Ω^β as the wave overlapping parameter such that the corresponding policy W_{Ω^β} achieves the same steady-state gridlock probability as policy ADP^β, i.e., such that $\lim_{\tau\to\infty} \mathbb{P}(Y(\tau) + Z(\tau) > n) = (1-\alpha)\beta$ (see §4.4).

5.2 Split Sorter Policy We now consider the split sorter policy described at the end of §3.1. With n denoting as before the total number of accumulation chutes, this policy involves the sequential release of waves with a number of orders equal to half that quantity, or $n/2$ (e.g., in a sorter with two vertical levels of accumulation chutes, each level would include $n/2$ chutes). As described in §3.1 and Ruben and Jacobs (1992), Russell and Meller (2003) and Perkins (2008), each sorter half (e.g., upper or lower level) is dedicated to a separate wave, and a new wave is released for induction as soon as one of the two halves becomes empty.

The model we use to represent this policy is the same as described in §5.1, except that the size of each wave is halved (see discussion below) and the second station as well as the queue of the third station are duplicated, creating a fork from the first station and a merge into the servers of the third station. Each wave is assigned the second station duplicate which was empty upon its release, and this assignment is implemented at the fork following the first (induction) station. Finally, orders belonging to the oldest wave are given a higher priority by packers. That is, packers may start working on orders from a more recent wave, but only if no order from the previous wave is ready to be packed. In the remainder of this paper, this policy is referred to as $W/2$.

Note that the relative wave sizes we consider for our split sorter and overlapping waves policies described in §5.1 and §5.2 correspond to a sorter where all the chutes can be flexibly used for both accumulation and packing, i.e., without pack-out only chutes (see §3.1). The reasons for this choice are that (i) the sorters with flexible chutes that we consider are very widespread in practice; and (ii) while for the purpose of our benchmark with waveless policies we want to consider the best performing wave-based policies, systems with pack-out only chutes impose the additional restrictions that one part of the sorter storage space can only be used for order accumulation and the other part for order packing. As a result, all else being equal they generate a lower throughput under a split-sorter policy relative to a system without such constraints (for example with pack-out only chutes idling packers may not be

able to work on a wave which has not yet completed accumulation). In addition, under wave-based policies that do not split the sorter (as described in §5.1) these systems reduce the maximum wave size by half, because they do not allow all sorter storage space to be used simultaneously for sorting a single larger wave (see Footnote 6).

6 Simulation Study

The goal of our simulation study is to estimate and understand the relative performance of the main wave-based and waveless release policies described in §3. We first review the policies considered and other experimental design issues in §6.1, then present and discuss our results in §6.2.

6.1 Experimental Design In addition to the waveless release policy ADP^β derived in §4 and the wave-based release policies W_Ω and $W/2$ defined in §5.1 and §5.2 respectively, we also consider the following simple waveless release policies:

Policy CST^β (constant release): Releases orders at the constant rate $\lambda^{CST} \in [0, \bar{\lambda}]$ corresponding to the best constant solution to (2). That rate is easily found by simulation-based search.

Policy CWP^β (constant work in process): Releases orders at a rate given by the function

$$\lambda^{CWP}(x, y, z) = \begin{cases} \bar{\lambda}^{CWP} \text{ if } x + y + z < \bar{k} \\ 0 \text{ otherwise} \end{cases}, \tag{5}$$

where the parameters $\bar{\lambda}^{CWP} \in [0, \bar{\lambda}]$ and $\bar{k} \in \mathbb{N}$ are likewise determined by simulation-based search so that the resulting policy is the best solution to (2) within the family of CONWIP policies defined by (5).

The simulation scenarios we consider are characterized by five different values for the number of packers $w \in \{0.75p, 0.875p, p, 1.125p, 1.25p\}$, where p denotes the average number of packers working in our industrial partner's warehouse during a peak demand period. We also consider two risk values $\bar{\beta}$ (high risk) and $\underline{\beta}$ (low risk), which correspond under our assumed discount factor $\alpha = 0.97$ to limiting gridlock probabilities of $\bar{\beta}(1-\alpha) \simeq 10^{-3}$ and $\underline{\beta}(1-\alpha) \simeq 10^{-6}$ (see §4.4). In practice the level of gridlock risk associated with $\bar{\beta}$ would be unacceptably high, but we consider it here to study the impact of the risk parameter.

The main performance measure we investigate is the simulated *throughput* $\gamma^D \triangleq \mathrm{E}[\sum_{t=0}^{k-1} \lambda_t^D]/k$ of each policy $D \in \{ADP^\beta, CWP^\beta, CST^\beta, W_\Omega, W/2\}$, where the notation λ_t^D denotes the

simulated release rate of policy D at time t and time index k corresponds to 3.5 simulated days (all policies have long reached steady-state by then). In order to enable a meaningful assessment, all throughput results are provided as a ratio to the average throughput γ^{HIST} observed in our industrial partner's warehouse when p packers are assigned to the sorter. Note that for all policies the packing capacity $w(\mathrm{E}[C])^{-1}$ constitutes an upper bound for throughput, and that the *packing utilization* is given by $\gamma^D/\left(w(\mathbb{E}[C])^{-1}\right)$. While policies ADP^β, CWP^β, CST^β and W_{Ω^β} are constructed so that their limiting gridlock probability is set by design to $\beta(1-\alpha)$, we also report the estimated gridlock probability $\mathbb{P}(\text{Gridlock})$ associated with W_Ω for other values of Ω. Finally, we also report the *sorter utilization* given by $\mathbb{E}[Y_\infty^D + Z_\infty^D]/n$, where the numerator is an estimate obtained from simulation for the average number of busy chutes in steady-state under each policy D considered.

6.2 Results and Discussion Table 1 summarizes our simulation results. We discuss next in §6.2.1 the results for the waveless policies then the results for the wave-based policies in §6.2.2. A summary and qualitative performance comparison between the two types of policies is finally provided in §7.

6.2.1 Waveless Release Policies Table 1 shows that for the waveless policies $D \in \{ADP^\beta, CWP^\beta, CST^\beta\}$ considered, the effective packing capacity utilization $\gamma^D/w(\mathbb{E}[C])^{-1}$ is quite high at 97% and above when $w \leq p$, then drops to 92.6% and below for $w = 1.125p$ and even more drastically at 84.1% and below for $w = 1.25p$. We observe that three factors may conceptually constrain throughput in this system: the maximum release rate $\bar{\lambda}$, the packing capacity $w(\mathbb{E}[C])^{-1}$ and the gridlock probability constraint. Because $\bar{\lambda}$ is substantially higher than the packing capacity in all simulation scenarios considered however, only the last two factors are relevant. When $w \leq p$, packing is effectively the system bottleneck and the throughput of all policies remains close to the overall packing capacity. Because packing capacity is an upper bound on the throughput of all policies independently of the gridlock risk β, this also indicates that all three policies are near-optimal then, and that the gridlock probability constraint results in very little throughput loss relative to the unconstrained problem. When the number of packers increases ($w > p$) however, the gridlock constraint effectively becomes the system bottleneck.

A deeper interpretation of these results stems from Theorem 1 in Chao and Scott (2000), which states that the stochastic processes representing the number of jobs in $G/M/w$ queue-

	Number of Packers (w)	$0.75p$	$0.875p$	p	$1.125p$	$1.25p$
	Chutes per Packer (n/w)	73.3	62.8	55	48.8	44
$ADP^{\underline{\beta}}$ [$ADP^{\bar{\beta}}$]	Throughput	78.7 [78.7]	91.5 [91.5]	104.4 [104.4]	109.3 [109.5]	110.3 [110.4]
	Packing Utilization	99.9 [99.9]	99.6 [99.6]	99.4 [99.4]	92.6 [92.8]	84.1 [84.1]
	Sorter Utilization	73.2[73.2]	79.6[79.6]	85.5 [88.2]	78.2 [79]	80 [81.2]
$CWP^{\underline{\beta}}$ [$CWP^{\bar{\beta}}$]	Throughput	78.3 [78.3]	91.3 [91.3]	104 [104.3]	105.3 [106.4]	105.8 [106.5]
	Packing Utilization	99.3 [99.3]	99.4 [99.4]	99.0 [99.3]	89.2 [90.1]	80.6 [81.2]
	Sorter Utilization	88.4[88.8]	88.1[88.2]	80.4 [83.2]	77.1 [78.2]	77.1 [78.9]
$CST^{\underline{\beta}}$ [$CST^{\bar{\beta}}$]	Throughput	77.1 [77.1]	89.5 [90.7]	101.9 [103]	102.7 [104.6]	102.8 [104.7]
	Packing Utilization	97.9 [98]	97.4 [98.7]	97.0 [98.0]	87.0 [88.7]	78.3 [79.8]
	Sorter Utilization	61.7 [64.6]	65.1[69.1]	74.3 [74.5]	74.1 [74.3]	74.9 [75.2]
$W_{\Omega^{\underline{\beta}}}$ [$W_{\Omega^{\bar{\beta}}}$]	Throughput	73.4 [75.4]	85.2 [87.3]	94.7 [98.9]	101.7 [108.2]	109.9 [110.6]
	Packing Utilization	93.7 [96.3]	93.2 [95.5]	90.7 [94.6]	86.4 [92.1]	84.0 [84.6]
	Sorter Utilization	72.6 [74.9]	76.2 [78.2]	78.1 [81.6]	78.5 [83.6]	80 [76.8]
	$\Omega^{\underline{\beta}}$ [$\Omega^{\bar{\beta}}$]	61.2 [58]	54.6 [51.8]	50 [45.6]	44 [40.6]	30.5 [30]
W_{60}	Throughput	74.3	81	86.9	92.1	96.7
	Packing Utilization	94.8	88.6	83.3	78.3	74.0
	Sorter Utilization	73.4	72.5	71.7	71.1	70.5
	$\mathbb{P}$(Gridlock)	0	0	0	0	0
W_{100}	Throughput	53.4	59.3	64.7	69.5	74
	Packing Utilization	68.1	64.8	61.9	59	56.6
	Sorter Utilization	52.7	53	53.3	53.6	53.9
	$\mathbb{P}$(Gridlock)	0	0	0	0	0
$W/2$	Throughput	77.2	90.1	103.2	105.3	105.4
	Packing Utilization	98.4	98.6	94.9	89.6	80.7
	Sorter Utilization	65.7	64.2	61.6	38.2	36.2
	$\mathbb{P}$(Gridlock)	0	0	0	0	0

Table 1: Numerical Simulation Results. Notes: All numbers shown in the third and subsequent rows are percentages. The length of the 95% confidence interval for all simulation results reported is smaller than 0.2% of the corresponding estimate.

ing systems with constant service effort $w(\mathbb{E}[C])^{-1}$ increase with the number of servers w for the stochastic ordering relationship. This implies in our setting that the fractiles of the distribution of busy chutes $Y_t + Z_t$ increase with the number of packers w when the overall packing utilization is held constant, or equivalently that with more packers a lower utilization is required to maintain any of these fractiles at a constant value (which the gridlock probability constraint imposes).

Another relevant insight from queueing theory is that the performance measures of highly congested queues are more sensitive to a given change in their capacity utilization than that of less congested queues. Consequently, when the number of packers is low and packing utilization is high, even a small change in the release rate significant impacts the fractiles of the distribution of busy chutes and the probability of gridlock. Equivalently, a given increase in the tolerated probability of gridlock affords little additional throughput then. Indeed, for

every policy considered in Table 1 the additional average throughput obtained by increasing the gridlock risk parameter from $\underline{\beta}$ to $\bar{\beta}$ increases with the number of packers w, and it is almost negligible for policies ADP^{β} and CWP^{β} in the high congestion scenarios where $w \leq p$. This also explains why the throughput superiority of $ADP^{\underline{\beta}}$ relative to $CWP^{\underline{\beta}}$ increases from 0.4% for $w = p$ to 4.5% for $w = 1.25p$, and that of $CWP^{\underline{\beta}}$ relative to $CST^{\underline{\beta}}$ increases from 2.1% to 3% as w increases from p to $1.25p$ (similar results are observed with $\beta = \underline{\beta}$). Indeed, the range of instantaneous release rates that do not lead to a violation of the gridlock probability constraint is more limited when $w \leq p$ and packing utilization is high. As a result, the greater structural ability of ADP relative to CWP (resp. CWP relative to CST) to dynamically adapt the release rate to process conditions does not provide substantial benefits then. As seen in Figure 3 however, when $w = 1.25p$ policy ADP and to a slightly lesser extent CWP are more able to address temporary stochastic increases of the number of busy chutes $Y_t + Z_t$ above their operating averages by reducing the instantaneous release rate accordingly, which results in a smaller volatility of the process $Y_t + Z_t$, and ultimately maintains higher sorter and packing utilization than CST for the same level of risk.

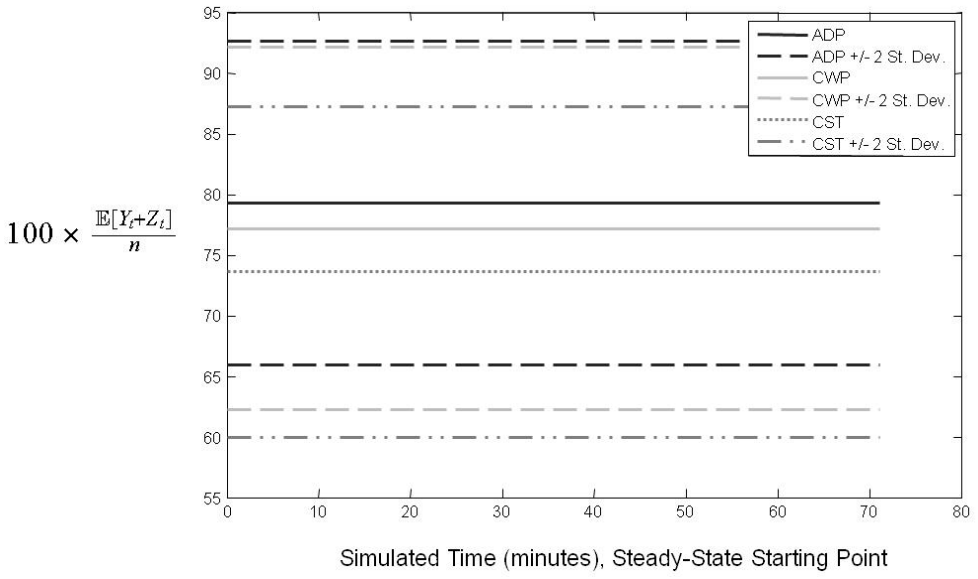


Figure 3: Two Standard Deviation Range of the Steady-State Number of Busy Chutes $\mathbb{E}[Y_t + Z_t] \pm 2\sigma[Y_t + Z_t]$ for Policies $ADP^{\underline{\beta}}$, $CWP^{\underline{\beta}}$ and $CST^{\underline{\beta}}$ with $w = 1.25p$ Packers.
Note: Statistics are computed after 2 days of simulated time.

We also believe that the significant decrease of packing capacity utilization just beyond

the average number of packers p actually used by our industrial partner lends support to the validity of our results. A key performance metric used for the packing team is average worker productivity, defined for any time period of time as the number of orders packed divided by the corresponding number of man$\times$hours used. This metric thus creates a local incentive to maximize packing capacity utilization, which explains that the actual staffing level of packers coincides with the point beyond which the marginal (throughput) return of additional packing capacity starts to markedly decrease. However, we submit that during peak demand periods when the financial benefits of additional throughput and short customer lead-times are particularly high, keeping packing capacity heavily utilized may not be as important per se, and the current policy may result in staffing less packers than is optimal. In that respect, the results shown in Table 1 enable a better understanding of the impact of local staffing policies on system throughput.

Finally, the results shown in Table 1 suggest that, in the case where p packers are assigned to the sorter, policies CST^{β}, CWP^{β} and ADP^{β} may yield a throughput increase of 1.9% to 4.4% relative to the throughput γ^{HIST} experienced by our industrial partner then. Unfortunately, historical performance data was not available to us for other staffing levels than p. However, assuming that the relative performance of CST^{β} and our partner's historical policy would be maintained in such scenarios, we can speculate from Table 1 that policy ADP^{β} (resp. CWP^{β}) would only yield a very modest throughput improvement with fewer packers than p, but an increase in throughput close to 8% (resp. 3%) with 25% more packers than when $w = p$. In any case, our model predicts that the combined use of policy ADP^{β} and addition of 25% more packers than p would increase throughput by about 10%.

6.2.2 Wave-Based Release Policies As should be obvious from their definition in §5, policies W_{Ω} and $W/2$ give rise to a periodic or cyclical steady-state, with a period equal to the average time between consecutive wave releases. To help interpret the results reported in Table 1, Figures 4 (a) to (d) show how the steady-state averages of the main system processes evolve within a period for four wave-based policies of interest.

We first examine the example of policy W_{100} illustrated by Figure 4 (a), in part because it is useful for understanding the other wave-based policies. Its period corresponds to the completion of an entire wave of n orders (the number of sorter chutes), and is characterized by three consecutive phases: (i) as the induction stations start to process the wave of orders

just released, the number of incomplete chutes Y_t drastically increase. Its rate of increase progressively slows down as the items inducted become more likely to belong to an order for which a chute has already been opened. During this first phase, packers mostly idle, and the transition to the second phase occurs at the peak of the number of incomplete chutes Y_t over the period; (ii) as the induction rate of *chute closers* (last inducted item in an order) starts to exceed that of other items, the number of incomplete chutes Y_t starts to decrease and the number of completed orders ready to be packed starts to overwhelm the packing capacity, hence the increase of Z_t. The transition to the third phase occurs when no more incomplete chutes remain, which coincides with the peak of the number of chutes waiting to be packed Z_t; (iii) all busy chutes are occupied by complete orders, which the packers work on until the sorter becomes empty. Note that the dynamics simulated by our model for policy W_{100} are thus very consistent with several empirical observations of its behavior found in the literature (see §3). In particular, because packing activity is concentrated during the last two phases described above, overall packing utilization is relatively low for this policy (less than 64% in all scenarios reported in Table 1). Sorter utilization is also very low (less than 53%) since waves are only released when the sorter is empty[18]. As a result, the throughput of this policy is about 30% lower than that of other policies tested and it is not significantly increased by the addition of packers.

The much better throughput performance of policies W_{Ω^β} and W_{60} seen in Table 1 may be understood by noting from Figure 4 (a) that the peak of the number of busy chutes $Y_t + Z_t$ is very localized within the period of W_{100}, and that the volatility of that process around its mean is relatively small. This suggests that overall throughput may be considerably increased with little additional risk of gridlock by processing several appropriately staggered waves in parallel. Specifically, by releasing another wave late in the second phase (as W_{Ω^β} does) or in the third phase (as W_{60} does) highlighted in Figure 4 (a), the increase of busy chutes corresponding to that second wave coincides with the decrease of busy chutes corresponding to the first one, so that the total number of busy chutes may still be kept below the gridlock threshold (in physics terms, gridlock occurs when consecutive waves approach resonance). Also, packing capacity is better utilized then because the second phase (W_{60}) or third phase (W_{Ω^β}) associated with one wave effectively overlaps with the first phase of the next one.

[18] Sorter utilization can be observed graphically in Figures 4 (a) to (d) as the ratio of the area under the curve $\mathbb{E}[Y_t + Z_t]$ to the total plot area.

Figures 4 (a), (b) and (c) illustrate the evolution of system dynamics when waves are made to increasingly overlap in this manner, and the shorter periods seen in their x-axis show the corresponding improvement in throughput[19] – note also from these Figures that packers utilization increases with this overlap. More specifically, Figure 4 (b) (resp. (c)) shows that, consistent with the definition of W_{60} (resp. $W_{\Omega^{\underline{\beta}}}$), new waves are released when 40% (resp. $1-\Omega^{\underline{\beta}}=69\%$) of the chutes in the sorter are occupied. When the relatively mild overlap $\Omega = 60$ is used, nearly all these chutes contain complete orders from the previous wave. At the beginning of the period, packers work initially on these chutes while new chutes are being opened by inducted items from the current wave, explaining why the total number of busy chutes increases initially at a slower rate for as long as packing work from the previous wave remains. With the more extensive wave overlap $\Omega^{\underline{\beta}} = 31$, induction stations are still processing items from the previous wave upon the release of a new wave. As a result, among all the sorter chutes which are occupied by orders from the previous wave then, approximately 45% contain incomplete orders and only 24% contain complete orders. Because it takes some time for the induction stations to start processing items from the new wave after it is released, the number of busy chutes thus initially decreases at the beginning of the cycle.

Note also that the estimated gridlock probability reported in Table 1 for W_{60} is zero across all scenarios. This shows the existence of a range for the overlap parameter Ω where there is no trade-off between throughput performance and risk. However, policy $W_{\Omega^{\underline{\beta}}}$ and Figure 4 (c) show that the risk of gridlock does appear when the overlap parameter Ω is further decreased – with p packers for example, it is estimated to be 10^{-6} at $\Omega^{\underline{\beta}} = 49$, and 10^{-3} at $\Omega^{\bar{\beta}} = 46$. This follows from the resonance effect mentioned earlier combined with the greater variability of the process $Y_t + Z_t$ representing the number of busy chutes, now generated as the sum of several random processes corresponding to different waves. The numerical values of $\Omega^{\underline{\beta}}$ and $\Omega^{\bar{\beta}}$ for $w = p$ and a fortiori those for $w = 1.25p$ ($\Omega^{\underline{\beta}} = 31$ and $\Omega^{\bar{\beta}} = 30$) also suggest that, below a certain threshold for Ω, the gridlock probability is extremely sensitive to that parameter.

Finally, the dynamics associated with the split sorter policy $W/2$ illustrated by Figure 4 (d) are easily understood by noting that while each sorter half processes then non-overlapping

[19] The number of orders released at the beginning of each period is equal to n for all policies W_Ω considered.

waves of size $n/2$, waves assigned to different sorter halves do overlap. This creates dynamics for the overall number of incomplete, complete and busy chutes which are qualitatively comparable to that observed under W_{60}, with the difference that the peaks of the number of busy chutes under $W/2$ are about twice as small and occur about twice as frequently. As seen in Table 1, this results in relatively high packing utilization (80.7% with $1.25p$ packers), and therefore a throughput performance only slightly lower to that of $W_{\Omega^{\beta}_{-}}$, which is particularly remarkable because $W/2$, in contrast with $W_{\Omega^{\beta}_{-}}$, does not involve any risk of gridlock and does not require the determination of a critical policy parameter such as Ω. It must be pointed out however that the reduction of wave sizes under policy $W/2$ may negatively affect picking labor productivity (see §3, §5.2 and §7).

7 Conclusion

We now discuss the progress presented in this paper towards the two research goals stated in §1:

(Objective 1) *Develop a quantitative model to generate prescriptive control guidelines for waveless picking.* We presented in §4 a queueing model of a waveless operation that was validated empirically using field data (see §4.3.2). This model can be embedded in an optimization formulation for which an approximate DP solution procedure can be used (see §4.4 and §4.5), which amounts to an operational solution to the problem defined in the objective statement. In addition, our simulation experiments based on field data (see §6) suggest that both our DP-based policy (ADP^{β}) and a simpler heuristic such as CONWIP (CWP^{β}) may increase the throughput of an actual waveless operation by several percentage points. They also suggest that local staffing incentives promoting packers' utilization can lead to ignore the system-wide importance of surge packing capacity for increasing throughput and avoiding gridlock.

(Objective 2) *Leverage this model to conduct a rigorous performance comparison between wave-based and waveless picking.* Slight modifications of the waveless model described above allowed us to simulate and understand the performance of the most relevant wave-based release policies used in practice (see §5 and §6.2.2). Because we used the same detailed field input dataset to simulate both wave-based and waveless policies, our methodology enables a meaningful comparison between these two controls. First, contrasting Figures 3 and 4 shows that the qualitative workload patterns seen by the sorter and the packers are strikingly

different under these policies: while wave-based policies give rise to contrasted periodic cycles featuring high predictable variability, their waveless counterparts exhibit constant steady-state averages.

From a throughput standpoint, the results presented in Table 1 suggest that when sufficient packing capacity is available (e.g., $w = 1.25p$), the performance of our waveless policy ADP^{β} is virtually identical to that of the best performing wave policy $W_{\Omega^{\beta}}$ considered. The throughput superiority of ADP^{β} over $W_{\Omega^{\beta}}$ increases with less packing capacity however, because the extent to which waves can overlap without violating the gridlock probability constraint is reduced then (see Figure 4 (a)-(c) and discussion in §6.2.2). Also noteworthy is the poor performance of the simple non-overlapping policy W_{100} relative to the waveless policies ADP^{β} and CWP^{β}. Remarkably, our simulation results of $30 - 40\%$ lower throughput for W_{100} depending on the scenario considered are consistent with the throughput increase of up to 35% reported in Hinojosa (2006) for wave-based facilities switching to waveless processing. Finally, the good throughput performance of the split-sorter policy $W/2$ across all scenarios considered is also remarkable, because that policy is simple to implement, involves no risk of gridlock, and does not require the determination of any parameter or other preliminary computation. For sorters with certain physical designs however, it may have a negative impact on picking productivity (see §3, §5.2 and discussion below).

It should be noted that our wave-based results rely on the key assumption that a complete wave of picked items is always available to be released into the sorter when required. Should it not always be satisfied in practice, the actual throughput of the wave-based policies could be lower than predicted by our model. More broadly, waveless policies seem unquestionably superior along the dimensions of order cycle time, storage space and work-in-process/buffer inventory. On the other hand, a significant advantage of wave-based release policies which should not be underestimated is their simplicity and low implementation cost. Another critical performance dimension that our model leaves aside is picking labor, which account for a significant fraction of many warehouses' operating costs. While a detailed quantitative study of this issue seems a particularly good opportunity for future research, it lies beyond the scope of this paper because any model used for such study would need to capture many more operational details than the one developed in §4, such as storage policies and the layout of the order picking area. We point out however that waveless picking seems to have strong

advantages in that regard. Specifically, under waveless picking no picker is ever starved for work at the end of a wave and, relative to facilities allowing picking waves to overlap, waveless picking does not require a pre-sorting operation before induction for separating items belonging to different waves. Finally, while our model was primarily developed to aid with order release control in split-case warehouse operations, we believe it also applies to most case-based order fulfillment processes with automated sorting.

Acknowledgments. We would first and foremost like to thank our industrial partner for providing an exciting collaboration opportunity between industry and academia, and for partly funding this project. We are also grateful to Jeffrey Wilke, Russell Allgor, Louis Usarzewicz, Brent Beabout, Dean Kassmann, Irina Aronova, Robert Abbondanzza, Mouin Sayegh, Felix Anthony, Nadia Shouraboura, Eric Young, François Rouaix, Adam Baker, Pawel Cholewinski, Timothy Tien, Kevin Liu, Arturo Hinojosa, Marc Austin, Dan Perry, Robert Carver, the participants of the Operations Management Seminar at the MIT Sloan School of Management and three anonymous reviewers for the 2006 MSOM Conference for helpful feedback and discussions. We are also indebted to Steve Graves and Katherine Miller for providing feedback on earlier versions of this paper. The MIT Leaders For Manufacturing Program has provided invaluable support. This work was partly funded by the Singapore-MIT Alliance and the J. Spencer Standish (1945) career development chair of the MIT Sloan School of Management.

References

Altman, E. (1999). *Constrained Markov Decision Processes*. Chapman Hall/CRC.

Apple, J. (2006, October). Attacking the peak. *Modern Materials Handling*.

Armstrong, R. D., W. D. Cook, and A. L. Saipe (1979). Optimal batching in a semi-automated order picking system. *The Journal of the Operational Research Society 30*(8), 711–720.

Beutler, F. J. and K. W. Ross (1985). Optimal policies for controlled markov chains with a constraint. *Journal of Mathematical Analysis and Applications 112*, 236–252.

Beutler, F. J. and K. W. Ross (1986). Time-average optimal constrained semi-markov decision processes. *Advances of Applied Probability 18*, 341–359.

Blackwell, D. (1962). Discrete dynamic programming. *Annals of Mathematical Statistics 33*, 719–726.

Bozer, Y. A., M. A. Quiroz, and G. P. Sharp (1988). An evaluation of alternative control strategies and design issues for automated order accumulation and sortation systems. *Material Flow 4*, 265–282.

Bozer, Y. A. and G. P. Sharp (1985). An empirical evaluation of a general purpose automated order accumulation and sortation system used in batch picking. *Material Flow 2*, 111–131.

Bradley, P. (2007, September). Smoothing the waves. *DC Velocity*.

Chao, X. and C. Scott (2000). Several results on the design of queueing systems. *Operations Research 48*(6), 965–970.

Chew, E.-P. and L. C. Tang (1999). Travel time analysis for general item location assignment in a rectangular warehouse. *European Journal of Operations Research 112*, 582–597.

Cooke, J. A. (2007, April). WCS learn to think for themselves. *DC Velocity*.

de Koster, R., T. Le-Duc, and K. J. Roodbergen (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research 182*, 481–501.

Demery, P. (2007, November). Space launch. *Internet Retailer*.

Duc, T. L. and R. de Koster (2007). Travel time estimation and order batching in a 2-block warehouse. *European Journal of Operations Research 176*, 374–388.

Eick, S. G., W. A. Massey, and W. Whitt (1993). The physics of the Mt/G/inf queue. *Operations Research 41*(4), 731–742.

Gallien, J. and L. M. Wein (2001). A simple and effective component procurement policy for stochastic assembly systems. *Queueing Systems Theory and Applications 38*, 221–248.

Gilmore, D. (2006a, November). Distribution management: Ingram books smooths the waves. *Supply Chain Digest*.

Gilmore, D. (2006b, June). Warehouse management: To wave or not to wave? *Supply Chain Digest*.

Hinojosa, A. (1996, August). Designing distribution centers: Shifting to an automated system. *IIE Solutions*, 32–37.

Hinojosa, A. (2006). What is waveless processing and how can it optimize my operation? Technical report, Fortna, Inc., Atlanta, GA.

Holste, C. (2008, February). When is a sortation system right for distribution operations? *Supply Chain Digest*.

Il-Choe, K., G. P. Sharp, and R. F. Serfozo (1992). Aisle-based order pick systems with batching, zoning, and sorting. *Progress in Material Handling Research*, 245–276.

Johnson, E. (1998). The impact of sorting strategies on automated sortation system perfor-

mance. *IIE Transactions 30*, 67–77.

Johnson, E. and T. Lofgren (1994). Model decomposition speeds distribution center design. *Interfaces 24*(5), 95–106.

Johnson, E. and R. D. Meller (2002). Performance analysis of split-case sorting systems. *Manufacturing Service Operations Management 4*(4), 258–274.

Le-Duc, T. and R. de Koster (2005). Determining number of zones in a pick-and-pack orderpicking system. Technical Report ERS-2005-029-LIS, RSM Erasmus University, The Netherlands.

McMahon, J. (2008, February). American eagle outfitters' new DC stands as a model for combined retail and direct-to-consumer distribution. *Logistics World*.

Meller, R. D. (1997). Optimal order-to-lane assignments in an order Accumulation/Sortation system. *IIE Transactions 29*, 293–301.

Morris, J. (2008, April). *American Eagle Outfitters on Cutting Labor Expenses*.

Norman Saenz, J. (2002, July). Which way to convey. *IIE Solutions*, 35–42.

Owyong, M. and Y. Yih (2006). Picklist generation algorithm with order-consolidation consideration for split-case module-based fulfilment centres. *International Journal of Production Research 44*, 4529–4550.

Perkins, M. (2008, February 1). Vice President, Distribution and Returns Operations, L.L. Bean (Personal Communication).

Perry, D. (2007a). Continuous processing using a sorter. Technical report, Vargo Adaptive Software, LLC., Austin, TX.

Perry, D. (2007b). Process optimization of a sorter-based operation. Technical report, Vargo Adaptive Software, LLC., Austin, TX.

Petersen, C. G. (2000). An evaluation of order picking policies for mail order companies. *Production and Operations Management 9*(4), 319–335.

Ruben, R. A. and F. R. Jacobs (1992). Order processing at land's end: A plant tour. In *Proceedings of the National Decision Science Institute*, pp. 282–284.

Russell, M. L. and R. D. Meller (2003). Cost and throughput modeling of manual and automated order fulfillment systems. *IIE Transactions 35*, 589–603.

Trebilcock, B. (2007, September). American eagle reinvents retail. *Modern Materials Handling*.

Weber, T. (2005). Conditional dynamics of non-markovian, infinite-server queues. Master's thesis, Massachusetts Institute of Technology.

Whitt, W. (2002). *Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Applications to Queues*. New York: Springer-Verlag.

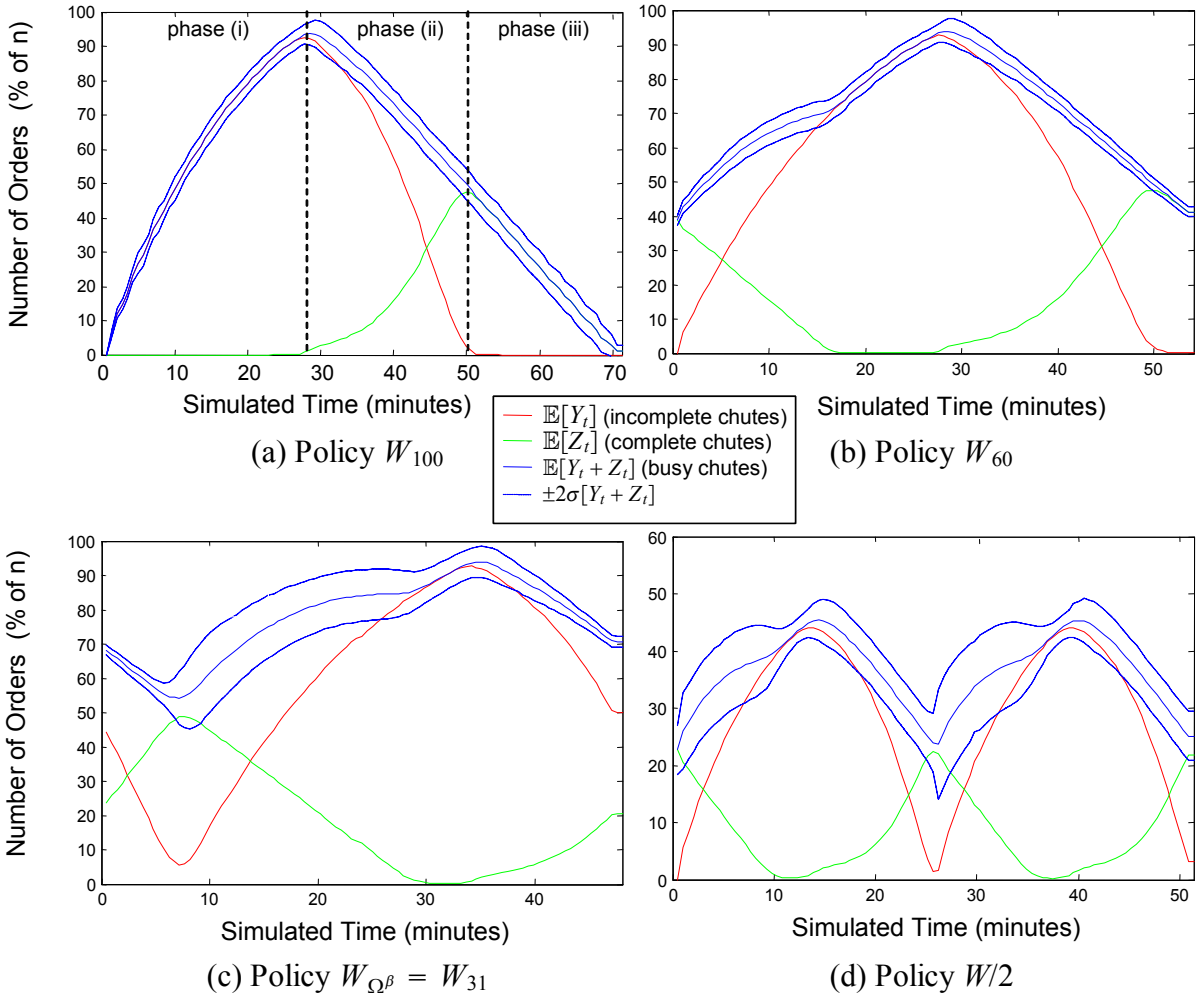


Figure 4: Periodic Cycles of the Average Number of Incomplete Chutes $\mathbb{E}[Y_t]$, Complete Chutes $\mathbb{E}[Z_t]$ and Busy Chutes $\mathbb{E}[Y_t+Z_t]\pm 2\sigma[Y_t+Z_t]$ in Steady-State for Selected Wave-Based Release Policies with $w = 1.25p$ Packers. Notes: Statistics are computed between the first two consecutive wave release times after 2 days of simulated time, except for W/2 for which dynamics following two wave releases are shown.